



ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ  
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ  
ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ  
ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ  
ΕΦΑΡΜΟΣΜΕΝΑ ΗΛΕΚΤΡΟΝΙΚΑ ΣΥΣΤΗΜΑΤΑ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ  
«Σύστημα καταγραφής κοινόχρηστων ενεργειακών  
πόρων σε μια πολυκατοικία»

Του φοιτητή  
Ακριβιάδη Χρήστου  
Αρ. Μητρώου: 52101Μ

Επιβλέπων  
Γιακουμής Άγγελος  
Βαθμίδα  
Επίκουρος Καθηγητής

*Θεσσαλονίκη, Ιούνιος 2023*

Τίτλος Δ.Ε.: «Σύστημα καταγραφής κοινόχρηστων ενεργειακών πόρων σε μια πολυκατοικία»

Κωδικός Δ.Ε.: 22351

Όνοματεπώνυμο φοιτητή: Χρήστος Ακριβιάδης

Όνοματεπώνυμο εισηγητή: Άγγελος Γιακουμής

Ημερομηνία ανάληψης Δ.Ε. 12/03/2023

Ημερομηνία περάτωσης Δ.Ε. 15/06/2023

*Βεβαιώνω ότι είμαι ο συγγραφέας αυτής της εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, έχω καταγράψει τις όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών, εικόνων και κειμένου, είτε αυτές αναφέρονται ακριβώς είτε παραφρασμένες. Επιπλέον, βεβαιώνω ότι αυτή η εργασία προετοιμάστηκε από εμένα προσωπικά, ειδικά ως διπλωματική εργασία, στο Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του ΔΙ.ΠΑ.Ε.*

*Η παρούσα εργασία αποτελεί πνευματική ιδιοκτησία του φοιτητή Χρήστου Ακριβιάδη που την εκπόνησε. Στο πλαίσιο της πολιτικής ανοικτής πρόσβασης, ο συγγραφέας/δημιουργός εκχωρεί στο Διεθνές Πανεπιστήμιο της Ελλάδος άδεια χρήσης του δικαιώματος αναπαραγωγής, δανεισμού, παρουσίασης στο κοινό και ψηφιακής διάχυσης της εργασίας διεθνώς, σε ηλεκτρονική μορφή και σε οποιοδήποτε μέσο, για διδακτικούς και ερευνητικούς σκοπούς, άνευ ανταλλάγματος. Η ανοικτή πρόσβαση στο πλήρες κείμενο της εργασίας, δεν σημαίνει καθ' οιονδήποτε τρόπο παραχώρηση δικαιωμάτων διανοητικής ιδιοκτησίας του συγγραφέα/δημιουργού, ούτε επιτρέπει την αναπαραγωγή, αναδημοσίευση, αντιγραφή, πώληση, εμπορική χρήση, διανομή, έκδοση, μεταφόρτωση (downloading), ανάρτηση (uploading), μετάφραση, τροποποίηση με οποιονδήποτε τρόπο, τμηματικά ή περιληπτικά της εργασίας, χωρίς τη ρητή προηγούμενη έγγραφη συναίνεση του συγγραφέα/δημιουργού.*

Η έγκριση της διπλωματικής εργασίας από το Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του Διεθνούς Πανεπιστημίου της Ελλάδος, δεν υποδηλώνει απαραίτητως και αποδοχή των απόψεων του συγγραφέα, εκ μέρους του Τμήματος.

*«Σε αυτούς που πίστεψαν σε μένα και φυσικά στην Κ.»*



## Πρόλογος

Η παρούσα διπλωματική εργασία αποτελεί το επιστέγασμα των τελευταίων δύο χρόνων, όντας φοιτητής του Μεταπτυχιακού προγράμματος σπουδών, στα Εφαρμοσμένα Ηλεκτρονικά Συστήματα, του Δι.Πα.Ε. Κατά τη διάρκεια των σπουδών, διδάχθηκα μαθήματα που καλύπτουν ένα ευρύ φάσμα της επιστήμης της Ηλεκτρονικής και του Προγραμματισμού. Ωστόσο η μεγάλη μου αγάπη για τα embedded systems και τα αισθητήρια, με έκανε να ξεχωρίσω τρία μαθήματα. Αυτά ήταν οι Εφαρμογές Διαδικτύου των Πραγμάτων, ο Σχεδιασμός Ενσωματωμένων Συστημάτων καθώς και τα Αισθητήρια και Συστήματα Μετρήσεων. Έτσι, η παρούσα διπλωματική εργασία, αποτελεί ένα συνδυασμό των τριών προαναφερθέντων μαθημάτων. Οι γνώσεις που έλαβα κατά τη διάρκεια της διδασκαλίας και άλλες που αποκόμισα στο διάβα της έρευνας μου, ήταν το κλειδί για την ομαλή περάτωση της εργασίας. Μέσα από αυτό το σύντομο αλλά συνάμα μακρύ ταξίδι, αποκόμισα νέες γνώσεις, νέες εμπειρίες και ήρθα αντιμέτωπος με καινούριες προκλήσεις. Όλα αυτά με ωφέλησαν και με βοήθησαν, να εξελιχθώ τόσο προσωπικά όσο και επαγγελματικά, σε τομείς που με ενδιαφέρουν.

## Περίληψη

Οι ενεργειακοί πόροι (πηγές ενέργειας) που συναντάμε σε μια πολυκατοικία (ή σε ένα οποιοδήποτε λειτουργικό οίκημα), είναι το ηλεκτρικό ρεύμα για την παροχή φωτισμού και λειτουργίας του ανελκυστήρα (όταν υπάρχει) καθώς και το μέσο θέρμανσης (Ηλεκτρικό Ρεύμα, Πετρέλαιο, Φυσικό Αέριο ή/και Γεωθερμία). Η παρούσα διπλωματική εργασία, ασχολείται μέσω της προσομοίωσης, με τη καταγραφή των κοινόχρηστων ενεργειακών πόρων, που αφορούν σε πολυκατοικία με κεντρική θέρμανση πετρελαίου. Πιο συγκεκριμένα, το σύστημα αποτελεί ένα συνδυασμός υλικού hardware και λογισμικού software (back-end και front-end). Η συσκευή μέσω του μικροελεγκτή STM32F103C8T6, καταγράφει την κατανάλωση του πετρελαίου θέρμανσης και τις ώρες ενεργοποίησης του λέβητα.

Ακόμα, μιας και σε μια πραγματική εγκατάσταση, αυτή θα τοποθετούνταν στο χώρο του λεβητοστάσιου, έχει προβλεφθεί να φέρει και αισθητήρια με σκοπό την επιτήρηση του χώρου, για λόγους ασφάλειας. Όλα τα δεδομένα που συλλέγονται, αποθηκεύονται στη MySQL βάση δεδομένων, κι έπειτα εμφανίζονται σε μια ιστοσελίδα, προσαρμοσμένη στις ανάγκες μιας πολυκατοικίας. Το back-end κομμάτι της διαδικτυακής εφαρμογής, έγινε μέσω της γλώσσας προγραμματισμού Python και του εργαλείου Flask. Το front-end κομμάτι υλοποιήθηκε μέσω της HTML και των Jinja templates καθώς και της CSS, JavaScript και του εργαλείου Bootstrap. Ο κύριος στόχος του συστήματος, είναι να παρέχει πλήρη διαφάνεια και άμεση ενημέρωση σε κοινόχρηστα ζητήματα και τέλος, μέσω της μακροχρόνιας καταγραφής να επιτυγχάνεται η εξοικονόμηση χρημάτων, όπου αυτή είναι εφικτή.

# «Apartment Building Shared Energy Resources Recording System»

«Christos Akriviadis»

## **Abstract**

The energy resources, (energy sources) existent in an apartment building (or in any other functional building), are the electricity for the power provision of lighting and operation of the elevator (when present) as well as for heating purposes (Electrical Power, Oil, Natural Gas or/and Geothermal Technology). This thesis, deals through simulation, with the recording of shared energy resources, consumed in an apartment building equipped with oil central heating. More specifically, the system is consisted of a combination of hardware and software (back-end and front-end). The STM32F103C8T6 micro-controller of device records the oil consumption and the boiler activation hours.

In addition, since in a real installation, this would be placed in the boiler room area, provisions were taken for a sensor to be also installed in order to monitor the area, for safety purposes. All data collected are stored in the MySQL database, and then are displayed on a website, adapted to the needs of an apartment building. The back-end part of the web application was engineered by the Python programming language and the Flask tool. The front-end part was developed through HTML and Jinja templates as well as CSS, JavaScript and Bootstrap tool. The main goal of the system is to provide complete transparency and immediate information on energy consumption shared issues and finally, through long-term recording, to achieve money savings, wherever is feasible.

## Ευχαριστίες

Αρχικά θα ήθελα να ευχαριστήσω την οικογένεια μου και περισσότερο την μητέρα και τον αδερφό μου, για την πολύτιμη στήριξη τους. Ακόμα, θα ήθελα να ευχαριστήσω όλους όσους ήταν δίπλα μου και με βοήθησαν αυτά τα δύο χρόνια, ώστε να μπορέσω να ολοκληρώσω με επιτυχία, τις μεταπτυχιακές μου σπουδές. Τέλος, ένα μεγάλο ευχαριστώ μέσα από τα βάθη της καρδιάς μου, στον επιβλέπων καθηγητή κ. Αγγελο Γιακουμή, που ήταν πάντα εκεί, κάθε φορά που χρειαζόμουν τη βοήθεια του. Οι γνώσεις και η εμπειρία, σε συνδυασμό με την προθυμία του να μοιράζεται και να βοηθά τους φοιτητές, ήταν αυτό που εισέπραττα σε κάθε πολύωρη συζήτηση μας.



# Περιεχόμενα

Πρόλογος.....	v
Περίληψη.....	vi
Abstract .....	vii
Ευχαριστίες .....	viii
Περιεχόμενα .....	ix
Κατάλογος Εικόνων .....	xii
Κατάλογος Σχημάτων .....	xv
Κατάλογος Πινάκων.....	xv
Συντομογραφίες.....	xvi
Εισαγωγή.....	1
Κεφάλαιο 1ο: Κεντρική θέρμανση και αυτονομία διαμερίσματος.....	3
1.1 Εισαγωγή.....	3
1.2 Ωρομέτρηση .....	4
1.3 Θερμιδομέτρηση.....	5
1.4 Επίλογος .....	5
Κεφάλαιο 2ο: Ηλεκτρονικές διατάξεις και προγραμματιστικές ρυθμίσεις της συσκευής .....	6
2.1 Εισαγωγή.....	6
2.2 Κύκλωμα τροφοδοσίας.....	6
2.3 Μικροελεγκτής STM32F103C8T6 (BluePill).....	7
2.4 Εξωτερική διακοπή EXTI (External Interrupt) .....	11
2.5 Δημιουργία καθυστέρησης σε microseconds .....	12
2.6 Ρύθμιση timer 3 και timer 4 για χρόνο 1 second.....	13
2.7 Κύκλωμα ανίχνευσης ενεργοποίησης-απενεργοποίησης του λέβητα .....	15
2.8 Αισθητήριο θερμοκρασίας-σχετικής υγρασίας DHT22 .....	15
2.9 Αισθητήριο μέτρησης απόστασης HC-SR04 .....	17
2.10 Αισθητήριο καπνού MQ-2 .....	19
2.11 Αισθητήριο ελέγχου στάθμης (magnetic float switch).....	20
2.12 Αντλία .....	21
2.13 Buzzer.....	21
2.14 Οθόνη αφής TFT (2.8 ιντσών) .....	22
2.15 ESP-01.....	25
2.16 Κατασκευή πλακέτας .....	26

2.17	Επίλογος.....	31
Κεφάλαιο 3ο:	Αρχή λειτουργίας της συσκευής.....	32
3.1	Εισαγωγή.....	32
3.2	Ανάλυση αρχής λειτουργίας της συσκευής (κυρίως πρόγραμμα).....	32
3.3	Επίλογος.....	42
Κεφάλαιο 4ο:	Λογισμικό του συστήματος.....	43
4.1	Εισαγωγή.....	43
4.2	Βάση δεδομένων MySQL .....	44
4.3	Flask framework.....	46
4.4	Σύνδεση της εφαρμογής με τη Βάση Δεδομένων.....	48
4.5	Δημιουργία λογαριασμού, είσοδος-έξοδος και επαναφορά κωδικού πρόσβασης.....	50
4.6	Αποστολή δεδομένων από τη συσκευή και επιτήρηση της.....	57
4.7	Αρχική σελίδα .....	61
4.8	Μενού της εφαρμογής.....	63
4.9	Δημιουργία, τροποποίηση ή διαγραφή και προβολή Ανακοίνωσης.....	64
4.10	Ανέβασμα αρχείου .....	72
4.11	Στοιχεία ενοίκων και οι συνολικές τους καταγεγραμμένες ώρες.....	75
4.12	Ιστορικό συμβάντων.....	76
4.13	Ιστορικό Στάθμης δεξαμενής .....	77
4.14	Ιστορικό Ενεργοποίησης λέβητα και Ωρες λειτουργίας του λέβητα.....	78
4.15	Ιστορικό λογαριασμών (αποθηκευμένα αρχεία) .....	80
4.16	Φόρμα επικοινωνίας.....	80
4.17	Ενημέρωση στοιχείων λογαριασμού .....	83
4.18	Επίλογος.....	84
Κεφάλαιο 5ο:	Συμπεράσματα και προτάσεις βελτίωσης.....	85
5.1	Συμπεράσματα.....	85
5.2	Προτάσεις βελτίωσης.....	85
	BIBΛΙΟΓΡΑΦΙΑ.....	87
	ΠΑΡΑΡΤΗΜΑ Α : Κώδικας για το STM32F103C8T6.....	92
	ΠΑΡΑΡΤΗΜΑ Β : Κώδικας για το ESP-01 .....	182
	ΠΑΡΑΡΤΗΜΑ Γ : Scripts για τη Βάση Δεδομένων.....	203
	ΠΑΡΑΡΤΗΜΑ Δ : Κλάσεις για τη Βάση Δεδομένων .....	205
	ΠΑΡΑΡΤΗΜΑ Ε : Κλάσεις για τις φόρμες .....	208
	ΠΑΡΑΡΤΗΜΑ ΣΤ : Αρχικοποιήσεις.....	214
	ΠΑΡΑΡΤΗΜΑ Ζ : Μέθοδοι (routes) .....	216

ΠΑΡΑΡΤΗΜΑ Η : HTML αρχεία .....	253
ΠΑΡΑΡΤΗΜΑ Θ : CSS αρχείο .....	312

## Κατάλογος Εικόνων

Εικόνα 1.1: Γενικό διάγραμμα εγκατάστασης κεντρικής θέρμανσης .....	3
Εικόνα 1.2: Από αριστερά ο αναλογικός [51] και δεξιά ο ψηφιακός [52] ωρομετρητής.....	4
Εικόνα 1.3: Ψηφιακός θερμοδομετρητής [53-54].....	5
Εικόνα 2.1: Μικροελεγκτής STM32F103C8T6 (BluePill) [55] .....	7
Εικόνα 2.2: Επιλογή του μικροελεγκτή στο πρόγραμμα STM32CubeIDE 1.8.0 .....	8
Εικόνα 2.3: Ρύθμιση του Clock στα 72MHz.....	8
Εικόνα 2.4: Από αριστερά η ρύθμιση του Debug και δεξιά η ρύθμιση του HSE.....	9
Εικόνα 2.5: Δεσμευμένα pins για την υλοποίηση της συσκευής .....	9
Εικόνα 2.6: Παραμετροποίηση των pins (1) .....	10
Εικόνα 2.7: Παραμετροποίηση των pins (2) .....	10
Εικόνα 2.8: Κώδικας χειρισμού εξωτερικής διακοπής από το pin PA9 του καναλιού EXTI line[9:5] interrupt.....	12
Εικόνα 2.9: Ρύθμιση του timer 2.....	13
Εικόνα 2.10: Συνάρτηση καθυστέρησης σε microsecond.....	13
Εικόνα 2.11: Ρύθμιση του timer 3.....	14
Εικόνα 2.12: Ρύθμιση του timer 4.....	14
Εικόνα 2.13: Αισθητήριο θερμοκρασίας DHT22.....	16
Εικόνα 2.14: Οπτική απεικόνιση λειτουργίας του αισθητηρίου DHT22 .....	17
Εικόνα 2.15: Αισθητήρας υπερήχων HC-SR04 [56].....	18
Εικόνα 2.16: Αναπαράσταση εκπομπής και ανάκλασης του σήματος [56] .....	18
Εικόνα 2.17: Αισθητήρας καπνού MQ-2 [57].....	19
Εικόνα 2.18: Μαγνητικό αισθητήριο ελέγχου στάθμης (magnetic float switch) [13] .....	20
Εικόνα 2.19: Λειτουργία μαγνητικού αισθητηρίου ελέγχου στάθμης .....	20
Εικόνα 2.20: Αντλία .....	21
Εικόνα 2.21: Buzzer πλακέτας .....	22
Εικόνα 2.22: Οθόνη αφής 2.8 ιντσών [58].....	22
Εικόνα 2.23: Ρυθμίσεις SPI1 .....	23
Εικόνα 2.24: Ρύθμιση DMA για το SPI1 .....	24
Εικόνα 2.25: Ρυθμίσεις SPI2.....	24
Εικόνα 2.26: ESP-01 [59-60] .....	25
Εικόνα 2.27: Ρυθμίσεις USART3.....	25
Εικόνα 2.28: Σχεδίαση πλακέτας 2 όψεων.....	27
Εικόνα 2.29: Πάνω όψη της πλακέτας (top) .....	28
Εικόνα 2.30: Κάτω όψη της πλακέτας (bottom) .....	28
Εικόνα 2.31: Κάτω όψη της τυπωμένης πλακέτας.....	29
Εικόνα 2.32: Πάνω όψη της τυπωμένης πλακέτας.....	29
Εικόνα 2.33: 3D εμφάνιση της πλακέτας μέσω του προγράμματος Proteus .....	30
Εικόνα 2.34: Η κατασκευασμένη πλακέτα (τύπου μητρικής).....	30
Εικόνα 2.35: Η συνολική κατασκευή.....	31
Εικόνα 3.1: Ορισμός 4ψήφιου κωδικού πρόσβασης.....	32
Εικόνα 3.2: Επιβεβαίωση κωδικού πρόσβασης .....	32
Εικόνα 3.3: Μήνυμα λανθασμένου και σωστού κωδικού πρόσβασης.....	33
Εικόνα 3.4: Μενού επιλογών .....	33
Εικόνα 3.5: Μενού επιλογής "Settings" .....	33

Εικόνα 3.6: Επιλογή σύνδεσης στο δίκτυο .....	34
Εικόνα 3.7: Επιλογή καυσίμου κεντρικής θέρμανσης .....	34
Εικόνα 3.8: Καταχώρηση βαθμού απόδοσης του λέβητα .....	35
Εικόνα 3.9: Μέτρηση δεξαμενής πριν και μετά την πλήρωση.....	35
Εικόνα 3.10: Οθόνη αναμονής.....	35
Εικόνα 3.11: Σύνδεση ή επανασύνδεση του ESP-01 στο δίκτυο .....	36
Εικόνα 3.12: Γέμισμα του πίνακα με τα εισερχόμενα δεδομένα .....	36
Εικόνα 3.13: Δημιουργία JSON αρχείου προς αποστολή .....	37
Εικόνα 3.14: Χρόνοι και μεταβλητές για τις μετρήσεις .....	37
Εικόνα 3.15: Διαφοροποίηση πρώτου γράμματος κατά τη σειριακή μεταφορά των δεδομένων .....	38
Εικόνα 3.16: Διαδικασία επεξεργασίας των δεδομένων μέτρησης της δεξαμενής από το ESP-01 .....	38
Εικόνα 3.17: Από δεξιά τα δεδομένα προς σειριακή αποστολή από το μικροελεγκτή ενώ από αριστερά δεδομένα που λήφθηκαν από το ESP-01 και στάλθηκαν στο server .....	39
Εικόνα 3.18: Από δεξιά η ενεργοποίηση του αισθητήρα καπνού μέσω του STM32CubeIDE 1.8.0 και αριστερά τα ληφθέντα δεδομένα στο ESP-01 και η αποστολή τους.....	40
Εικόνα 3.19: Η κατασκευή με την προσομοίωση της δεξαμενής και την τοποθέτηση του πλαισίου κάτω από αυτή καθώς και η τοποθέτηση των αισθητήρων ελέγχου πλημμύρας και διαρροής δεξαμενής....	41
Εικόνα 3.20: Αποστολή κενού POST.....	41
Εικόνα 4.1: Αρχιτεκτονική του συστήματος [62-81].....	43
Εικόνα 4.2: Σύνδεση στον MySQL Server .....	44
Εικόνα 4.3: Αρχική οθόνη του MySQL Workbench και εμφάνιση του πίνακα diamerismata .....	44
Εικόνα 4.4: Ρυθμίσεις των ενεργειών που θα μπορεί να εκτελεί ο χρήστης.....	45
Εικόνα 4.5: Δημιουργία του πίνακα diamerismata.....	46
Εικόνα 4.6: Παράδειγμα ενός Jinja template .....	46
Εικόνα 4.7: Παράδειγμα απλής web εφαρμογής με το Flask.....	47
Εικόνα 4.8: HTML αρχείο και Jinja template .....	47
Εικόνα 4.9: Εμφάνιση των δυο αποτελεσμάτων του παραδείγματος στην ιστοσελίδα .....	48
Εικόνα 4.10: Κλάση Diamerismata για τον πίνακα diamerismata .....	48
Εικόνα 4.11: Κλάση Users για τον πίνακα users.....	49
Εικόνα 4.12: Νέα εγγραφή στον πίνακα users της βάσης δεδομένων.....	49
Εικόνα 4.13: Ερώτημα στη Β.Δ για τον πίνακα users με κριτήριο το email και επιστροφή της πρώτης εγγραφής που θα βρεθεί .....	49
Εικόνα 4.14: Ερώτημα στη Β.Δ για τον πίνακα users με κριτήριο το admin_key και επιστροφή όλων των εγγραφών που θα βρεθούν .....	49
Εικόνα 4.15: Εκκίνηση του server .....	50
Εικόνα 4.16: Τερματικό του PyCharm κατά την εκκίνηση του server .....	50
Εικόνα 4.17: Παράθυρο εισόδου – εγγραφής στην εφαρμογή.....	51
Εικόνα 4.18: Καρτέλα εγγραφής νέου χρήστη.....	51
Εικόνα 4.19: Υποχρεωτικά πεδία της φόρμας εγγραφής .....	52
Εικόνα 4.20: Μέθοδος <i>def signup()</i> .....	52
Εικόνα 4.21: Μήνυμα μη επικύρωσης (validation) κάποιων πεδίων .....	53
Εικόνα 4.22: Μήνυμα επιτυχούς εγγραφής στην εφαρμογή .....	53
Εικόνα 4.23: Εισαγωγή του email για την αλλαγή του κωδικού πρόσβασης .....	54
Εικόνα 4.24: Μήνυμα επιτυχούς ή μη αποστολής του email για τον ορισμό νέου κωδικού πρόσβασης .....	54
Εικόνα 4.25: Email που περιέχει το σύνδεσμο για τον ορισμό νέου κωδικού πρόσβαση .....	55

Εικόνα 4.26: Ορισμός νέου κωδικού πρόσβασης .....	55
Εικόνα 4.27: Διαδικασία αποστολής email με πολλαπλούς παραλήπτες.....	55
Εικόνα 4.28: Επιτυχής σύνδεση και αποσύνδεση από την εφαρμογή.....	56
Εικόνα 4.29: Αποστολή ωρών μέσω του Postman.....	57
Εικόνα 4.30: Εμφάνιση δεδομένων των λίτρων δεξαμενής και των ωρών των διαμερισμάτων στο τερματικό του PyCharm.....	58
Εικόνα 4.31: Εγγραφή και εμφάνιση των δεδομένων στη MySQL βάση.....	58
Εικόνα 4.32: Εισερχόμενο email μετά την ενεργοποίηση του αισθητήρα καπνού.....	59
Εικόνα 4.33: Email προς τους διαχειριστές για την ανάγκη πλήρωσης της δεξαμενής.....	59
Εικόνα 4.34: Email προς τους διαχειριστές ότι η συσκευή βρίσκεται εκτός σύνδεσης.....	59
Εικόνα 4.35: Εμφάνιση κάποιων διεργασιών μέσω του τερματικού (1).....	60
Εικόνα 4.36: Εμφάνιση κάποιων διεργασιών μέσω του τερματικού (2).....	60
Εικόνα 4.37: SQL ερωτήματα στη βάση για την 1/3/2023 .....	61
Εικόνα 4.38: Αρχική σελίδα της εφαρμογής (για την 1/3/2023).....	62
Εικόνα 4.39: Αρχική σελίδα και εμφάνιση δεδομένων για διάστημα από 1/3/2023 έως 7/3/2023.....	63
Εικόνα 4.40: Μενού της εφαρμογής.....	64
Εικόνα 4.41: Δημιουργία νέας ανακοίνωσης .....	65
Εικόνα 4.42: Έλεγχος επικύρωσης δεδομένων .....	65
Εικόνα 4.43: Μέθοδος για τη δημιουργία νέας ανακοίνωσης.....	66
Εικόνα 4.44: Μήνυμα επιτυχούς δημοσίευσης της ανακοίνωσης και η σελίδα των Ανακοινώσεων ..	66
Εικόνα 4.45: Σελίδα Ανακοινώσεων.....	67
Εικόνα 4.46: Εμφάνιση πλήρους ανακοίνωσης και τροποποίηση ή διαγραφή αυτής.....	68
Εικόνα 4.47: Τροποποίηση υπάρχουσας ανακοίνωσης.....	69
Εικόνα 4.48: Μήνυμα επιτυχούς επεξεργασίας ανακοίνωσης .....	69
Εικόνα 4.49: Μήνυμα επιβεβαίωσης διαγραφής ανακοίνωσης.....	70
Εικόνα 4.50: Μήνυμα επιτυχούς διαγραφής της ανακοίνωσης (Ανακοίνωση_5).....	70
Εικόνα 4.51: Όλες οι ανακοινώσεις ενός διαχειριστή.....	71
Εικόνα 4.52: Εμφάνιση των πέντε ανακοινώσεων στη Β.Δ.....	71
Εικόνα 4.53: Εμφάνιση της τροποποιημένης ανακοίνωσης (Ανακοίνωση_5) στη Β.Δ.....	72
Εικόνα 4.54: Εμφάνιση των ανακοινώσεων στη Β.Δ. μετά τη διαγραφή της ανακοίνωσης (Ανακοίνωση_5).....	72
Εικόνα 4.55: Επιλογές για το ανέβασμα αρχείου.....	73
Εικόνα 4.56: Επιλογή αρχείου και φακέλου .....	73
Εικόνα 4.57: Επιτυχής μεταφόρτωση του αρχείου .....	74
Εικόνα 4.58: Μέγεθος αρχείου μεγαλύτερο από 4MB.....	74
Εικόνα 4.59: Εμφάνιση αρχείων στη Β.Δ. ....	75
Εικόνα 4.60: Στοιχεία ενοίκων μέσα από την εφαρμογή .....	75
Εικόνα 4.61: Εμφάνιση στοιχείων των χρηστών στη Β.Δ. ....	75
Εικόνα 4.62: Συνολικές ώρες λειτουργίας των ενοίκων για το διάστημα ημερών 1/3/2023 ~ 7/3/2023 .....	76
Εικόνα 4.63: Εμφάνιση καταγεγραμμένων συμβάντων στην ιστοσελίδα.....	76
Εικόνα 4.64: Καταγεγραμμένα συμβάντα στη βάση δεδομένων .....	77
Εικόνα 4.65: Διάγραμμα κατανάλωσης λίτρων πετρελαίου για το διάστημα 1/3/2023 έως 7/3/2023. 77	
Εικόνα 4.66: Διάγραμμα ημερήσιας κατανάλωσης λίτρων πετρελαίου και λίτρα πλήρωσης της δεξαμενής (4/3/2023) .....	78
Εικόνα 4.67: Διάγραμμα ενεργοποίησης λέβητα και διαγράμματα θερμοκρασίας-υγρασίας .....	79

Εικόνα 4.68: Συνολικές ώρες λειτουργίας του λέβητα για το διάστημα 1/3/2023 έως 11/3/2023.....	80
Εικόνα 4.69: Προβολή όλων των φακέλων με τα αρχεία και αποθήκευση αυτών .....	81
Εικόνα 4.70: Φόρμα επικοινωνίας μεταξύ ενοίκων και διαχειριστών .....	82
Εικόνα 4.71: Μήνυμα επιτυχούς αποστολής του email .....	82
Εικόνα 4.72: Εισερχόμενο email στους ένοικους και τους διαχειριστές, αντίστοιχα .....	83
Εικόνα 4.73: Επιτυχής ενημέρωση στοιχείων λογαριασμού.....	83
Εικόνα 4.74: Ενημέρωση στοιχείων λογαριασμού .....	84

## **Κατάλογος Σχημάτων**

Σχήμα 2.1: Σχηματικό τροφοδοτικού.....	6
Σχήμα 2.2: Κύκλωμα ανίχνευσης ενεργοποίησης και απενεργοποίησης του λέβητα.....	15
Σχήμα 2.3: Συνδεσμολογία αισθητήρων DHT22 .....	16
Σχήμα 2.4: Συνδεσμολογία αισθητηρίου HC-SR04.....	18
Σχήμα 2.5: Συνδεσμολογία αισθητηρίου MQ-2.....	19
Σχήμα 2.6: Συνδεσμολογία μαγνητικών αισθητήρων ελέγχου στάθμης.....	21
Σχήμα 2.7: Κύκλωμα αντλίας.....	22
Σχήμα 2.8: Συνδεσμολογία οθόνης αφής .....	23
Σχήμα 2.9: Συνδεσμολογία ESP-01 .....	26
Σχήμα 2.10: Συνδεσμολογία επιμέρους κυκλωμάτων με τον μικροελεγκτή.....	27
Σχήμα 3.1: Μπλοκ διάγραμμα λειτουργίας της συσκευής.....	42

## **Κατάλογος Πινάκων**

Πίνακας 2.1: Βασικά χαρακτηριστικά του μικροελεγκτή .....	8
Πίνακας 3.1: Συνδυασμός συμβάντων .....	40

## Συντομογραφίες

Δι.Πα.Ε	Διεθνές Πανεπιστήμιο Ελλάδος
τ.μ.	τετραγωνικά μέτρα
κ.ο.κ.	Και ούτω καθεξής
SYS	System
RCC	Reset and Clock Controller
HSE	High Speed Clock
ARR	Auto Reload Register
Β.Δ.	Βάση Δεδομένων



## Εισαγωγή

Σε όλες τις πολυκατοικίες, η διαχείριση γίνεται είτε από άτομα που διαμένουν σε αυτές, είτε από κάποιο γραφείο διαχείρισης πολυκατοικιών. Ο διαχειριστής (ή οι διαχειριστές) είναι υπεύθυνος για τη διασφάλιση της εύρυθμης λειτουργίας της πολυκατοικίας. Μια από τις αρμοδιότητες του, είναι να συλλέγει τους κοινόχρηστους λογαριασμούς. Σε αυτούς συνήθως συγκαταλέγονται το ηλεκτρικό ρεύμα, η καθαριότητα των χώρων, η συντήρηση του ανελκυστήρα και ίσως κάποια έκτακτη ζημιά. Επιπρόσθετα, η συντήρηση του λέβητα όταν υπάρχει κεντρική θέρμανση. Ακόμα στις ευθύνες του, κατά τους χειμερινούς μήνες, εντάσσεται και η πλήρωση της δεξαμενής, όταν το καύσιμο είναι το πετρέλαιο θέρμανσης και τείνει να τελειώσει. Πρέπει να μεριμνά, για την ανά τακτά χρονικά διαστήματα παρακολούθηση της, ώστε η πλήρωση να γίνει σε σωστό χρόνο, για προφανείς λόγους. Ακόμα, κάποιες άλλες αρμοδιότητες, έχουν να κάνουν με την έκδοση των μηνιαίων κοινοχρήστων και την επικοινωνία με τους ένοικους της πολυκατοικίας, για οποιοδήποτε πρόβλημα, ανάγκη ή δυσλειτουργία προκύπτει.

Ίσως κάποιος, διαβάζοντας τις αρμοδιότητες που αναφέρθηκαν, να πιστέψει πως ο ρόλος του διαχειριστή είναι μια σχετική απλή διαδικασία. Κάτι τέτοιο όμως φυσικά και δεν ισχύει, γι' αυτό και αρκετοί είναι αυτοί που δεν επιθυμούν, να βρίσκονται σε αυτή τη θέση. Αρκεί να αναλογιστεί κανείς, πόσο δύσκολη μπορεί να γίνει η επικοινωνία, όταν αυτή επιβάλλεται να είναι μεταξύ πολλών ανθρώπων και με διαφορετικές αντιλήψεις. Στο ίδιο πλαίσιο εντάσσεται και ό,τι έχει να κάνει με οικονομικά ζητήματα. Δεν είναι λίγες οι φορές που η καχυποψία, η ηθελημένη ή μη αδιαφάνεια κάποιων καταστάσεων, ή και η μη έγκαιρη ενημέρωση, οδηγούν σε σοβαρά προβλήματα, μεταξύ των ενοίκων ή μεταξύ των ενοίκων και των διαχειριστών. Ένα κλασσικό παράδειγμα έχει να κάνει με τη συζήτηση για το πετρέλαιο. Κάποιοι μπορεί να αμφιβάλλουν για τα λίτρα που καταναλώθηκαν, θεωρώντας τα πολλά. Ακόμη, μπορούν να εικάσουν πως μια γρήγορη ελάττωση της στάθμης, ενδεχομένως να οφείλεται σε κάποια δυσλειτουργία του καυστήρα, είτε σε κάποιο άλλο πρόβλημα. Άλλοι να αμφισβητήσουν τα λίτρα πλήρωσης κ.ο.κ.

Όλα όσα αναφέρθηκαν, γέννησαν την ιδέα αυτής της διπλωματικής εργασίας. Ένα σύστημα που θα μπορεί ο καθένας να έχει πρόσβαση και να ενημερώνεται για κοινόχρηστα ζητήματα, που τον αφορούν. Μετά από έρευνα που έγινε, διαπιστώθηκε πως ενώ υπάρχουν μεμονωμένα κάποια συστήματα (π.χ. μέτρηση δεξαμενής ή ωρομέτρηση διαμερίσματος), δεν υπάρχει ένα ενιαίο σύστημα που θα χρησιμοποιείται σε μια πολυκατοικία με σκοπό την άμεση ενημέρωση των ενοίκων, την πλήρη διαφάνεια μεταξύ ενοίκων και διαχειριστών και ως ένα βαθμό την εξοικονόμηση χρημάτων. Έτσι, η προσομοίωση που έγινε στα πλαίσια της υλοποίησης της εργασίας, σκοπό έχει να αναδείξει πως αυτά τα καίριας σημασίας ζητήματα, μπορούν να λυθούν μέσω της τεχνολογίας.

Η υλοποίηση της ιδέας χωρίζεται σε δυο μέρη. Το πρώτο αφορά το hardware, δηλαδή τη συσκευή που φέρει τα αισθητήρια. Αυτή σχεδιάστηκε ώστε να τοποθετείται στο χώρο του λεβητοστάσιου και πάνω της να συνδέονται αισθητήρια που καταγράφουν, την εσωτερική και εξωτερική θερμοκρασία όταν ο λέβητας βρίσκεται σε λειτουργία. Επίσης, συνδέεται το αισθητήριο μέτρησης στάθμης δεξαμενής και αισθητήρια που ενεργοποιούνται, αν υπάρξει διαρροή αυτής. Ακόμα, υπάρχουν και κάποια άλλα αισθητήρια, για τον έλεγχο του χώρου. Το δεύτερο μέρος αφορά το λογισμικό, δηλαδή τη σύνδεση της βάσης δεδομένων με το server και την προώθηση των δεδομένων στην ιστοσελίδα.

Ολοκληρώνοντας, στο πρώτο κεφάλαιο γίνεται αναφορά στη κεντρική θέρμανση και στην αυτονομία των διαμερισμάτων, με τις λύσεις της ωρομέτρησης και της θερμοδομέτρησης. Στο δεύτερο και τρίτο

## Εισαγωγή

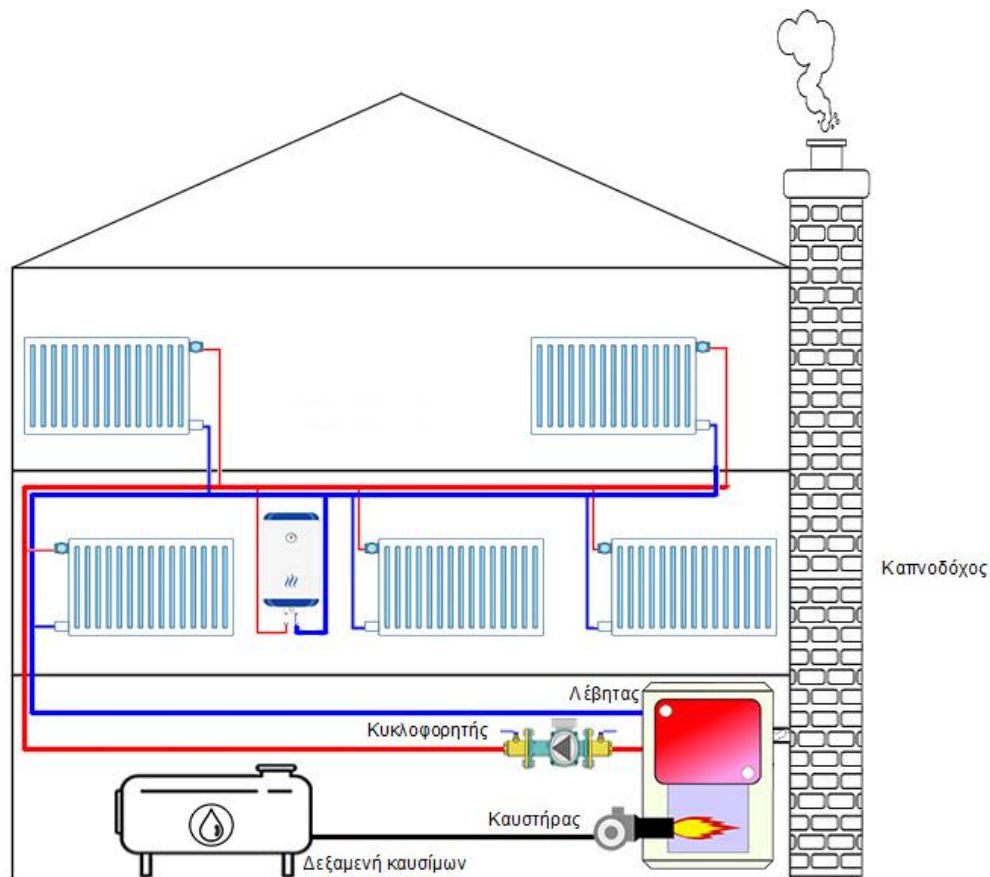
κεφάλαιο αναλύεται εκτενώς, το κατασκευαστικό κομμάτι της συσκευής. Στο τέταρτο κεφάλαιο παρουσιάζεται ενδελεχώς, το κομμάτι που αφορά το λογισμικό του συστήματος, περιλαμβάνοντας τη βάση δεδομένων, το back-end και το front-end. Τέλος, το πέμπτο κεφάλαιο, περιλαμβάνει τα συμπεράσματα του εγχειρήματος, καθώς και τις μελλοντικές βελτιώσεις που μπορεί να υποστηρίξει.

## Κεφάλαιο 1ο: Κεντρική θέρμανση και αυτονομία διαμερίσματος

### 1.1 Εισαγωγή

Κεντρική θέρμανση καλείται η εγκατάσταση στην οποία, ένας μόνο λέβητας χρησιμοποιείται, με σκοπό να παράγει θερμότητα, για όλους του χώρους ενός κτηριακού συγκροτήματος. Η εγκατάσταση αποτελείται από τον λέβητα, τον καυστήρα, την καπνοδόχο και τον κυκλοφορητή. Ακόμα, από τη δεξαμενή καυσίμων πετρελαίου (όταν το μέσο θέρμανσης είναι το πετρέλαιο), τις σωληνώσεις και τα θερμαντικά σώματα του κάθε διαμερίσματος. Την εγκατάσταση συμπληρώνουν επίσης, κάποιες διατάξεις ασφαλείας και τις τελευταίες δεκαετίες, τα όργανα μέτρησης αυτονομίας του κάθε διαμερίσματος. Τέλος, το μέσο καύσης μπορεί να είναι το πετρέλαιο ή το φυσικό αέριο [1]. Ένα γενικό διάγραμμα μιας τέτοιας εγκατάστασης, παρατίθεται στην Εικόνα 1.1.

Στις πρώτες πολυκατοικίες με κεντρική θέρμανση, στην αρχή κάθε χειμερινής σεζόν γίνονταν συνέλευση, με σκοπό οι ένοικοι να συμφωνήσουν για τις ώρες της ημέρας, που θα λειτουργούσε η θέρμανση. Και κάπου εκεί ήταν που ξεκινούσαν οι διενέξεις μεταξύ των ενοίκων, γιατί σχεδόν ποτέ δεν μπορούσε να βρεθεί, μια κοινή γραμμή συνεννόησης (που πολλές φορές μπορεί να έκρυβε και σκοπιμότητες). Ένα δεύτερο πρόβλημα που προέκυπτε, ήταν όταν έπρεπε να γίνει ο καταμερισμός του ποσού, που θα πλήρωνε στο τέλος του μήνα, κάθε διαμέρισμα. Με αυτή τη μέθοδο, δεν μπορούσε να γίνει δίκαιη κατανομή. Κάθε διαμέρισμα είχε διαφορετικές ανάγκες σε θερμότητα, ή ακόμα και ένα διαμέρισμα μπορεί να μην κατοικούνταν, αλλά ο ιδιοκτήτης καλούνταν να πληρώσει το ποσό που του



Εικόνα 1.1: Γενικό διάγραμμα εγκατάστασης κεντρικής θέρμανσης

αναλογούσε. Αυτά ήταν δύο από τα μεγαλύτερα προβλήματα, που οδήγησαν στην ανάγκη να βρεθεί ένας πιο δίκαιος τρόπος, κατανομής των δαπανών θέρμανσης. Έτσι, η λύση δόθηκε μέσα από την αυτονόμηση του διαμερίσματος, χωρίς αυτό να αποσπάται από τη κεντρική θέρμανση. Ο πρώτος τρόπος αυτονομίησης υλοποιείται με την ωρομέτρηση, ενώ ένας δεύτερος τρόπος είναι με θερμιδομέτρηση.

### 1.2 Ωρομέτρηση

Ο ωρομετρητής είναι μια συσκευή, που χρησιμοποιείται για κάθε διαμέρισμα ξεχωριστά. Τοποθετείται είτε μετά τον κυκλοφορητή του λέβητα, στο χώρο του λεβητοστάσιου, είτε σε κάποιο σημείο με εύκολη πρόσβαση (στο κεντρικό διάδρομο, έξω από κάθε διαμέρισμα). Η συσκευή μπορεί να είναι αναλογική ή ψηφιακή (Εικόνα 1.2) και καταγράφει τις ώρες όπου ο ένοικος του διαμερίσματος, ενεργοποίησε τον ατομικό του θερμοστάτη. Ανάλογα τις ώρες που έχει καταγράψει η συσκευή, γίνεται η αντιστοιχία του ποσού που καλείται να πληρώσει.

Αξίζει να σημειωθεί πως ο συγκεκριμένος τρόπος, δεν είναι και ο πιο δίκαιος. Υπάρχουν δύο παραδείγματα που αναδεικνύουν με σαφήνεια, το γιατί. Δεδομένου ότι το σύστημα μετράει ώρες, κι έστω ότι ο λέβητας είναι σβηστός, άρα δεν κυκλοφορεί ζεστό νερό στις σωληνώσεις. Όταν κάποιο διαμέρισμα θα ενεργοποιήσει πρώτο το θερμοστάτη του (άρα θέσει σε λειτουργία τον λέβητα), θα καταγραφούν στις δικές του ώρες, αθροιστικά δύο διαστήματα χρόνου. Το πρώτο διάστημα αφορά το χρόνο που θα χρειαστεί να λειτουργήσει ο λέβητας, μέχρι το νερό να φτάσει στη συγκεκριμένη θερμοκρασία που απαιτείται. Τότε, θα ενεργοποιηθεί η ηλεκτροβάννα και θα τεθεί σε λειτουργία ο κυκλοφορητής, ώστε να γίνει η ροή του ζεστού νερού προς τα θερμαντικά σώματα. Το δεύτερο διάστημα, θα είναι αυτό στο οποίο ο θερμοστάτης θα είναι ανοιχτός, για να ζεστάνει το διαμέρισμα. Συνεπώς, όλα τα επόμενα διαμερίσματα δεν θα έχουν καταγράψει το πρώτο χρονικό διάστημα, αφού αυτό “χρεώνεται” στο πρώτο διαμέρισμα που εκκίνησε τον λέβητα.

Το επόμενο παράδειγμα έχει να κάνει με τη διαφορά των τ.μ., καθώς και με τον αριθμό των σωμάτων κάθε διαμερίσματος. Έστω ότι το 1<sup>ο</sup> διαμέρισμα είναι 80τ.μ., με 6 θερμαντικά σώματα και το 2<sup>ο</sup> διαμέρισμα είναι 40τ.μ., με 3 θερμαντικά σώματα. Γίνεται εύκολα αντιληπτό, πως τα δύο διαμερίσματα δεν έχουν τις ίδιες ανάγκες θερμικής ισχύος. Αν όμως ενεργοποιήσουν ταυτόχρονα τους θερμοστάτες τους, οι συσκευές τους θα καταγράψουν τις ίδιες ώρες λειτουργίας (άσχετα με το ότι το πρώτο διαμέρισμα, θα απορροφήσει μεγαλύτερη ενέργεια και θα χρειαστεί να γίνει καύση περισσότερων λίτρων πετρελαίου).

Η ενδιάμεση λύση που υιοθετείται σε αυτή τη μέθοδο, είναι πέρα των ωρών καταγραφής και ένα μηνιαίο πάγιο (μεγαλύτερο από αυτό της θερμιδομέτρησης), που αφορά τις απώλειες, καθώς και τις ανάγκες σε θερμότητα του κάθε διαμερίσματος.



Εικόνα 1.2: Από αριστερά ο αναλογικός [51] και δεξιά ο ψηφιακός [52] ωρομετρητής



Εικόνα 1.3: Ψηφιακός θερμιδομετρητής [53-54]

Τέλος, αν και η ωρομέτρηση δεν είναι το πιο δίκαιο σύστημα, ακόμα και σήμερα χρησιμοποιείται σε πολλές πολυκατοικίες. Ο λόγος που προτιμάται, είναι επειδή είναι οικονομικότερο (έως και 5 φορές), συγκριτικά με την πιο ορθή λύση, αυτή της θερμιδομέτρησης.

### 1.3 Θερμιδομέτρηση

Όπως ο ωρομετρητής έτσι και ο θερμιδομετρητής (Εικόνα 1.3), είναι μια συσκευή που χρησιμοποιείται σε κάθε διαμέρισμα, με σκοπό την καταγραφή των θερμίδων (και όχι των ωρών), όταν ενεργοποιείται ο ατομικός θερμοστάτης. Είναι η πιο ακριβοδίκαιη λύση για την καταγραφή, σε πολυκατοικία με κεντρική θέρμανση, καθώς καταγράφει την πραγματική ενέργεια που καταναλώθηκε. Ως επί το πλείστον, οι συσκευές αυτές είναι ψηφιακές και η αρχή λειτουργίας τους βασίζεται στη μέτρηση παροχής νερού, καθώς και τις θερμοκρασίες προσαγωγής και επιστροφής στο κύκλωμα θέρμανσης. Έτσι μέσω των στοιχείων αυτών, υπολογίζεται η πραγματική καταναλισκόμενη ενέργεια. Συνεπώς, ο ένοικος καλείται να πληρώσει το ανάλογο ποσό, για τις θερμίδες που έχει καταγράψει η συσκευή του, συν ένα μικρότερο πάγιο (συγκριτικά με αυτό της ωρομέτρησης), που αφορά κυρίως τις απώλειες του διαμερίσματος. Αν και είναι η δικαιότερη λύση, δεν προτιμάται λόγω του κόστους εγκατάστασης, καθώς είναι δύσκολο να συμφωνήσουν όλοι οι ιδιοκτήτες, για την τοποθέτηση μιας τόσο ακριβής συσκευής.

### 1.4 Επίλογος

Στο κεφάλαιο αυτό παρουσιάστηκε συνοπτικά η σπονδύλωση μιας κεντρικής εγκατάστασης θέρμανσης και στα προβλήματα που δημιουργήθηκαν τα πρώτα χρόνια χρήσης της, σε μια πολυκατοικία. Επίσης έγινε αναφορά στις λύσεις που δόθηκαν, με την πάροδο των χρόνων και παρουσιάστηκαν τα πιο βασικά στοιχεία του ωρομετρητή και του θερμιδομετρητή. Όλα συνοδεύτηκαν από κάποιες ενδεικτικές φωτογραφίες, που σκοπό είχαν να συμβάλουν στη βαθύτερη κατανόηση, όσων παρουσιάστηκαν.

## Κεφάλαιο 2ο: Ηλεκτρονικές διατάξεις και προγραμματιστικές ρυθμίσεις της συσκευής

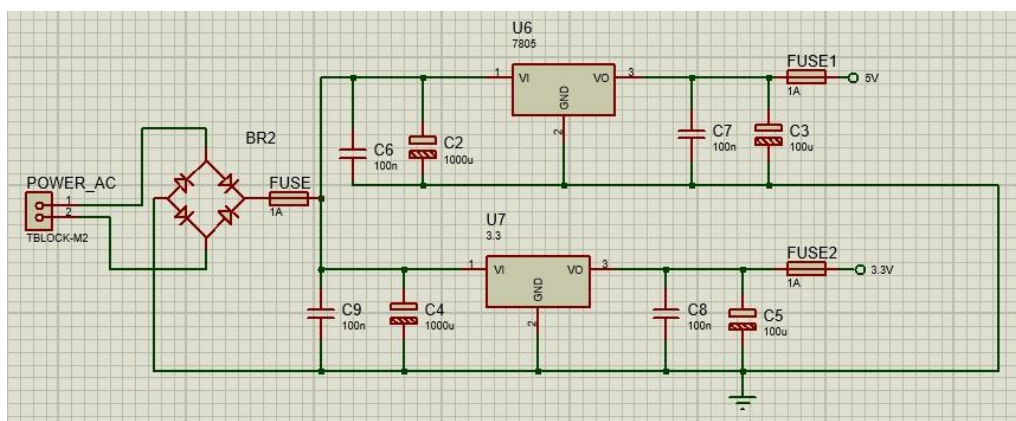
### 2.1 Εισαγωγή

Το γενικό σύστημα καταγραφής, μπορεί να διαχωριστεί σε δυο τμήματα. Στο πρώτο περιλαμβάνεται ο χώρος του λεβητοστάσιου, όπου εκεί υπάρχει εγκατεστημένος ο λέβητας και η δεξαμενή πετρελαίου. Το δεύτερο περιλαμβάνει ότι σχετίζεται με την καταγραφή των ωρών, στα διαμερίσματα. Η παρούσα διπλωματική εργασία, ασχολείται εξ ολοκλήρου με το πρώτο κομμάτι. Πάρθηκε αυτή η απόφαση, αφού στην αγορά το δεύτερο κομμάτι έχει ήδη υλοποιηθεί. Υπάρχουν συστήματα καταγραφής ωρών, που μπορούν να υποστηρίξουν διάφορες δυνατότητες [2-3].

Συνεπώς, η υλοποιηθείσα κατασκευή αποτελείται από το κύκλωμα τροφοδοσίας, τον μικροελεγκτή STM32F103C8T6 και δυο αισθητήρια θερμοκρασίας-υγρασίας DHT22. Ακόμα, υπάρχει ένας αισθητήρας υπερήχων HCSR04, για τη μέτρηση της δεξαμενής. Έπειτα, υπάρχουν τέσσερα μαγνητικά αισθητήρια ελέγχου στάθμης (magnetic float switch) και ένα κύκλωμα που ανιχνεύει την ενεργοποίηση του λέβητα. Επιπρόσθετα, η συσκευή φέρει ένα αισθητήριο καπνού, μια TFT οθόνη αφής 2.8 ιντσών, ένα buzzer και ένα ESP-01 για την αποστολή των δεδομένων. Τέλος, περιέχει και το κύκλωμα ενεργοποίησης μιας αντλίας, που τοποθετείται δίπλα στη δεξαμενή.

### 2.2 Κύκλωμα τροφοδοσίας

Τα επιμέρους κυκλώματα της συσκευής, έχουν απαιτήσεις για συνεχή τάση 3.3V και 5V. Το κύκλωμα τροφοδοσίας (Σχήμα (2.1)) αποτελείται από το μετασχηματιστή, όπου υποβιβάζει την εναλλασσόμενη τάση των 230V AC του δικτύου σε 9V AC. Ακολουθεί μια κοινή γέφυρα ανόρθωσης, όπου ανορθώνει την AC τάση και παρεμβάλλεται μια γενική ασφάλεια, μεταξύ της γέφυρας και των πυκνωτών που έπονται. Αυτοί αποτελούν ένα παράλληλο ζεύγος, ηλεκτρολυτικού και πολυεστερικού πυκνωτή. Ο πρώτος χρησιμοποιείται για την εξομάλυνση ενώ ο δεύτερος ως φίλτρο, με σκοπό να οδηγήσει οποιοσδήποτε τυχαίες ακμές (spikes) στη γείωση. Υπάρχουν δυο ίδια ζεύγη, όπου το ένα άκρο συνδέεται στη κοινή γείωση, ενώ το άλλο στην είσοδο κάθε ολοκληρωμένου κυκλώματος. Στην έξοδο του κάθε σταθεροποιητή, ακολουθεί ακόμα ένα ζεύγος πυκνωτών, με μικρότερες τιμές. Όπως και πριν σκοπό έχει, την επιπλέον εξομάλυνση και εξάλειψη τυχόν θορύβων. Τέλος, η κάθε σταθεροποιημένη τάση, θωρακίζεται με ασφάλειες πλακέτας. Η ύπαρξη τους, στοχεύει στην προστασία του κυκλώματος από κάποιο βραχυκύκλωμα, ή μεγαλύτερο ρεύμα από το επιθυμητό [4-5].



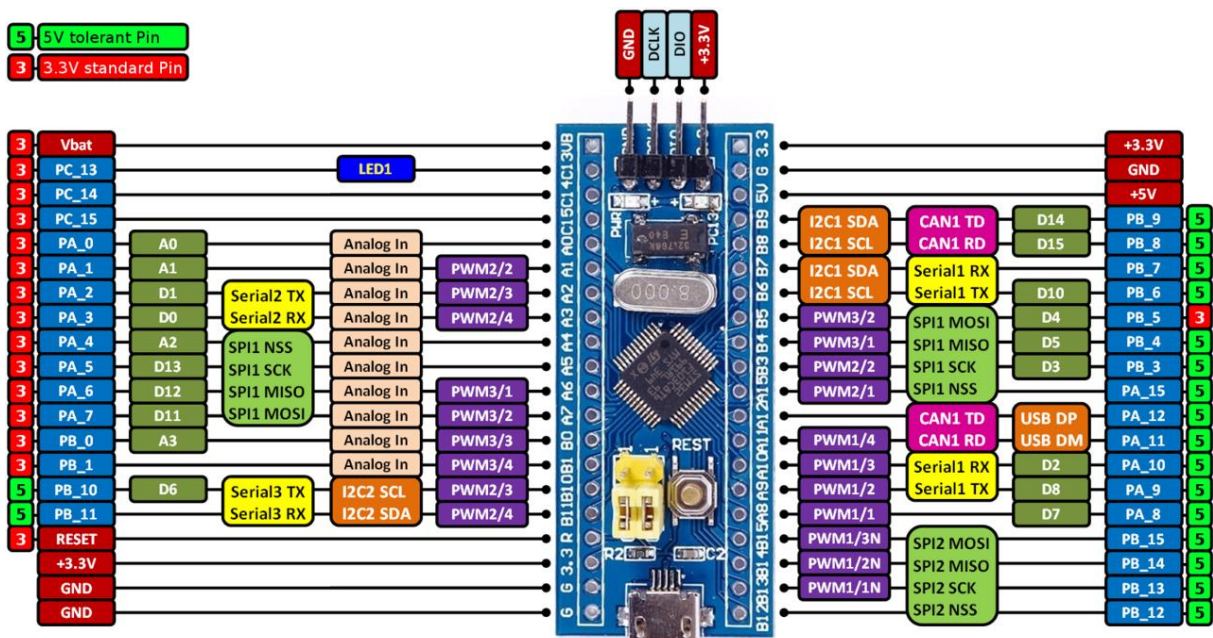
Σχήμα 2.1: Σχηματικό τροφοδοτικού

Στο σημείο αυτό, πρέπει να επισημανθεί πως η συσκευή να μπορεί να τροφοδοτείται εναλλακτικά (όπως και γίνεται) με DC τάση 9V, αντί για εναλλασσόμενη. Αρκεί να αφαιρεθεί ο μετασχηματιστής, η γέφυρα και να βραχυκυκλώσουν τα αντίστοιχα pins. Η αρχή λειτουργίας παραμένει η ίδια με αυτή που αναλύθηκε για την AC τάση.

### 2.3 Μικροελεγκτής STM32F103C8T6 (BluePill)

Η καρδιά της συσκευής καταγραφής, είναι ο μικροελεγκτής STM32F103C8T6, όπου παρουσιάζεται στην Εικόνα 2.1 [6]. Τροφοδοτείται με 3.3V DC και είναι υπεύθυνος για τον έλεγχο των αισθητηρίων, καθώς και τη λήψη μετρήσεων και τη σειριακή αποστολή των δεδομένων στο ESP-01. Ακόμα, είναι υπεύθυνος για την αμφίδρομη επικοινωνία με την οθόνη. Επιλέχθηκε ο συγκεκριμένος καθώς είναι μια αξιοπρεπή επιλογή, που συνδυάζει αρκετά καλά χαρακτηριστικά, με το χαμηλό κόστος. Τα πιο βασικά που σχετίζονται με τη κατασκευή και λήφθηκαν υπόψιν, παρουσιάζονται στον Πίνακα 2.1.

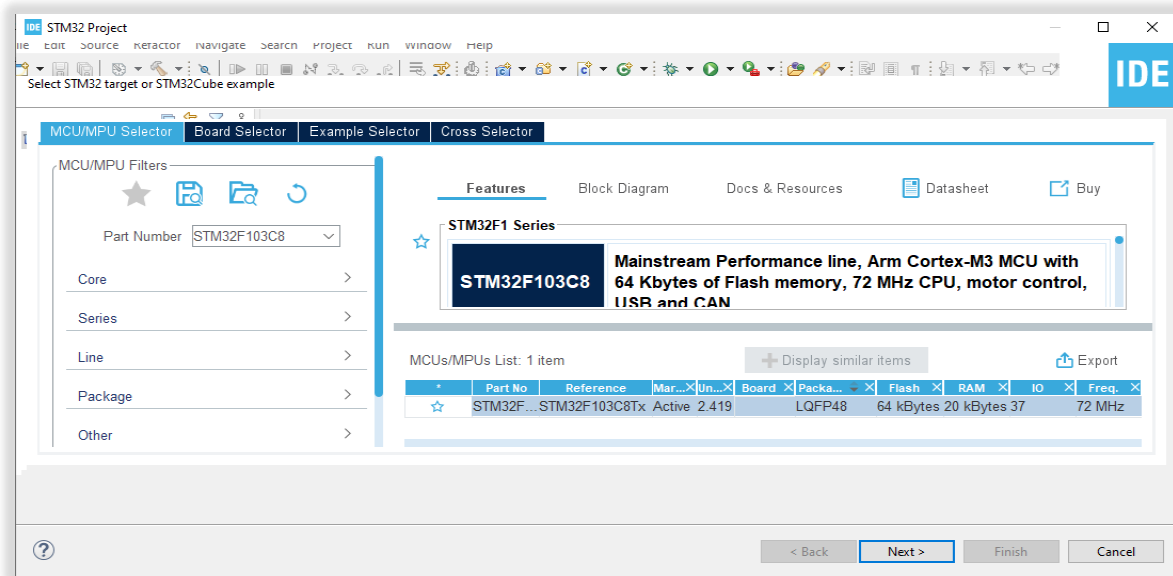
Ο προγραμματισμός του έγινε μέσω του προγράμματος STM32CubeIDE 1.8.0 σε γλώσσα C (embedded C), ενώ χρησιμοποιήθηκε η βιβλιοθήκη HAL [7]. Οι Εικόνες 2.2 έως 2.4 παρουσιάζουν τα πρώτα βασικά βήματα, του προγραμματισμού του. Αρχικά έγινε η επιλογή του μικροελεγκτή (Εικόνα 2.2), ρυθμίστηκε το Clock στα 72MHz (Εικόνα 2.3) και στη συνέχεια επιλέχθηκε μέσω του SYS, το Debug σε Serial Wire (Εικόνα 2.4). Αυτό έγινε ώστε να υπάρχει η δυνατότητα του debugging και serial monitoring, κατά τη διάρκεια συγγραφής και δοκιμών του κώδικα. Επιπλέον, από το RCC ρυθμίστηκε το HSE σε Crystal/Ceramic Resonator (Εικόνα 2.4). Στις Εικόνες 2.5 έως 2.7 μέσω του προγράμματος, αποτυπώνονται όσα pins δεσμεύτηκαν για τη λειτουργία της συσκευής και πως αυτά έχουν παραμετροποιηθεί. Τέλος, στις επόμενες ενότητες, όπου κρίνεται αναγκαίο, θα γίνεται μια πιο περιγραφική αναφορά στα αντίστοιχα pins και τις λειτουργίες τους, με στόχο τη βαθύτερη κατανόηση τους.



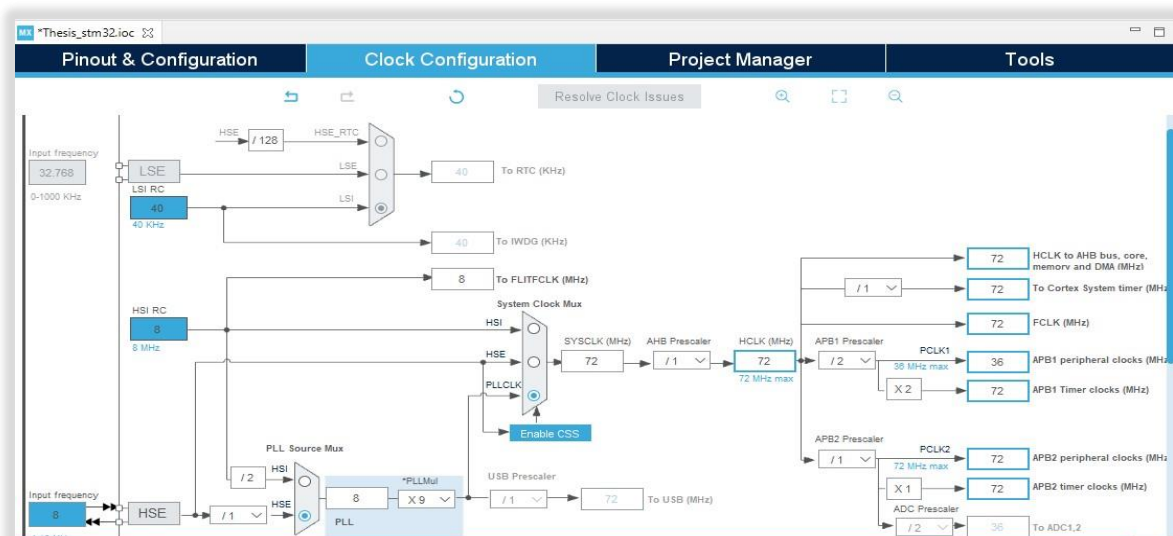
Εικόνα 2.1: Μικροελεγκτής STM32F103C8T6 (BluePill) [55]

Πίνακας 2.1: Βασικά χαρακτηριστικά του μικροελεγκτή

Operating voltage	3.3V
Digital I/O pins	37
External interrupts	PA0-PA13, PB0-PB15, PC13-PC15
DC source/sink from I/O pins	6mA
Flash memory (KB)	64/128
SRAM	20KB
Frequency (clock speed)	72MHz max.
Communication	I2C, SPI, UART, CAN, USB

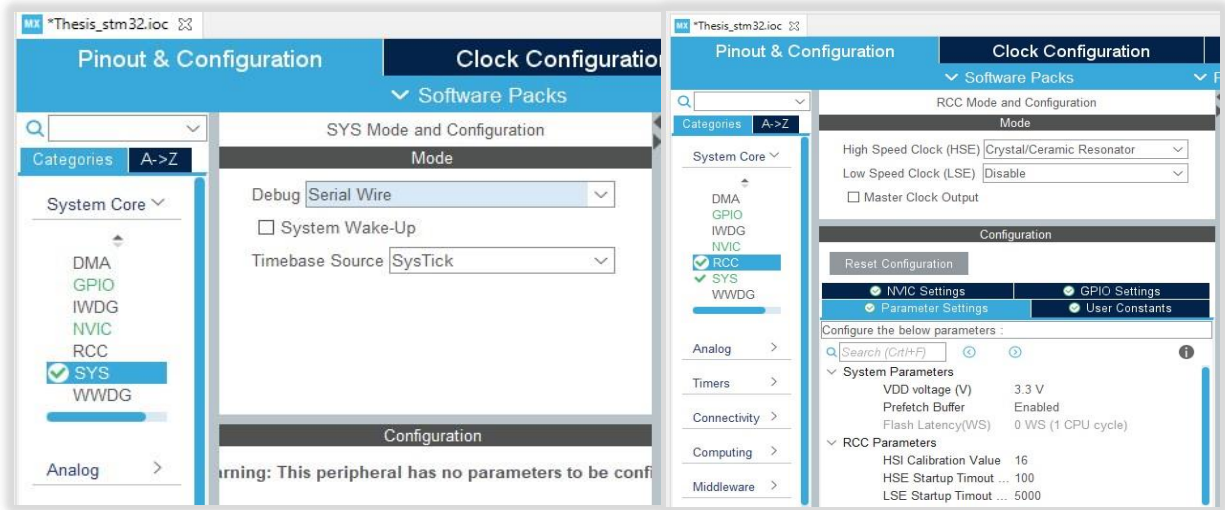


Εικόνα 2.2: Επιλογή του μικροελεγκτή στο πρόγραμμα STM32CubeIDE 1.8.0

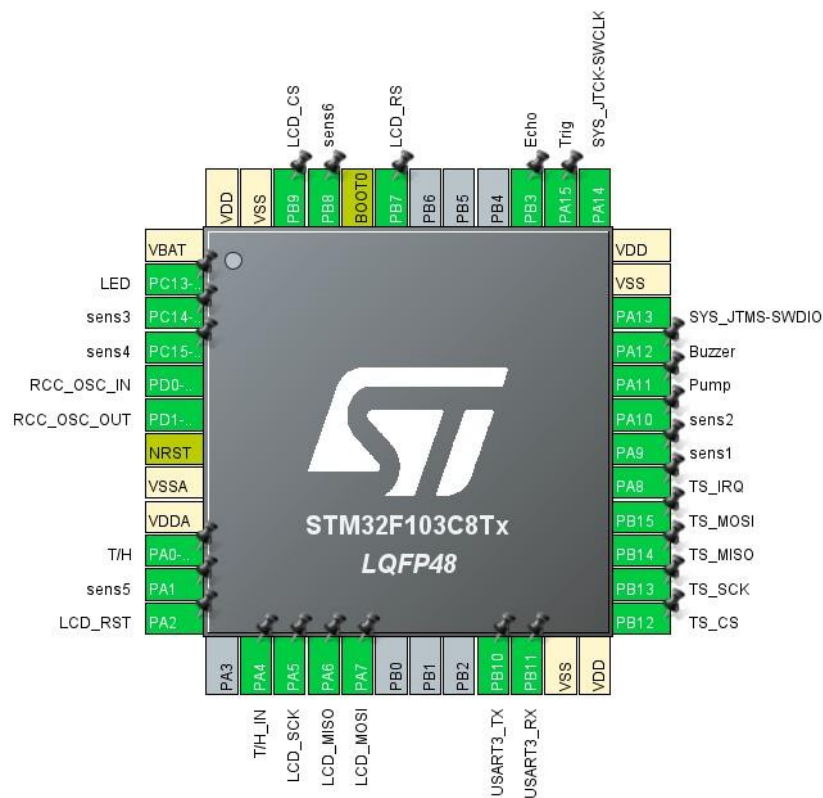


Εικόνα 2.3: Ρύθμιση του Clock στα 72MHz





Εικόνα 2.4: Από αριστερά η ρύθμιση του Debug και δεξιά η ρύθμιση του HSE



Εικόνα 2.5: Δεσμευμένα pins για την υλοποίηση της συσκευής

## Κεφάλαιο 2

Pin Name	Signal o...	GPIO ou...	GPIO mode	GPIO Pull-up/Pull-down	Maximu...	User Label	Modified
PA0-WKUP	n/a	Low	Output Push Pull	No pull-up and no pull-down	Low	T/H	<input checked="" type="checkbox"/>
PA1	n/a	n/a	External Interrupt Mode with Rising edge trigger dete...	Pull-up	n/a	sens5	<input checked="" type="checkbox"/>
PA2	n/a	Low	Output Push Pull	No pull-up and no pull-down	Low	LCD_RST	<input checked="" type="checkbox"/>
PA4	n/a	Low	Output Push Pull	No pull-up and no pull-down	Low	T/H_IN	<input checked="" type="checkbox"/>
PA8	n/a	n/a	Input mode	Pull-up	n/a	TS_IRQ	<input checked="" type="checkbox"/>
PA9	n/a	n/a	External Interrupt Mode with Rising edge trigger dete...	No pull-up and no pull-down	n/a	sens1	<input checked="" type="checkbox"/>
PA10	n/a	n/a	External Interrupt Mode with Rising edge trigger dete...	No pull-up and no pull-down	n/a	sens2	<input checked="" type="checkbox"/>
PA11	n/a	Low	Output Push Pull	No pull-up and no pull-down	Low	Pump	<input checked="" type="checkbox"/>
PA12	n/a	Low	Output Push Pull	No pull-up and no pull-down	Low	Buzzer	<input checked="" type="checkbox"/>
PA15	n/a	Low	Output Push Pull	No pull-up and no pull-down	Low	Trig	<input checked="" type="checkbox"/>
PB7	n/a	Low	Output Push Pull	No pull-up and no pull-down	High	LCD_RS	<input checked="" type="checkbox"/>
PB8	n/a	n/a	External Interrupt Mode with Rising edge trigger dete...	No pull-up and no pull-down	n/a	sens6	<input checked="" type="checkbox"/>
PB9	n/a	High	Output Push Pull	No pull-up and no pull-down	High	LCD_CS	<input checked="" type="checkbox"/>
PB12	n/a	High	Output Push Pull	No pull-up and no pull-down	High	TS_CS	<input checked="" type="checkbox"/>
PC13-TAMP...	n/a	High	Output Push Pull	No pull-up and no pull-down	Low	LED	<input checked="" type="checkbox"/>
PC14-OSC3...	n/a	n/a	External Interrupt Mode with Rising edge trigger dete...	No pull-up and no pull-down	n/a	sens3	<input checked="" type="checkbox"/>
PC15-OSC3...	n/a	n/a	External Interrupt Mode with Rising edge trigger dete...	No pull-up and no pull-down	n/a	sens4	<input checked="" type="checkbox"/>

Εικόνα 2.6: Παραμετροποίηση των pins (1)

NVIC Interrupt Table	Enabled	Preemption Priority	Sub Priority
Non maskable interrupt	<input checked="" type="checkbox"/>	0	0
Hard fault interrupt	<input checked="" type="checkbox"/>	0	0
Memory management fault	<input checked="" type="checkbox"/>	0	0
Prefetch fault, memory access fault	<input checked="" type="checkbox"/>	0	0
Undefined instruction or illegal state	<input checked="" type="checkbox"/>	0	0
System service call via SWI instruction	<input checked="" type="checkbox"/>	0	0
Debug monitor	<input checked="" type="checkbox"/>	0	0
Pendable request for system service	<input checked="" type="checkbox"/>	0	0
Time base: System tick timer	<input checked="" type="checkbox"/>	15	0
PVD interrupt through EXTI line 16	<input type="checkbox"/>	0	0
Flash global interrupt	<input type="checkbox"/>	0	0
RCC global interrupt	<input type="checkbox"/>	0	0
EXTI line1 interrupt	<input checked="" type="checkbox"/>	0	0
DMA1 channel3 global interrupt	<input checked="" type="checkbox"/>	0	0
EXTI line[9:5] interrupts	<input checked="" type="checkbox"/>	0	0
TIM2 global interrupt	<input checked="" type="checkbox"/>	0	0
TIM3 global interrupt	<input checked="" type="checkbox"/>	0	0
TIM4 global interrupt	<input checked="" type="checkbox"/>	0	0
SPI1 global interrupt	<input checked="" type="checkbox"/>	0	0
SPI2 global interrupt	<input type="checkbox"/>	0	0
USART3 global interrupt	<input type="checkbox"/>	0	0
EXTI line[15:10] interrupts	<input checked="" type="checkbox"/>	0	0

Εικόνα 2.7: Παραμετροποίηση των pins (2)

## 2.4 Εξωτερική διακοπή EXTI (External Interrupt)

Όπως αναφέρθηκε στην εισαγωγή του παρόντος κεφαλαίου, η συσκευή προορίζεται ώστε να τοποθετείται στο χώρο του λεβητοστάσιου. Αυτός συνήθως βρίσκεται στο υπόγειο και η πρόσβαση στο χώρο γίνεται κατά κύριο λόγο, από τους διαχειριστές. Αυτό συνεπάγεται με την περιορισμένη πρόσβαση, άρα και τη μη διέλευση των ενοίκων από εκεί. Για το λόγο αυτό, η συσκευή φέρει αισθητήρια, με σκοπό την επιτήρηση του χώρου, ώστε να διασφαλίζεται η ασφάλεια του και η άμεση ενημέρωση για οποιοδήποτε συμβάν. Αυτά είναι, ο αισθητήρας καπνού και τα αισθητήρια για τον έλεγχο της διαρροής δεξαμενής και της πλημμύρας του χώρου. Αυτονόητο δε είναι, πως αυτά με τυχόν ενεργοποίηση τους, πρέπει αμέσως να προηγούνται στη ροή του κώδικα και να εκτελούνται οι εντολές που τα αφορούν. Με γνώμονα αυτό, τα συγκεκριμένα pins των αισθητηρίων αυτών, επιλέχθηκαν να λειτουργούν σε ειδικό καθεστώς, «External Interrupt». Έγινε ρύθμιση σε ανοδικό ή καθοδικό μέτωπο παλμού, ανάλογα τον αισθητήρα, αλλά με ίδια υψηλή προτεραιότητα (Εικόνες 2.5 έως 2.7) [7].

Όταν ενεργοποιηθεί ή απενεργοποιηθεί κάποιος ή κάποιοι αισθητήρες, αυτόματα καλείται η συνάρτηση `void HAL_GPIO_EXTI_Callback(uint16_t GPIO_Pin)`, που σαν όρισμα δέχεται το αντίστοιχο pin, του αισθητήρα. Στην περίπτωση παραπάνω του ενός, η συνάρτηση καλείται με τη σειρά που αυτοί ενεργοποιήθηκαν. Έτσι, γνωρίζοντας το pin, “σηκώνονται” ή “κατεβαίνουν” και οι αντίστοιχες σημαίες (flags), ώστε να λάβουν μέρος στη ροή του κυρίως προγράμματος. Πριν όμως γίνει αυτό, πρέπει να εξακριβωθεί αν η αλλαγή στην είσοδο, προήλθε όντως από τον αισθητήρα, ή ήταν επακόλουθο κάποιου θορύβου (spike). Για το σκοπό αυτό μετά την κλήση της συνάρτησης, γίνεται απενεργοποίηση του αντίστοιχου καναλιού διακοπής (τα κανάλια διακρίνονται στην Εικόνα 2.7). Δημιουργείται μια καθυστέρηση μέσω μιας for loop, καθώς μια while loop ή κάποιο delay δεν μπορούν να συνεργαστούν ομαλά με το External Interrupt [7]. Έπειτα γίνεται η ανάγνωση του αντίστοιχου pin, ώστε να γίνει γνωστό το αρχικό μέτωπο του παλμού διακοπής και η τιμή του (high ή low) μεταφέρεται σε μια μεταβλητή. Μέσα σε μια συνθήκη if, που περιλαμβάνει την τιμή της μεταβλητής ίση με “0” ή “1” αναλόγως, γίνεται και πάλι η ανάγνωση του pin. Με το ίδιο σκεπτικό, ακολουθεί και μια δεύτερη συνθήκη if, χωρίς αυτή τη φορά να γίνεται ανάγνωση. Πλέον θεωρείται δεδομένο, ότι η αλλαγή προήλθε από τον αισθητήρα. Εκεί “σηκώνεται” ή “κατεβαίνει” η αντίστοιχη σημαία (η καθορισμένη μεταβλητή παίρνει τιμή “1” ή “0” αντίστοιχα). Εν συνεχεία της ροής του κώδικα, γίνεται εναλλαγή στο μέτωπο του παλμού, ώστε να είναι έτοιμη η είσοδος και να περιμένει την επόμενη αλλαγή (αν ενεργοποιήθηκε ο αισθητήρας τότε περιμένει την απενεργοποίηση του κ.ο.κ.), ενώ δεν πρέπει να παραληφθεί η εκ νέου ενεργοποίηση του καναλιού.

Με τη διαδικασία που περιεγράφηκε, εξασφαλίζεται ότι η διακοπή προήλθε από κάποιον αισθητήρα και όχι από θόρυβο. Τέλος, στην Εικόνα 2.8 απεικονίζεται το παράδειγμα ενός εκ των αισθητήρων, μιας και όλοι ακολουθούν την ίδια λογική. Η μόνη διαφορά, σχετίζεται με το αρχικό μέτωπο του παλμού σκανδάλης, που διαφέρει ανάλογα τον αισθητήρα.

```

void HAL_GPIO_EXTI_Callback(uint16_t GPIO_Pin)
{
    GPIO_InitTypeDef GPIO_InitStruct = {0};

    /*      Sensor1 Elegxos diarchohs deksamenhs      */

    if (GPIO_Pin == sens1_Pin)
    {
        HAL_NVIC_DisableIRQ(EXTI9_5_IRQn);
        for (int i=0; i<10000; i++) {}
        s1 = HAL_GPIO_ReadPin(GPIOA, GPIO_PIN_9);

        if (s1 == 1)
        {
            s1 = HAL_GPIO_ReadPin(GPIOA, GPIO_PIN_9);

            if (s1 == 1)
            {
                s1_f=1;

                GPIO_InitStruct.Pin = sens1_Pin;
                GPIO_InitStruct.Mode = GPIO_MODE_IT_FALLING;
                HAL_NVIC_EnableIRQ(EXTI9_5_IRQn);
            }
        }

        if (s1 == 0)
        {
            s1 = HAL_GPIO_ReadPin(GPIOA, GPIO_PIN_9);

            if (s1 == 0)
            {
                s1_f=0;

                GPIO_InitStruct.Pin = sens1_Pin;
                GPIO_InitStruct.Mode = GPIO_MODE_IT_RISING;
                HAL_NVIC_EnableIRQ(EXTI9_5_IRQn);
            }
        }
    }

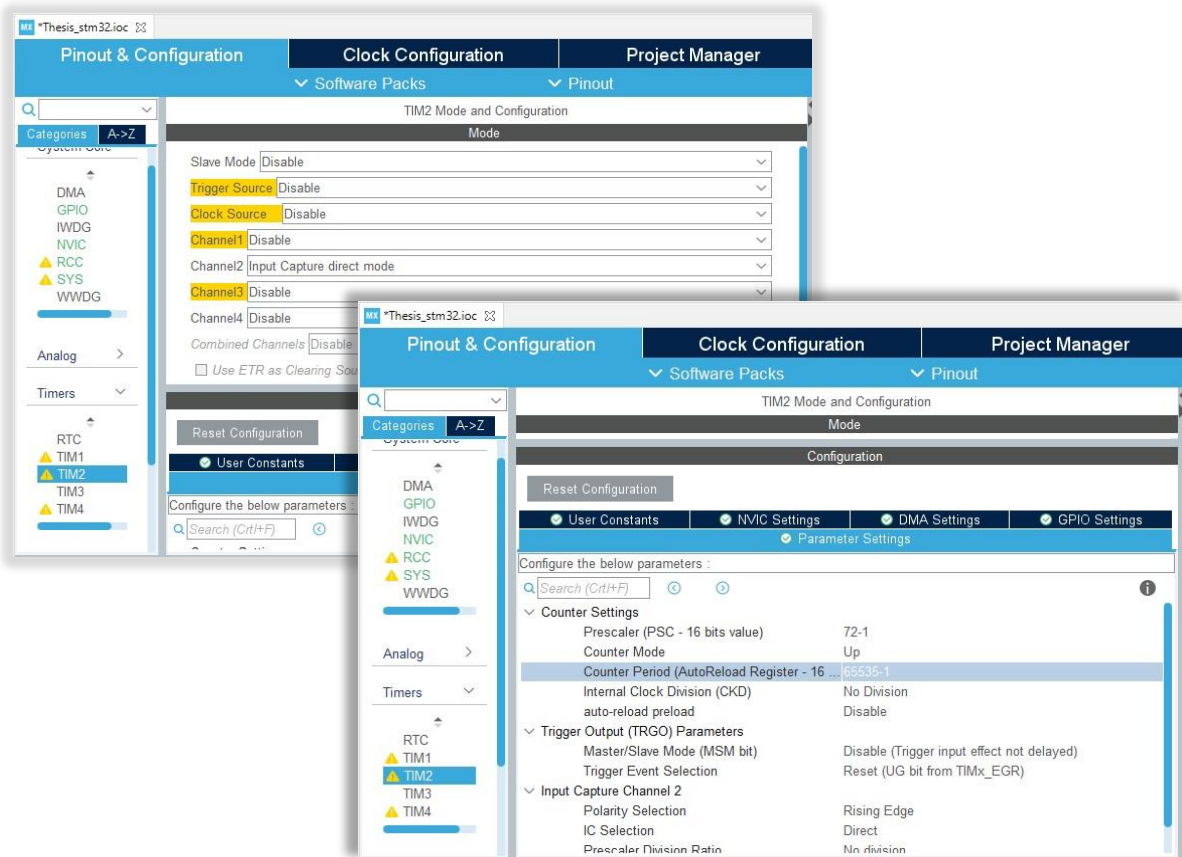
    ///.
    ///.
    ///.
    ///.
}

```

Εικόνα 2.8: Κώδικας χειρισμού εξωτερικής διακοπής από το pin PA9 του καναλιού EXTI line[9:5] interrupt

## 2.5 Δημιουργία καθυστέρησης σε microseconds

Κατά τη διαδικασία του προγραμματισμού, κατέστη αναγκαίο να χρησιμοποιηθεί καθυστέρηση (delay), σε χρόνους της τάξης των microseconds ( $10^{-6}$  sec). Ωστόσο η ελάχιστη καθυστέρηση που υποστηρίζει η βιβλιοθήκη HAL, είναι 1 millisecond ( $10^{-3}$  sec) [7]. Η λύση στο ζήτημα που δημιουργήθηκε, ήρθε μέσα από τη χρήση του timer 2, όπου ρυθμίστηκε κατάλληλα, ώστε ο χρόνος που προκύπτει να είναι σε μsecs (Εικόνες 2.7 και 2.9). Ακολούθησε η κατασκευή μιας συνάρτησης, όπου σαν όρισμα δέχεται τον επιθυμητό χρόνο και αυτός με τη σειρά του ενεργεί μέσα σε μια while loop. Η λούπα επαναλαμβάνεται, όσο ο χρόνος (το όρισμα που δόθηκε) είναι μικρότερο της αυξανόμενης μεταβλητής. Η μεταβλητή προκύπτει από εσωτερικό μετρητή (counter) για τον timer 2, όπου κάθε φορά που καλείται η συνάρτηση, αυτός αρχικά μηδενίζει. Η Εικόνα 2.10 εμφανίζει τη συνάρτηση με τον αντίστοιχο κώδικα.



Εικόνα 2.9: Ρύθμιση του timer 2

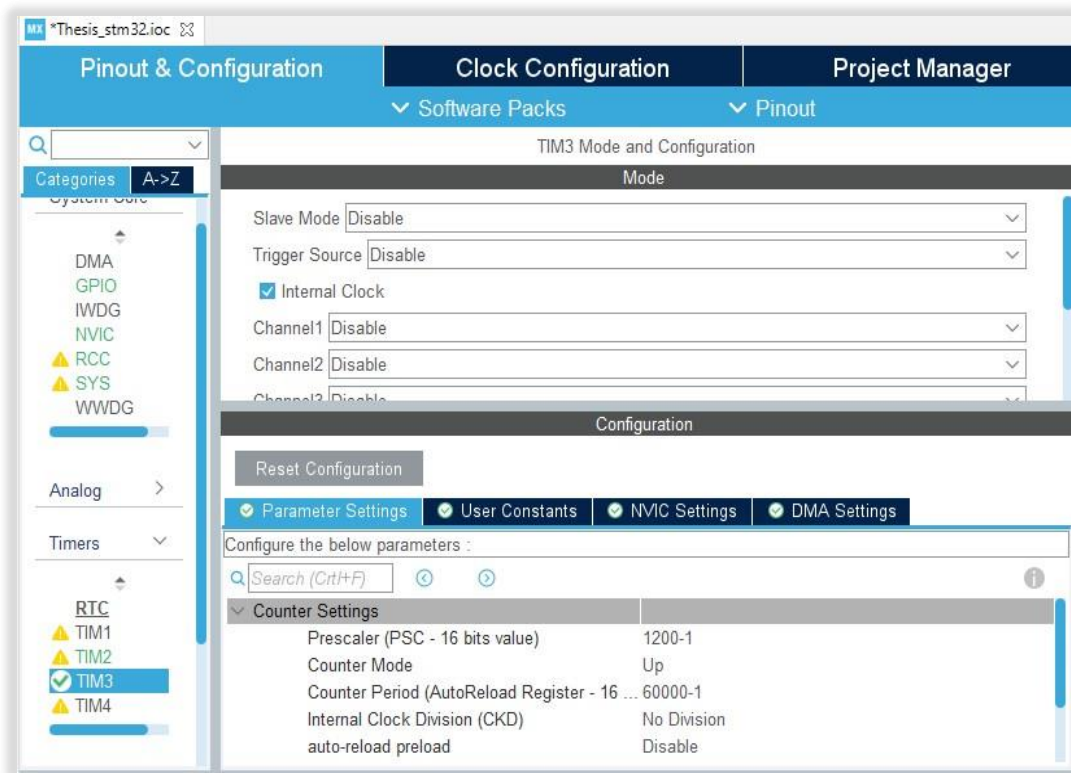
```
void delay (uint16_t time) //microseconds
{
    _HAL_TIM_SET_COUNTER(&htim2, 0);
    while (_HAL_TIM_GET_COUNTER (&htim2) < time);
}
```

Εικόνα 2.10: Συνάρτηση καθυστέρησης σε microsecond

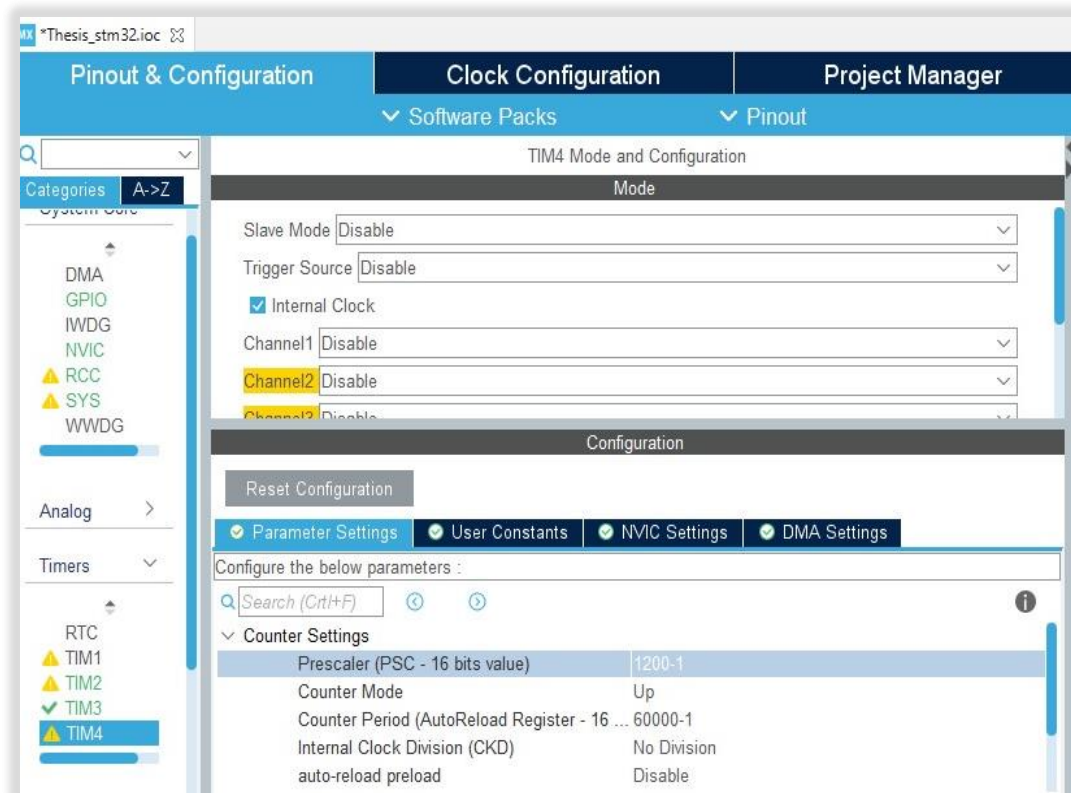
## 2.6 Ρύθμιση timer 3 και timer 4 για χρόνο 1 second

Όπως θα σχολιαστεί και παρακάτω στις αντίστοιχες ενότητες, η συσκευή έχει προγραμματιστεί ώστε να κάνει μέτρηση της στάθμης δεξαμενής, κάθε έξι ώρες. Επιπρόσθετα, η μέτρηση των θερμοκρασιών να πραγματοποιείται κάθε πέντε λεπτά και ακόμα, αν δεν γίνεται κάποια διεργασία, να αποστέλλεται ένα κενό POST, επίσης κάθε πέντε λεπτά. Αυτοί οι χρόνοι προκύπτουν από δυο timers (timer 3 και timer 4) όπου ρυθμίστηκαν για 1 second (Εικόνες 2.7, 2.11 και 2.12). Ο υπολογισμός των τιμών έγινε μέσω της σχέσης (2.1), θέτοντας τον Prescaler 1200 και γνωρίζοντας τη μέγιστη τιμή του ARR ( $2^{16}=65536$ ). Έτσι, κάθε 1 sec γίνεται υπερχειλίση, αυξάνονται οι αντίστοιχες μεταβλητές κατά 1 και σηκώνονται οι αντίστοιχες σημαίες (flags).

$$T = \frac{(\text{Prescaler} - 1) \times (\text{Counter Period} - 1)}{72\text{MHz}} \quad (2.1)$$



Εικόνα 2.11: Ρύθμιση του timer 3



Εικόνα 2.12: Ρύθμιση του timer 4

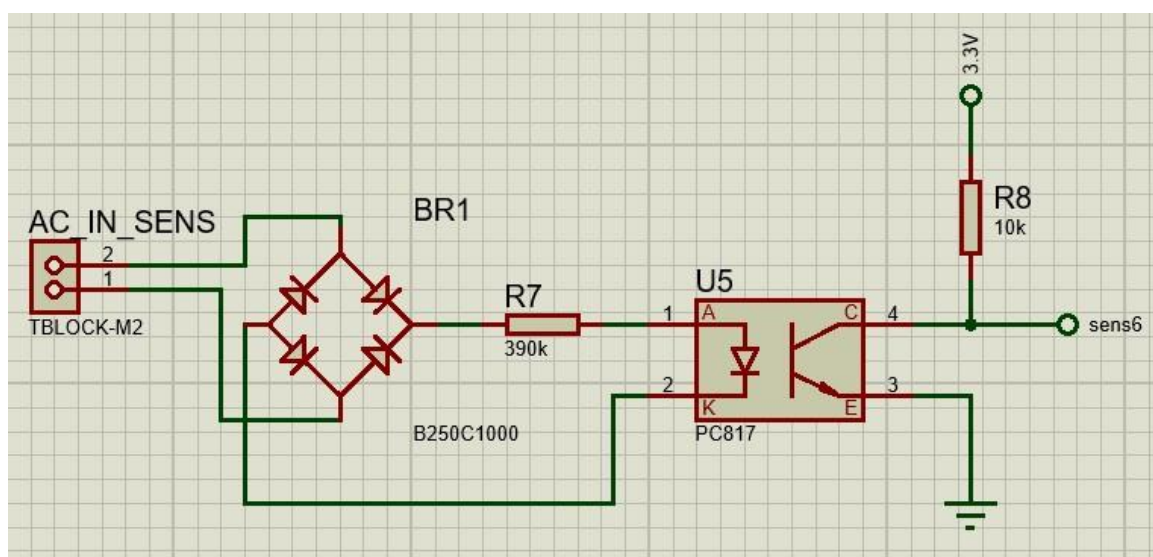
## 2.7 Κύκλωμα ανίχνευσης ενεργοποίησης-απενεργοποίησης του λέβητα

Το κύκλωμα του Σχήματος (2.2), είναι υπεύθυνο για την παρακολούθηση της ενεργοποίησης και απενεργοποίησης του λέβητα. Μέσω καλωδίων, η κλέμα (AC\_IN\_SENS) συνδέεται παράλληλα με την τροφοδοσία του λέβητα, για την επιτήρηση του. Όταν αυτός ενεργοποιηθεί, η ίδια τάση μέσω της κλέμας θα φτάσει στην γέφυρα, όπου θα γίνει η ανόρθωση της. Στη συνέχεια η συνεχής τάση, θα πολώσει ορθά τη δίοδο (IR-Led) του οπτοζεύκτη και μέσω του φωτός θα πολωθεί η βάση του φωτοτρανζίστορ. Όσο η δίοδος IR-Led άγει, η έξοδος του οπτοζεύκτη (ο συλλέκτης του φωτοτρανζίστορ) θα είναι σε χαμηλό δυναμικό. Όταν σταματήσει να άγει, δηλαδή θα σταματήσει να λειτουργεί ο λέβητας, θα έρθει σε υψηλό δυναμικό στα 3.3V DC [4-5],[8]. Η αντίσταση R7 χρησιμοποιείται για να περιορίσει το ρεύμα του IR-Led, ενώ η αντίσταση R8 είναι pull-up και χρησιμοποιείται για να παρέχει ομαλά το σήμα. Η είσοδος του μικροελεγκτή που δέχεται το σήμα, είναι το pin PB8 (sens6) και έχει ρυθμιστεί σε External Interrupt.

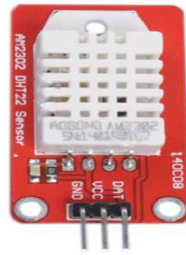
## 2.8 Αισθητήριο θερμοκρασίας-σχετικής υγρασίας DHT22

Στην πλακέτα της συσκευής, κουμπώνουν δυο ψηφιακής εξόδου αισθητήρια θερμοκρασίας-σχετικής υγρασίας DHT22 (Εικόνα 2.13). Για τη μέτρηση της θερμοκρασίας (T -40°C~80°C), χρησιμοποιείται ένα θερμίστορ, ενώ η μέτρηση της σχετικής υγρασίας (RH 0~100 %), γίνεται με χωρητικό αισθητήριο [9]. Η πλακέτα στην οποία είναι τοποθετημένα, έχει τρεις ακροδέκτες, ένα για τη γείωση, ένα για την τροφοδοσία (3.3V DC) και ένα για τα δεδομένα. Το αισθητήριο για τη μέτρηση της εξωτερικής θερμοκρασίας, συνδέεται στο pin PA0 (T/H) και το δεύτερο για την εσωτερική θερμοκρασία, στο pin PA4 (T/H\_IN). Η συνδεσμολογία τους προσφέρεται στο Σχήμα (2.3). Τα δυο pins λειτουργούν ως ψηφιακή είσοδος-έξοδος. Όταν απαιτείται η εναλλαγή για τις ανάγκες λήψης μετρήσεων, αυτή γίνεται με τις αντίστοιχες εντολές, μέσω του κώδικα.

Η διαδικασία κάθε μέτρησης, απαιτεί την ενεργοποίηση του αισθητήρα. Αυτή γίνεται, θέτοντας ως έξοδο το pin των δεδομένων και γειώνοντας το, για χρόνο μεγαλύτερο του 1 millisecond, ώστε ο αισθητήρας να ανιχνεύσει το σήμα, από το μικροελεγκτή. Εν συνέχεια, τροφοδοτώντας το ίδιο pin με τάση (η εναλλαγή γίνεται μέσω του προγράμματος) για 20 - 40 microseconds, δίνει στον αισθητήρα να



Σχήμα 2.2: Κύκλωμα ανίχνευσης ενεργοποίησης και απενεργοποίησης του λέβητα



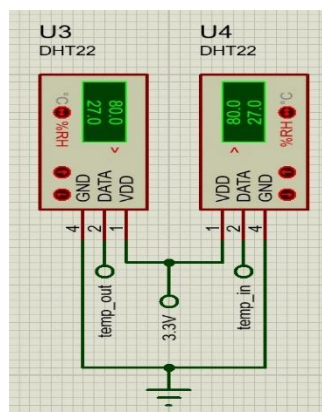
Εικόνα 2.13: Αισθητήριο θερμοκρασίας DHT22

καταλάβει ότι είναι η αρχή της μέτρησης. Με την ολοκλήρωση της διαδικασίας αρχικοποίησης, το pin των δεδομένων ρυθμίζεται εκ νέου σε είσοδο, περιμένοντας την απόκριση του αισθητήρα.

Για το σκοπό αυτό, γίνεται μια καθυστέρηση 40 μsecs ώστε να μεταβεί από το high στο low. Έπειτα, μια ακόμα καθυστέρηση 80 μsecs και ένας εκ νέου έλεγχος επιβεβαίωσης, ότι έχει μεταβεί. Αν βρίσκεται σε λογικό μηδέν, σημαίνει ότι ο αισθητήρας ανταποκρίθηκε, καθώς η απάντηση του είναι το χαμηλό δυναμικό, με διάρκεια 80 μsecs. Η διαδικασία αυτή διαρκεί συνολικά 120 μsecs και με το πέρας της, ο ακροδέκτης επανέρχεται σε υψηλό δυναμικό για 80μsecs, με σκοπό να σηματοδοτήσει την επιστροφή των δεδομένων της θερμοκρασίας και υγρασίας, στον μικροελεγκτή.

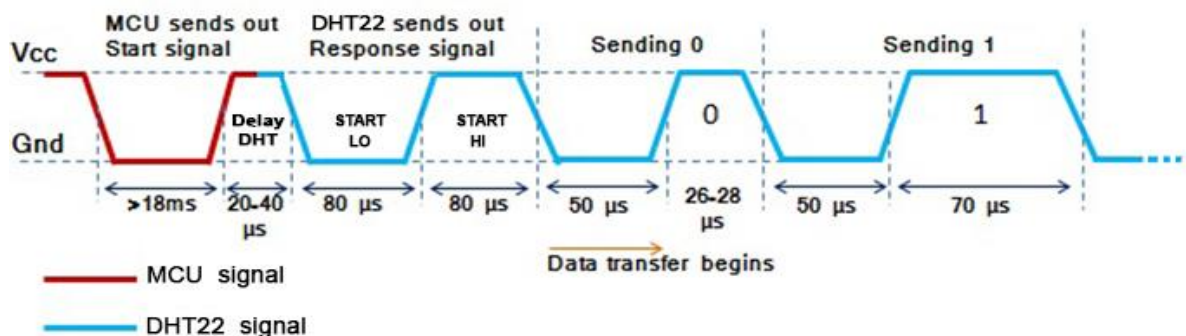
Θα αποστείλει συνολικά 40 bits, που αποτελούν το άθροισμα της σχέσης (2.2). Κάθε bit ξεκινάει με χαμηλό δυναμικό για 50 μsecs, κι ακολουθεί υψηλό δυναμικό μεταβαλλόμενης διάρκειας. Αν η διάρκεια κυμαίνεται μεταξύ 26-28 μsecs, αυτό αντικατοπτρίζει το λογικό “0”. Αντίθετα, αν η διάρκεια είναι περίπου 70 μsecs, αυτό μεταφράζεται σε λογικό “1”. Αυτά καταλαμβάνουν τα πρώτα 32 bits από τα συνολικά 40. Τα τελευταία 8 συμπληρώνουν το άθροισμα των 40 bits και προγραμματιστικά βοηθούν να ελέγχεται, για το αν η μετάδοση έγινε σωστά. Ένα άθροισμα δεδομένων μικρότερο του 40, οδηγεί στο συμπέρασμα της λανθασμένης μεταφοράς των δεδομένων, από τον αισθητήρα στον μικροελεγκτή [10]. Όλα όσα αναλύθηκαν σε αυτή την ενότητα, απεικονίζονται για την καλύτερη κατανόηση, μέσω της Εικόνας 2.14.

$$\Delta\epsilon\delta\omicron\mu\epsilon\nu\alpha = \left( \begin{array}{l} 8\ bit\ \acute{\alpha}\kappa\epsilon\rho\alpha\iota\alpha\ \delta\epsilon\delta\omicron\mu\epsilon\nu\alpha\ RH + 8\ bit\ \delta\epsilon\kappa\alpha\delta\iota\kappa\acute{\alpha}\ \delta\epsilon\delta\omicron\mu\epsilon\nu\alpha\ RH \\ + \\ 8\ bit\ \acute{\alpha}\kappa\epsilon\rho\alpha\iota\alpha\ \delta\epsilon\delta\omicron\mu\epsilon\nu\alpha\ T + 8\ bit\ \delta\epsilon\kappa\alpha\delta\iota\kappa\acute{\alpha}\ \delta\epsilon\delta\omicron\mu\epsilon\nu\alpha\ T \\ + 8\ bit\ \acute{\alpha}\theta\rho\omicron\iota\sigma\mu\alpha\ \epsilon\lambda\acute{\epsilon}\gamma\chi\omicron\upsilon \end{array} \right) \quad (2.2)$$



Σχήμα 2.3: Συνδεσμολογία αισθητήρων DHT22





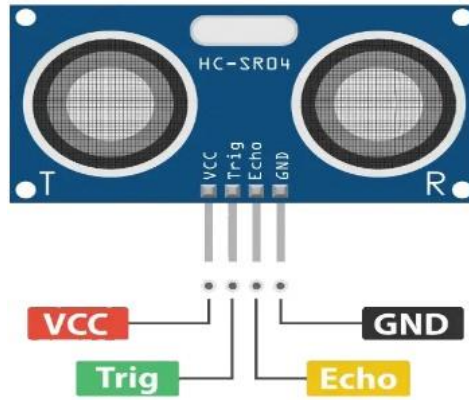
Εικόνα 2.14: Οπτική απεικόνιση λειτουργίας του αισθητηρίου DHT22

## 2.9 Αισθητήριο μέτρησης απόστασης HC-SR04

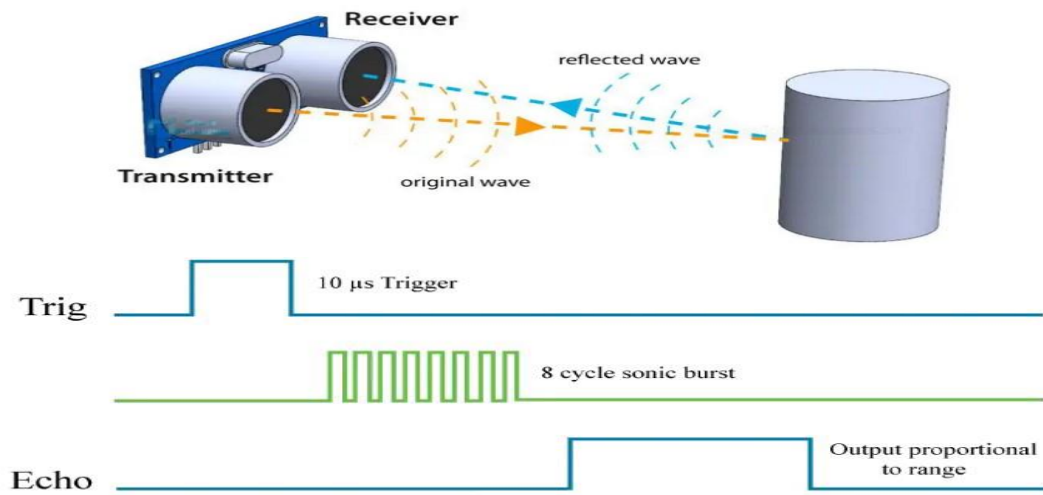
Για τις ανάγκες μέτρησης της στάθμης δεξαμενής, μιας και αυτό θα γινόταν σε επίπεδο προσομοίωσης, χρησιμοποιήθηκε ο αισθητήρας υπερήχων HC-SR04 (Εικόνα 2.15). Η πλακέτα του αποτελείται από τέσσερις ακροδέκτες. Ο πρώτος και ο τέταρτος είναι για την τροφοδοσία (5V) και τη γείωση αντίστοιχα. Ο τρίτος ακροδέκτης είναι το TRIG, όπου συνδέεται στο pin PA15 (Trig) και ρυθμίστηκε σε ψηφιακή έξοδο. Ο τέταρτος είναι το ECHO, όπου συνδέεται στο pin PB3 (Echo) και ρυθμίστηκε μέσω του timer 2 σε Input Capture direct mode (Εικόνα 2.9). Η συνδεσμολογία του προσφέρεται στο Σχήμα (2.4). Η πλακέτα στην οποία συγκροτείται φέρει ακόμα, ένα πομπό, ένα δέκτη, ένα κρύσταλλο 4MHz και κάποια κυκλώματα υποστήριξης και ελέγχου. Έχει τη δυνατότητα μέτρησης από 2cm – 400cm με ακρίβεια περίπου τα 3mm. Η αρχή λειτουργίας του, βασίζεται στο φαινόμενο της ανάκλασης και η απόσταση υπολογίζεται χρησιμοποιώντας, το χρόνο που το κύμα ταξίδεψε και την ταχύτητα διάδοσης του ήχου στον αέρα (Εικόνα 2.16). Η τελική απόσταση σε εκατοστά (cm), υπολογίζεται από τη σχέση (2.3) [4],[11].

Αρχικά εφαρμόζεται υψηλό δυναμικό στην ακίδα TRIG, για χρόνο μεγαλύτερο των 10 μsecs. Τότε ο πομπός, θα εκπέμψει οκτώ κύκλους υπερήχων στα 40kHz και ο δέκτης θα αναμένει να ανιχνεύσει, αν υπάρχει επιστρεφόμενο σήμα. Αν υπάρχει κάτι προς μέτρηση τότε, στην ακίδα ECHO θα δημιουργηθεί παλμός υψηλού δυναμικού, ανάλογου της απόστασης (Εικόνα 2.16). Για να μπορέσει να γίνει ανάγνωση της χρονικής διάρκειας του παλμού, το pin του ECHO (PB3) ρυθμίστηκε κατάλληλα, ώστε να αναγνωρίζει την άνοδο και την κάθοδο του παλμού. Αρχικά ρυθμίζεται να αναμένει την άνοδο και όταν αυτή ανιχνευθεί, καταγράφεται η συγκεκριμένη χρονική στιγμή (timestamp) σε μια μεταβλητή. Στη συνέχεια μέσω του προγράμματος, ρυθμίζεται εκ νέου ώστε να γίνει η εναλλαγή μετώπου και να αναμένει για την κάθοδο του παλμού. Όπως και πριν, όταν ανιχνευθεί η κάθοδος, θα καταγραφεί η συγκεκριμένη χρονική στιγμή (timestamp) σε μια δεύτερη μεταβλητή. Η διαφορά της δεύτερης μεταβλητής από την πρώτη, έχει ως αποτέλεσμα να δώσει τον επιθυμητό χρόνο σε μsecs, όπου απαιτείται για την εύρεση της απόστασης. Τέλος, δεν πρέπει να παραληφθεί η εκ νέου εναλλαγή, ώστε το pin PB3 να αναμένει το θετικό μέτωπο, του επόμενου παλμού (της επόμενης μέτρησης).

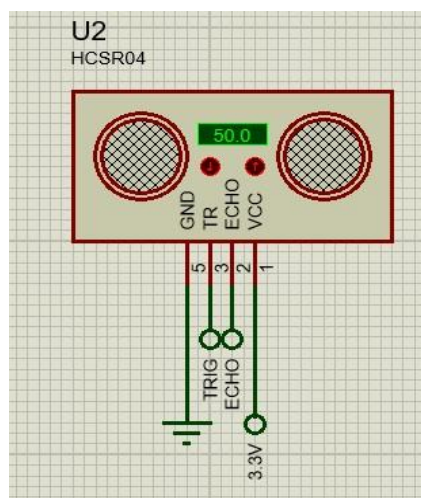
$$\text{Απόσταση(cm)} = \frac{\text{ECHO}(\mu\text{sec}) \times 0.034}{2} \quad (2.3)$$



Εικόνα 2.15: Αισθητήρας υπερήχων HC-SR04 [56]



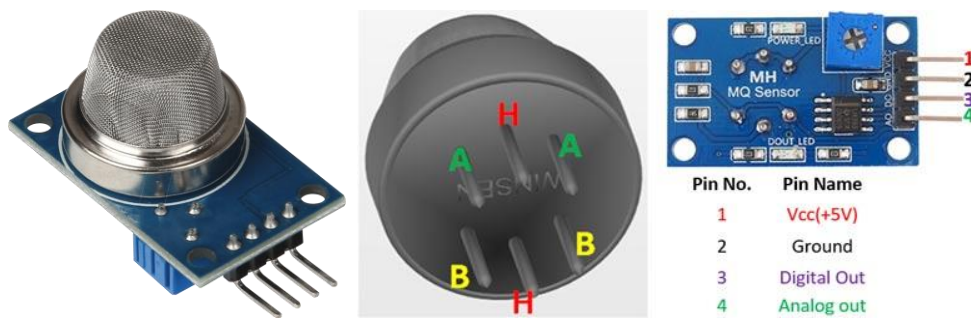
Εικόνα 2.16: Αναπαράσταση εκπομπής και ανάκλασης του σήματος [56]



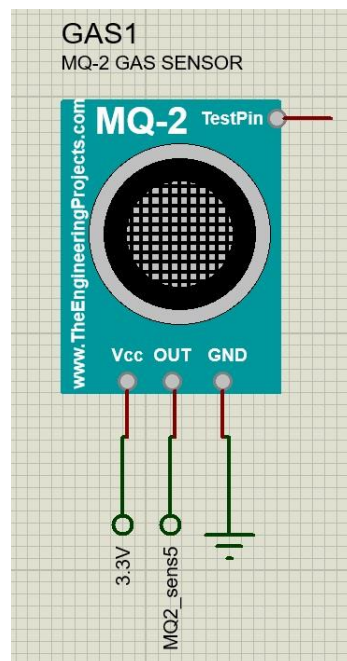
Σχήμα 2.4: Συνδεσμολογία αισθητηρίου HC-SR04

## 2.10 Αισθητήριο καπνού MQ-2

Το αισθητήριο καπνού που χρησιμοποιείται είναι το MQ-2 (Εικόνα 2.17). Η πλακέτα που το υποστηρίζει έχει τέσσερις ακίδες. Οι πρώτες δύο για την τροφοδοσία (3.3V DC) και τη γείωση, ενώ οι επόμενες δύο είναι οι έξοδοι (μια ψηφιακή και μια αναλογική), για την ανάγνωση των δεδομένων του αισθητήρα [12]. Η συνδεσμολογία προσφέρεται στο Σχήμα (2.5). Η αναλογική έξοδος χρειάζεται ADC μετατροπέα, ώστε να γίνεται η αντιστοιχία του ποσού ανίχνευσης καπνού, αν αυτό απαιτείται. Αντ' αυτού, η ψηφιακή έξοδος χρησιμοποιεί ένα συγκριτή τάσης. Έτσι, μεταφράζεται η ανίχνευση της ποσότητας καπνού σε τάση και συγκρίνεται με την τάση αναφοράς, που ρυθμίζεται μικρομετρικά από το ποτενσιόμετρο. Στην κατασκευή του συστήματος, χρησιμοποιείται η ψηφιακή έξοδος και συνδέεται στο pin PA1 (sens5) του μικροελεγκτή, όπου έχει ρυθμιστεί ως External Interrupt.



Εικόνα 2.17: Αισθητήρας καπνού MQ-2 [57]

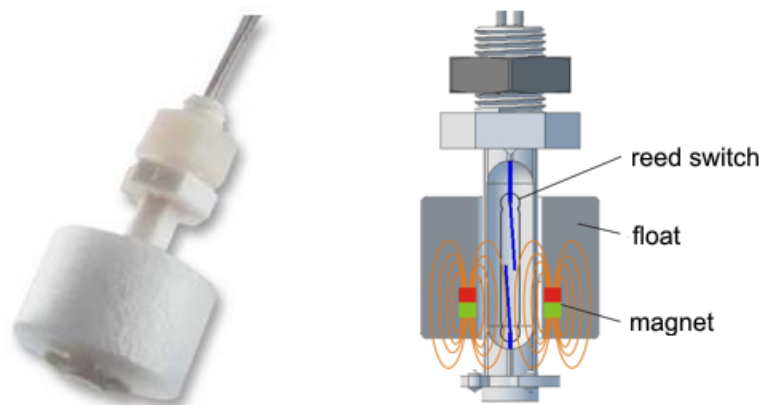


Σχήμα 2.5: Συνδεσμολογία αισθητήριου MQ-2

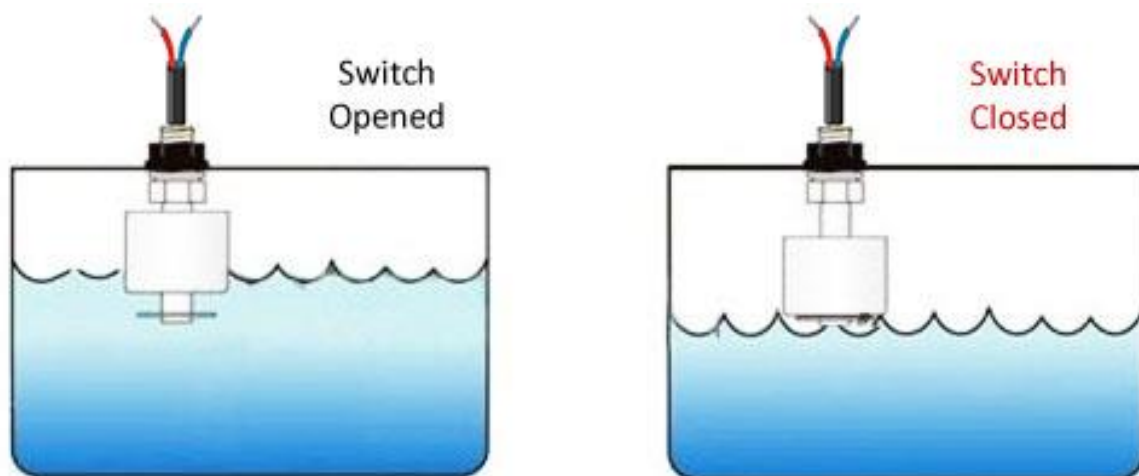
### 2.11 Αισθητήριο ελέγχου στάθμης (magnetic float switch)

Πάνω στη συσκευή τερματίζουν τέσσερα αισθητήρια ελέγχου στάθμης (magnetic float switch), σαν αυτό της Εικόνας 2.18. Τα δεσμευμένα pins γι' αυτά είναι, το pin PA9 (sens1), το pin PA10 (sens2), το pin PC14 (sens3) και το pin PC15 (sens4). Τα pins αυτά, ρυθμίστηκαν σε External Interrupt και η συνδεσμολογία τους προσφέρεται στο Σχήμα (2.6). Εκεί διακρίνονται οι pull-up αντιστάσεις των 10k, καθώς και η τροφοδοσία των αισθητηρίων, που γίνεται με συνεχή τάση 3.3V.

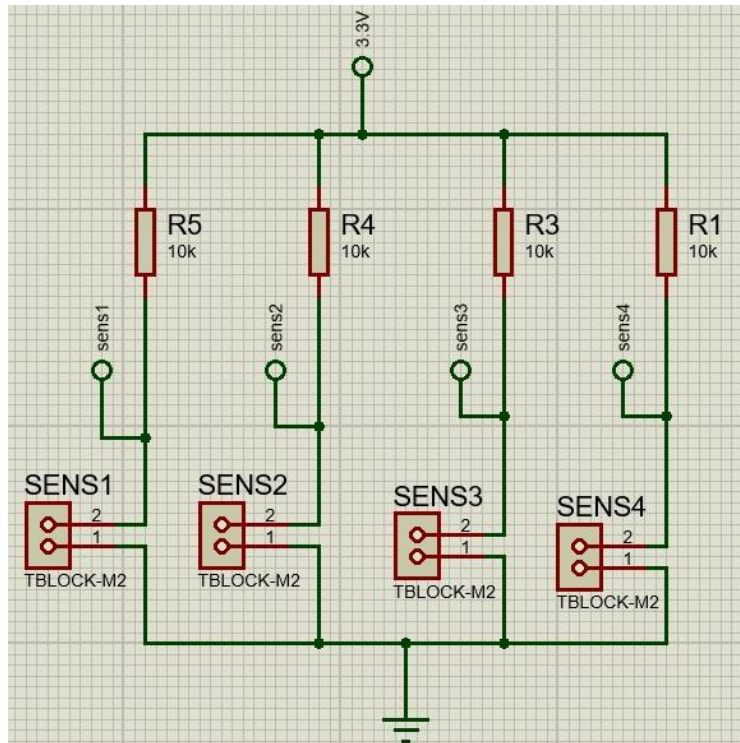
Το αισθητήριο στο εσωτερικό του σταθερού κεντρικού του άξονα, περιέχει ένα μαγνητικό διακόπτη. Πάνω σε αυτόν τον άξονα, τοποθετείται ένα στρογγυλό πλαστικό, με διαμπερή τρύπα στο κέντρο του. Περιμετρικά στην κάτω πλευρά του, υπάρχει εμφυτευμένος ένας μαγνήτης. Είναι σχεδιασμένο έτσι ώστε, να μπορεί να κινείται ελεύθερα πάνω-κάτω και εντός των ορίων, που του επιβάλλουν δυο τερματικές ασφάλειες. Αυτό έχει ως αποτέλεσμα, στην αρχική του θέση, ο διακόπτης να είναι κλειστός. Όταν το πλαστικό κινηθεί προς τα πάνω, παύει να μαγνητίζεται ο διακόπτης και ανοίγει. Οι Εικόνες 2.18 και 2.19 αναδεικνύουν με τον πλέον κατάλληλο τρόπο, το εσωτερικό του αισθητηρίου καθώς και τον τρόπο λειτουργίας του [13].



Εικόνα 2.18: Μαγνητικό αισθητήριο ελέγχου στάθμης (magnetic float switch) [13]



Εικόνα 2.19: Λειτουργία μαγνητικού αισθητηρίου ελέγχου στάθμης



Σχήμα 2.6: Συνδεσμολογία μαγνητικών αισθητήρων ελέγχου στάθμης

## 2.12 Αντλία

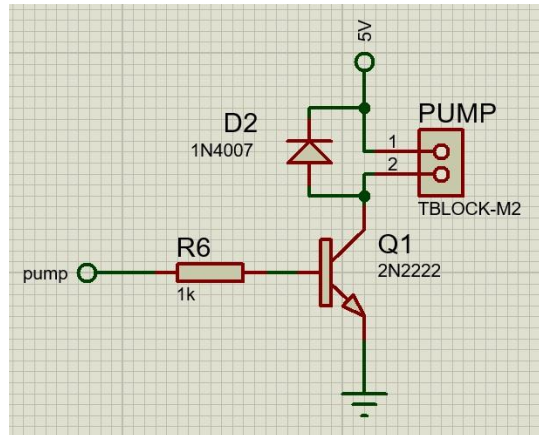
Η αντλία που χρησιμοποιήθηκε για λόγους προσομοίωσης, είναι η micro Motor Pump της Εικόνας 2.20. Λειτουργεί στην περιοχή 2.5-6V συνεχής τάσης και έχει ρυθμό ροής 80-120L/H [14]. Δέχεται εντολή εκκίνησης και παύσης από το pin PA11 (Pump) του μικροελεγκτή, όπου έχει ρυθμιστεί σε ψηφιακή έξοδο. Το κύκλωμα λειτουργίας του προσφέρεται στο Σχήμα (2.7). Εκεί διακρίνεται η τροφοδοσία που είναι στα 5V και ο έλεγχος της, που γίνεται μέσω του τρανζίστορ. Όταν αυτό πολωθεί ορθά, εκκινεί η αντλία. Παράλληλα με την αντλία, υπάρχει και μια ανάστροφα πολωμένη δίοδος, για την προστασία του κυκλώματος, ενώ η αντίσταση στη βάση του τρανζίστορ, σκοπό έχει την προστασία αυτού.

## 2.13 Buzzer

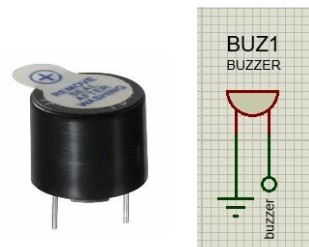
Για ηχητική προειδοποίηση, έχει προστεθεί ένα buzzer πλακέτας που το ένα άκρο συνδέεται στη γείωση, ενώ το άλλο στο pin PA12 (Buzzer) που έχει ρυθμιστεί, σε ψηφιακή έξοδο. Στην Εικόνα 2.21 παρουσιάζεται το buzzer και η συνδεσμολογία του.



Εικόνα 2.20: Αντλία



Σχήμα 2.7: Κύκλωμα αντλίας



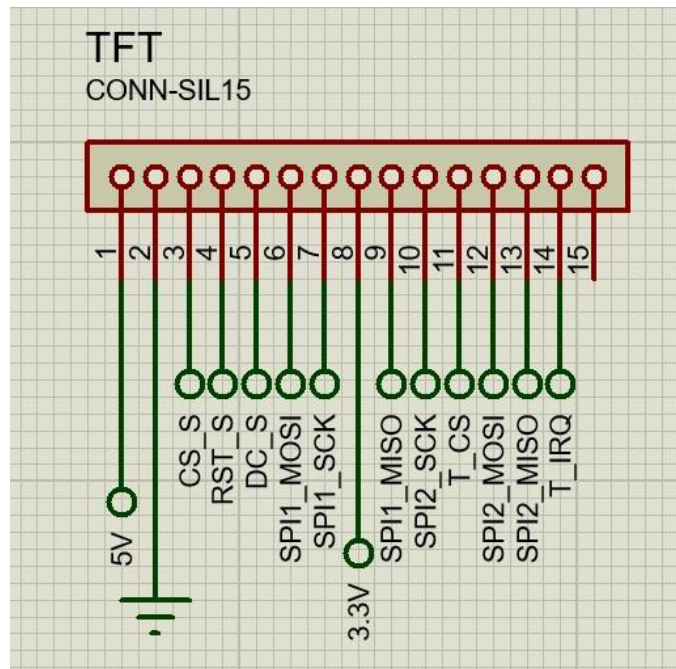
Εικόνα 2.21: Buzzer πλακέτας

## 2.14 Οθόνη αφής TFT (2.8 ιντσών)

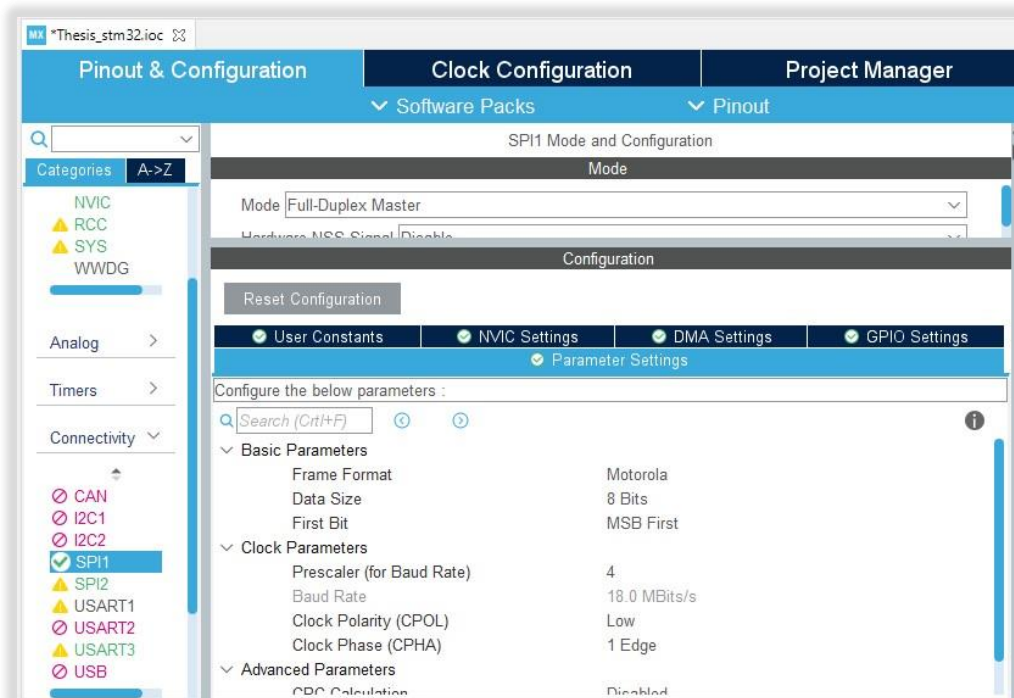
Για την επιλογή των λειτουργιών και του χειρισμού της συσκευής, χρησιμοποιήθηκε η οθόνη αφής της Εικόνας 2.22. Τροφοδοτείται με συνεχή τάση 5V και χρησιμοποιεί το πρωτόκολλο επικοινωνίας SPI [15]. Τα δεκατέσσερα δεσμευμένα pins και η σύνδεση τους με την οθόνη προσφέρονται στο Σχήμα (2.8). Στο μικροελεγκτή, η επικοινωνία μέσω του SPI1 δεσμεύτηκε για την οθόνη, ενώ η SPI2 για την λειτουργία της αφής. Οι παραμετροποιήσεις αυτές, προβάλλονται στις Εικόνες 2.23 έως 2.25.



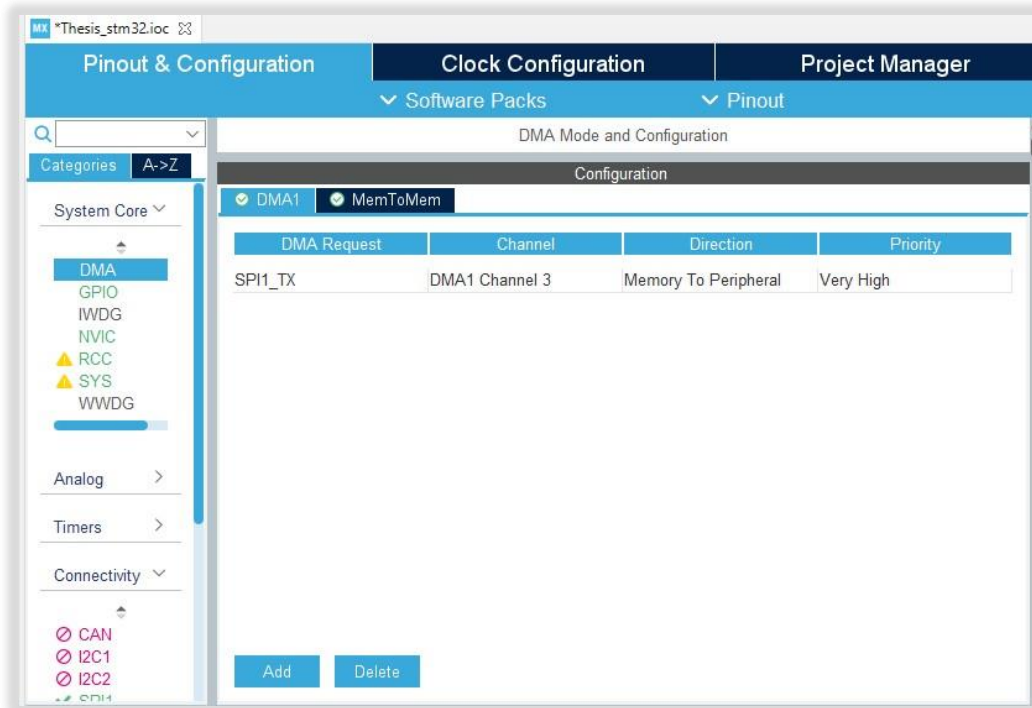
Εικόνα 2.22: Οθόνη αφής 2.8 ιντσών [58]



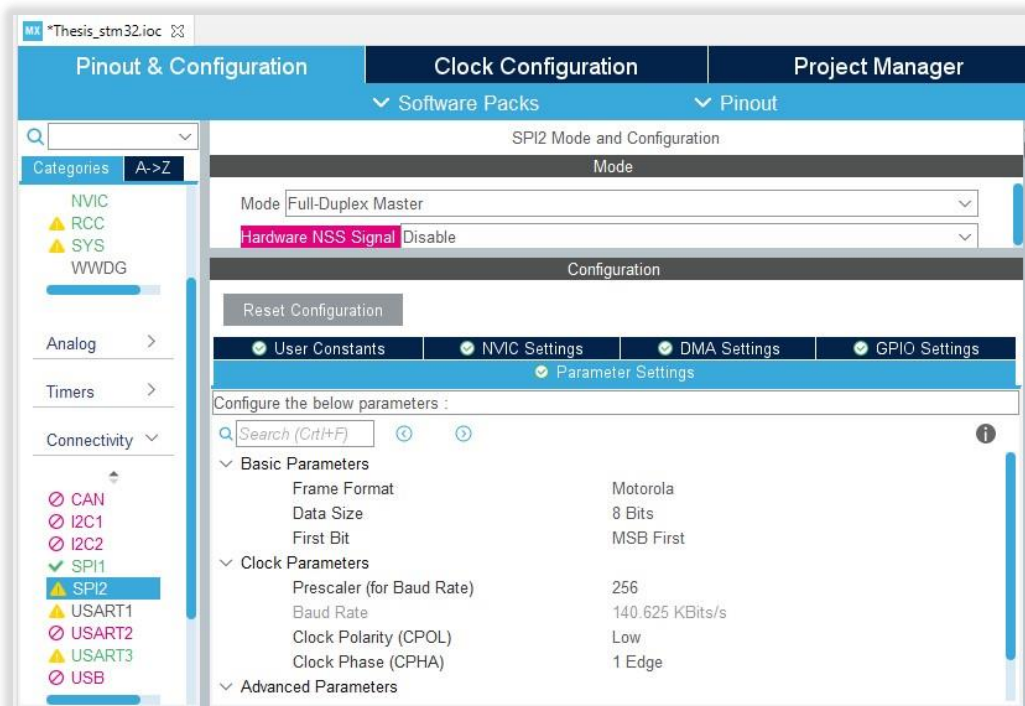
Σχήμα 2.8: Συνδεσμολογία οθόνης αφής



Εικόνα 2.23: Ρυθμίσεις SPI1



Εικόνα 2.24: Ρύθμιση DMA για το SPI1

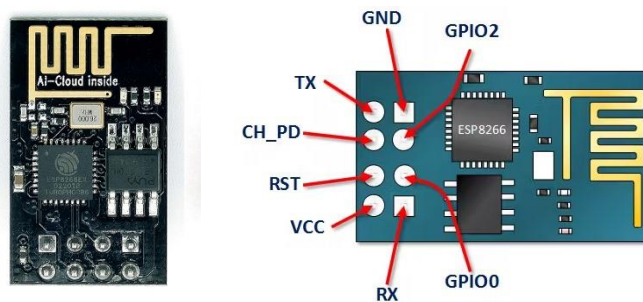


Εικόνα 2.25: Ρυθμίσεις SPI2

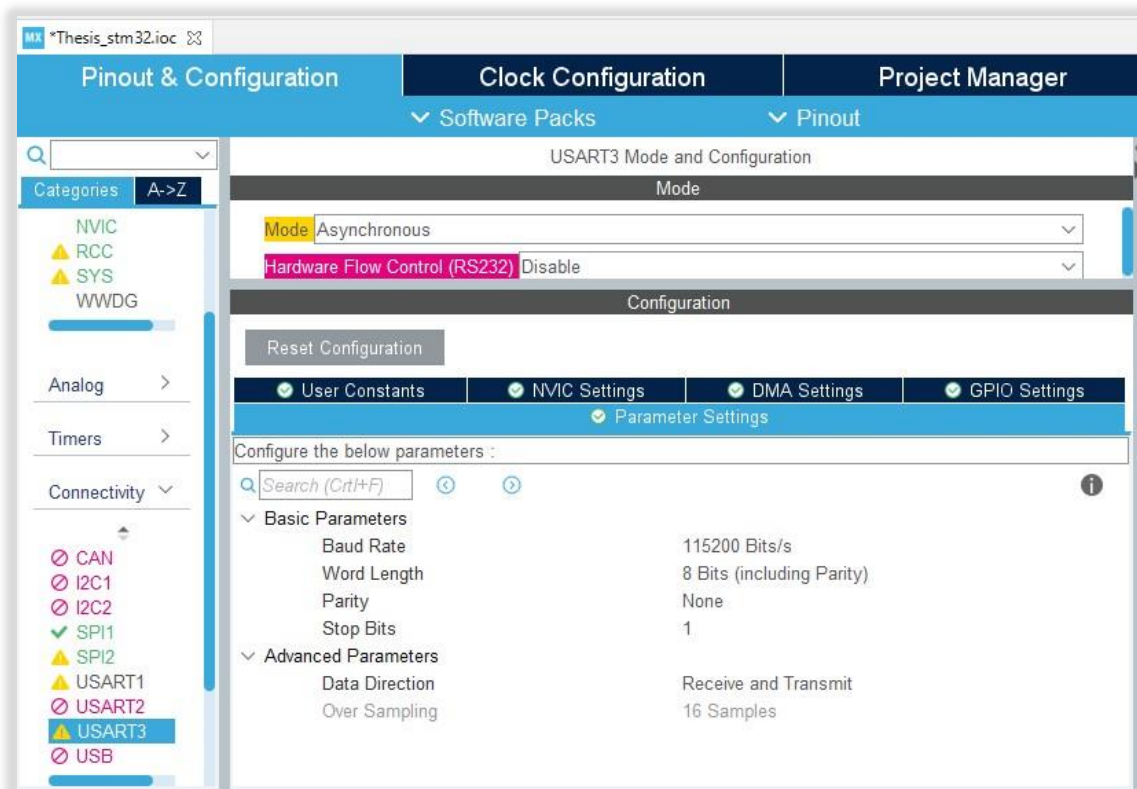


## 2.15 ESP-01

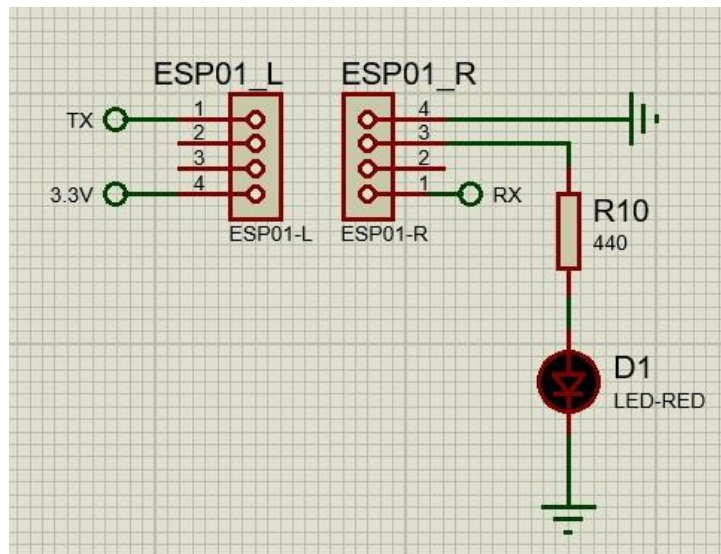
Για την αποστολή των δεδομένων στο server, χρησιμοποιείται το ESP-01 (Εικόνα 2.26), που τροφοδοτείται με 3.3V DC και επικοινωνεί σειριακά με τον μικροελεγκτή. Η σύνδεση γίνεται μέσω του USART3 και των pins PB10 (USART3\_TX) και PB11 (USART3\_RX), ενώ η παραμετροποίηση τους αποτυπώνεται στην Εικόνα 2.27. Τα δεδομένα μεταφέρονται μέσω της εντολής *sprintf* (Εικόνα 3.15), καθώς απαιτείται αυτά να είναι σε κώδικα ASCII. Επίσης, στο GPIO2 του ESP-01 έχει τοποθετηθεί ένα led σε σειρά με μια αντίσταση (για την προστασία του led). Όταν ενεργοποιείται αναβοσβήνει, παρέχοντας οπτική ένδειξη της απώλειας σύνδεσης στο δίκτυο. Η συνδεσμολογία του προσφέρεται στο Σχήμα (2.9). Ο προγραμματισμός του έγινε μέσω της πλατφόρμας Arduino IDE 2.1.0, ώστε να δέχεται τα δεδομένα, να τα επεξεργάζεται κι έπειτα να τα αποστέλλει σε μορφή JSON, με τη μέθοδο POST. Για το σκοπό αυτό, χρησιμοποιήθηκε η βιβλιοθήκη ArduinoJSON Version 6 [16].



Εικόνα 2.26: ESP-01 [59-60]



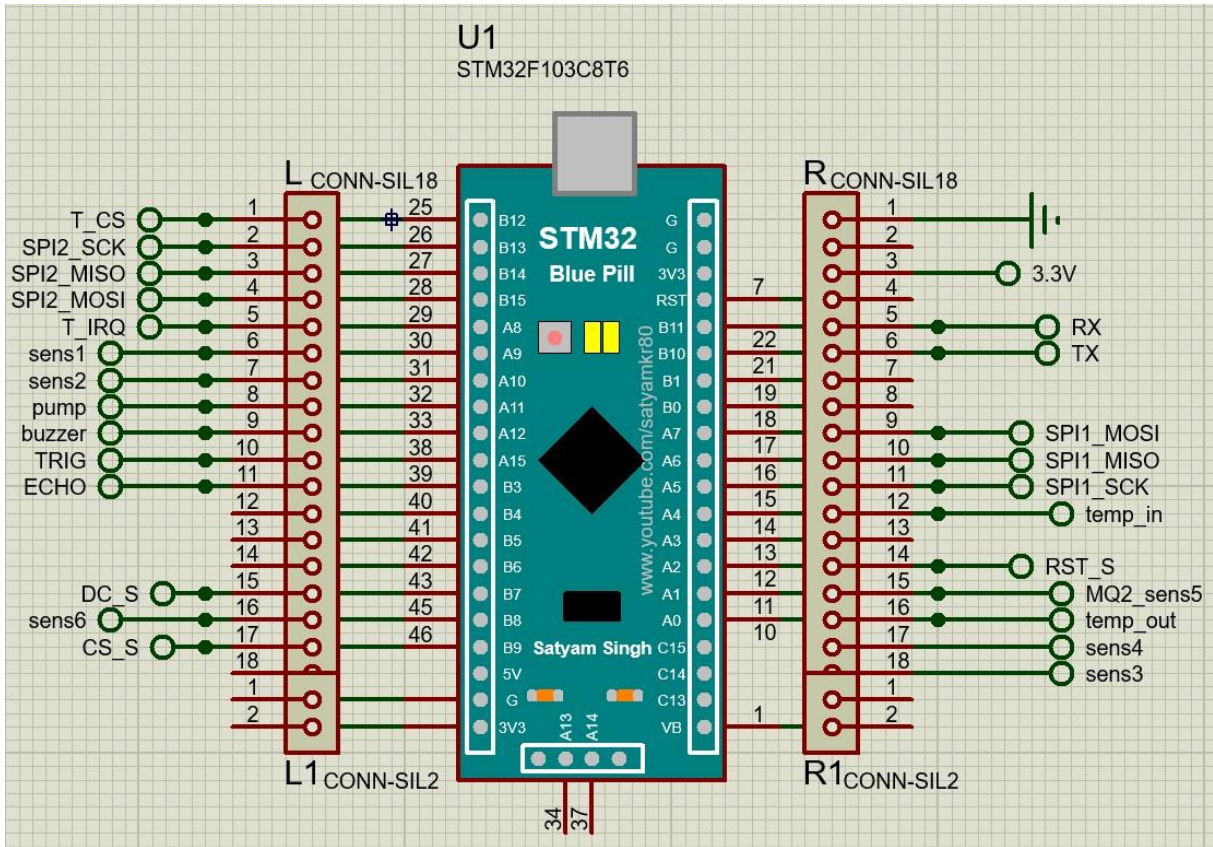
Εικόνα 2.27: Ρυθμίσεις USART3



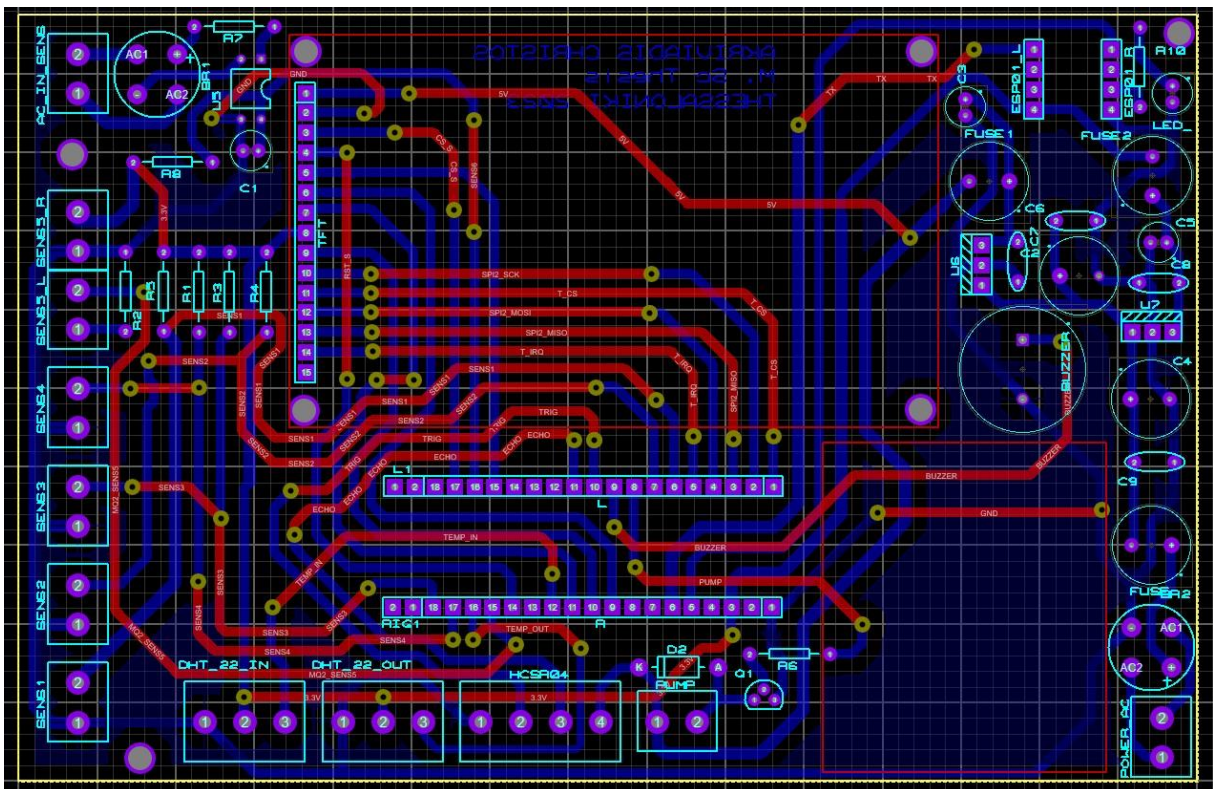
Σχήμα 2.9: Συνδεσμολογία ESP-01

## 2.16 Κατασκευή πλακέτας

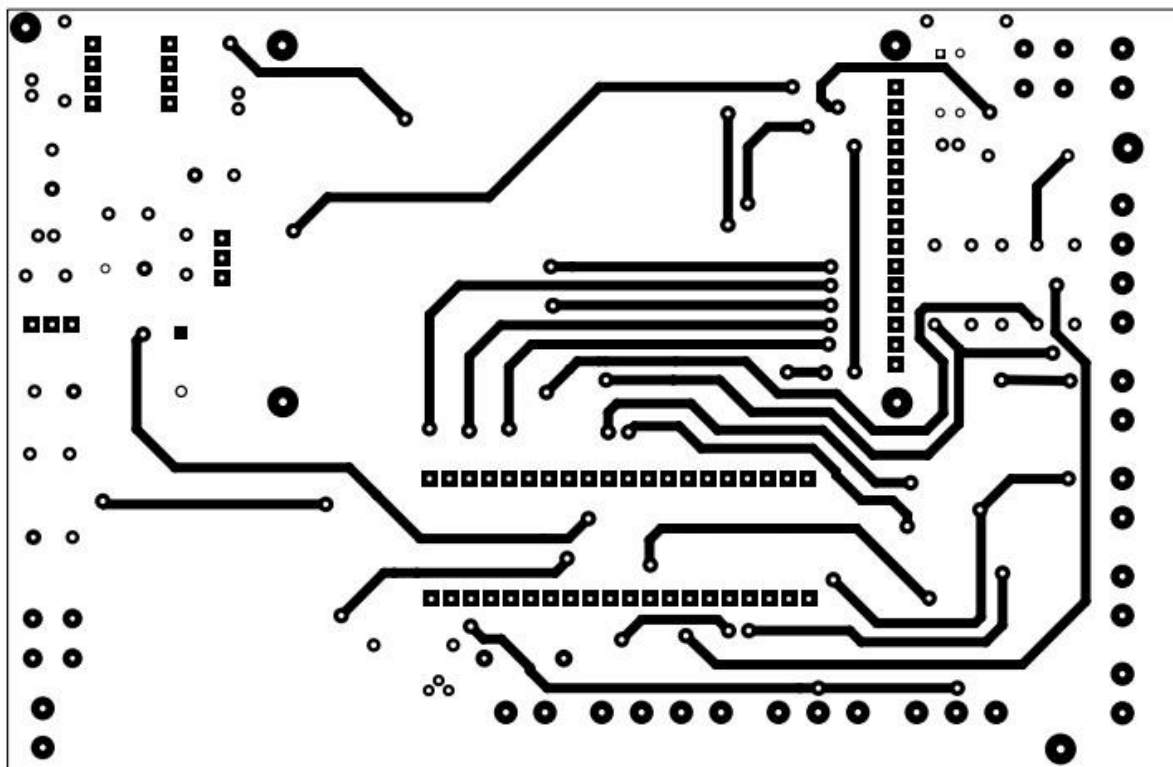
Η συνδεσμολογία των επιμέρους κυκλωμάτων με τον μικροελεγκτή προσφέρεται στο Σχήμα (2.10). Για τη δημιουργία του συνόλου των συνδέσεων, των σχηματικών και των σχεδίων για την τυπωμένη πλακέτα, χρησιμοποιήθηκε το ηλεκτρονικό πρόγραμμα Proteus 8 Professional Ver. 8.12. Η τυπωμένη πλακέτα, είναι διπλής όψης και αποτελεί ιδιοκατασκευή. Η διαδικασία που ακολουθήθηκε με τη σειρά, ήταν το κόψιμο της πλακέτας, στις επιθυμητές διαστάσεις. Στη συνέχεια η προετοιμασία αυτής, που αφορούσε το καθάρισμα και το τρίψιμο των επιφανειών, με σμυριδόπανο. Έπειτα το σιδέρωμα των επιφανειών, ώστε το τόνερ να αποκολληθεί από το χαρτί και να κολλήσει στην χάλκινη επιφάνεια, της κάθε πλευράς. Καθώς υλοποιήθηκε με τη μέθοδο του σιδερώματος [17], με σκοπό να γίνει σωστή ευθυγράμμιση της πάνω (top) και κάτω (bottom) πλευράς, αρχικά σιδερώθηκε η κάτω πλευρά. Ανοίχτηκαν δυο τρύπες στήριξης, που χρησιμοποιήθηκαν και ως οδηγίοι. Έπειτα έγινε η αποχάλκωση με ειδικά υγρά, της κάτω πλευράς και ακολούθησε το σιδέρωμα και η αποχάλκωση της πάνω πλευράς. Το τελευταίο στάδιο, αφορούσε το τρύπημα της πλακέτας και την κόλληση των εξαρτημάτων. Τέλος τα σχέδια της πλακέτας και οι φωτογραφίες μέσω του προγράμματος Proteus παρατίθενται στις Εικόνες 2.28 έως 2.30 και οι όψεις της τυπωμένης πλακέτας στις Εικόνες 2.31 και 2.32. Παράλληλα, η 3D όψη της πλακέτας και η τελική κατασκευασμένη πλακέτα (τύπου μητρικής) στις Εικόνες 2.33 και 2.34 ενώ μια γενική φωτογραφία της κατασκευής για τις ανάγκες της προσομοίωσης, στην Εικόνα 2.35.



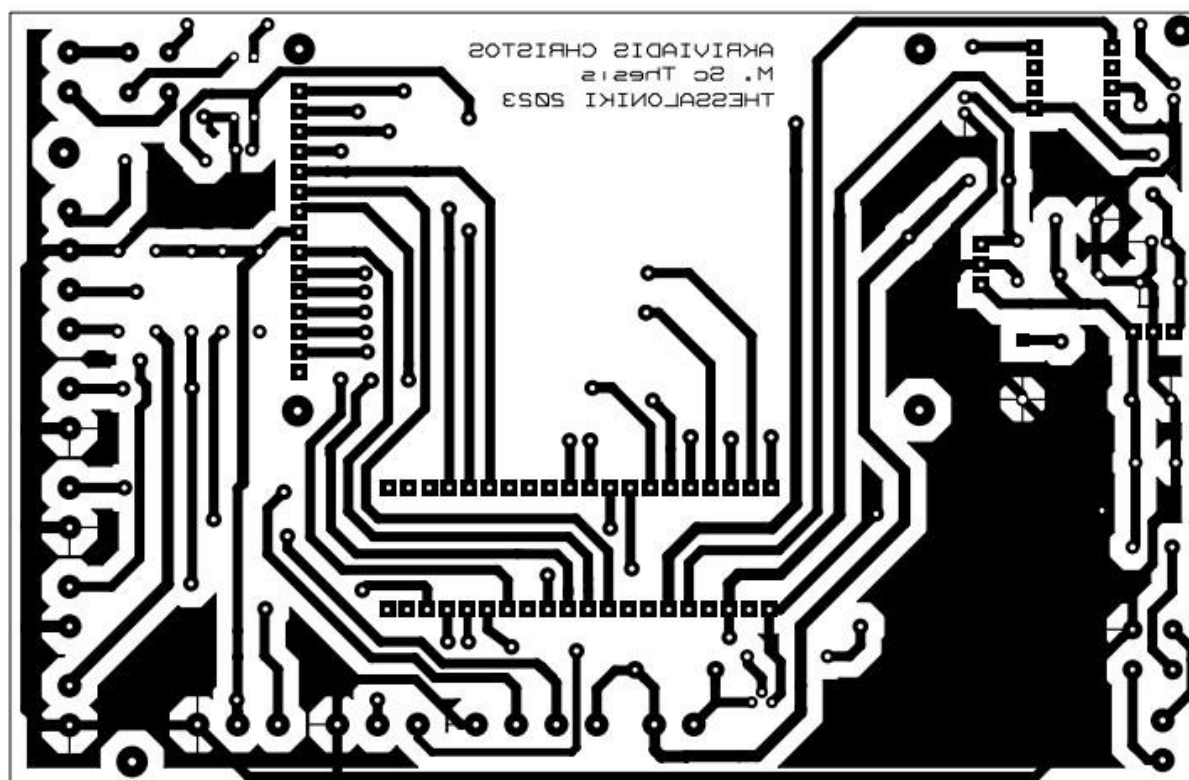
Σχήμα 2.10: Συνδεσμολογία επιμέρους κυκλωμάτων με τον μικροελεγκτή



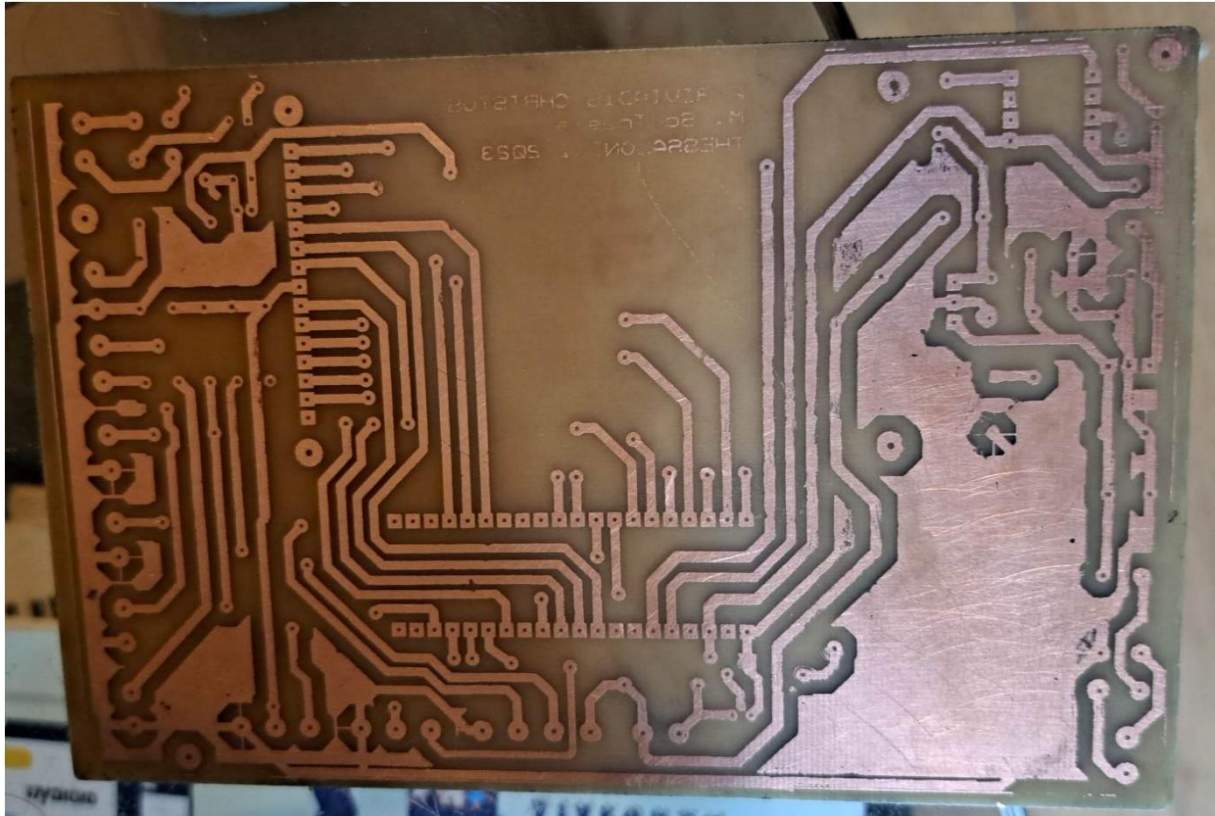
Εικόνα 2.28: Σχεδίαση πλακέτας 2 όψεων



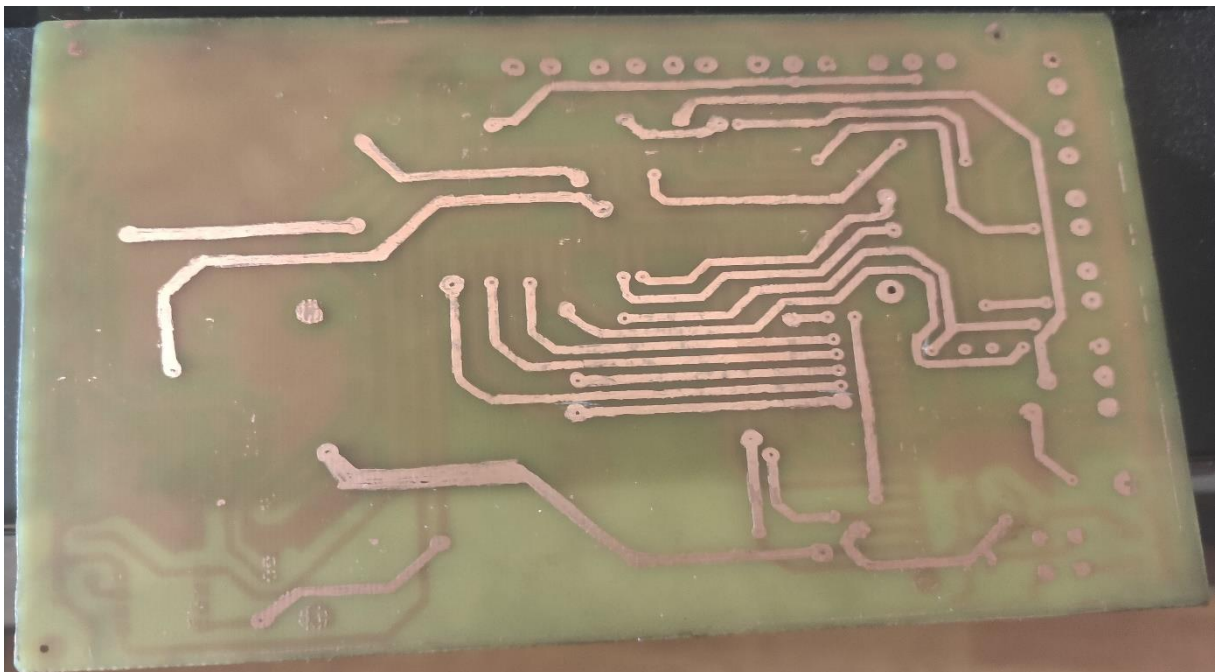
Εικόνα 2.29: Πάνω όψη της πλακέτας (top)



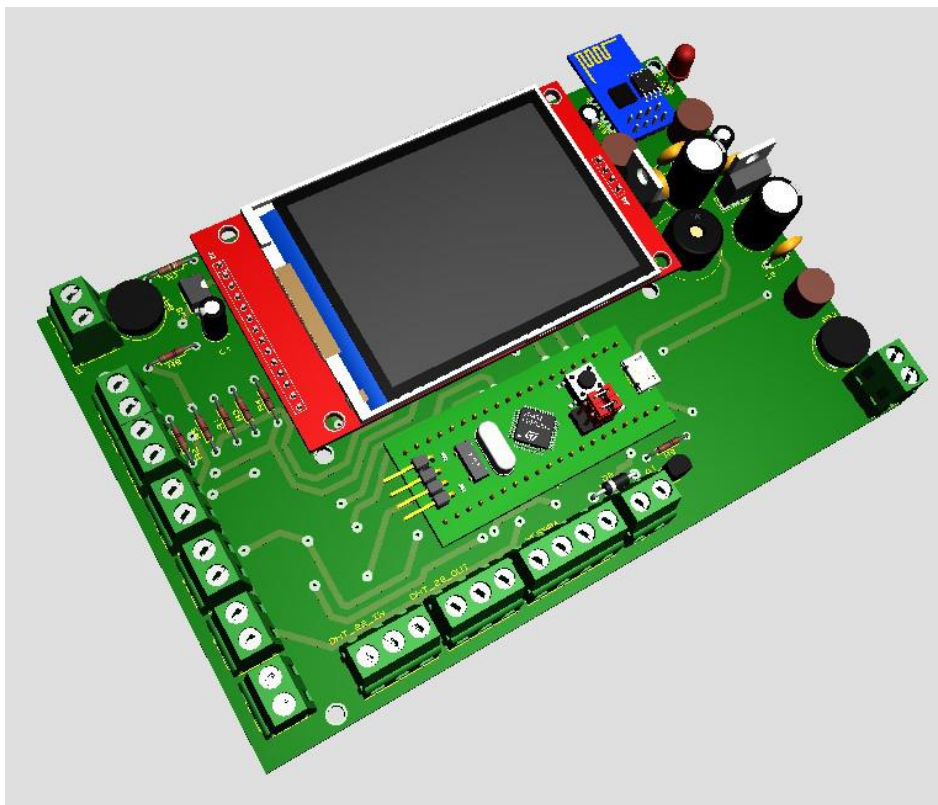
Εικόνα 2.30: Κάτω όψη της πλακέτας (bottom)



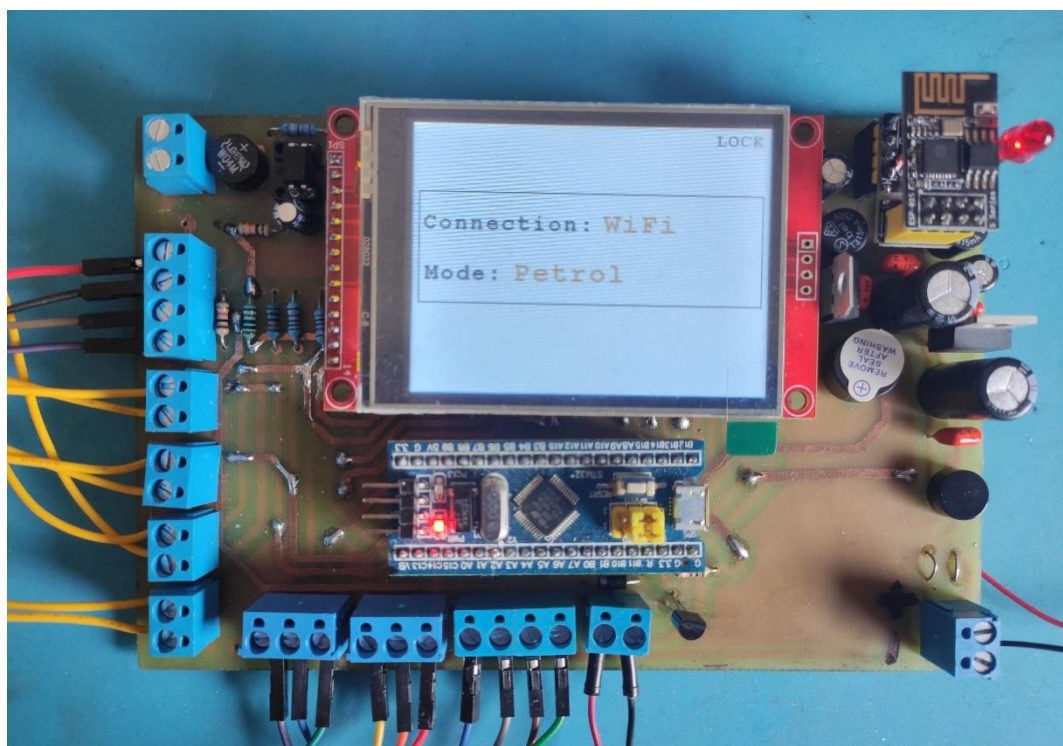
Εικόνα 2.31: Κάτω όψη της τυπωμένης πλακέτας



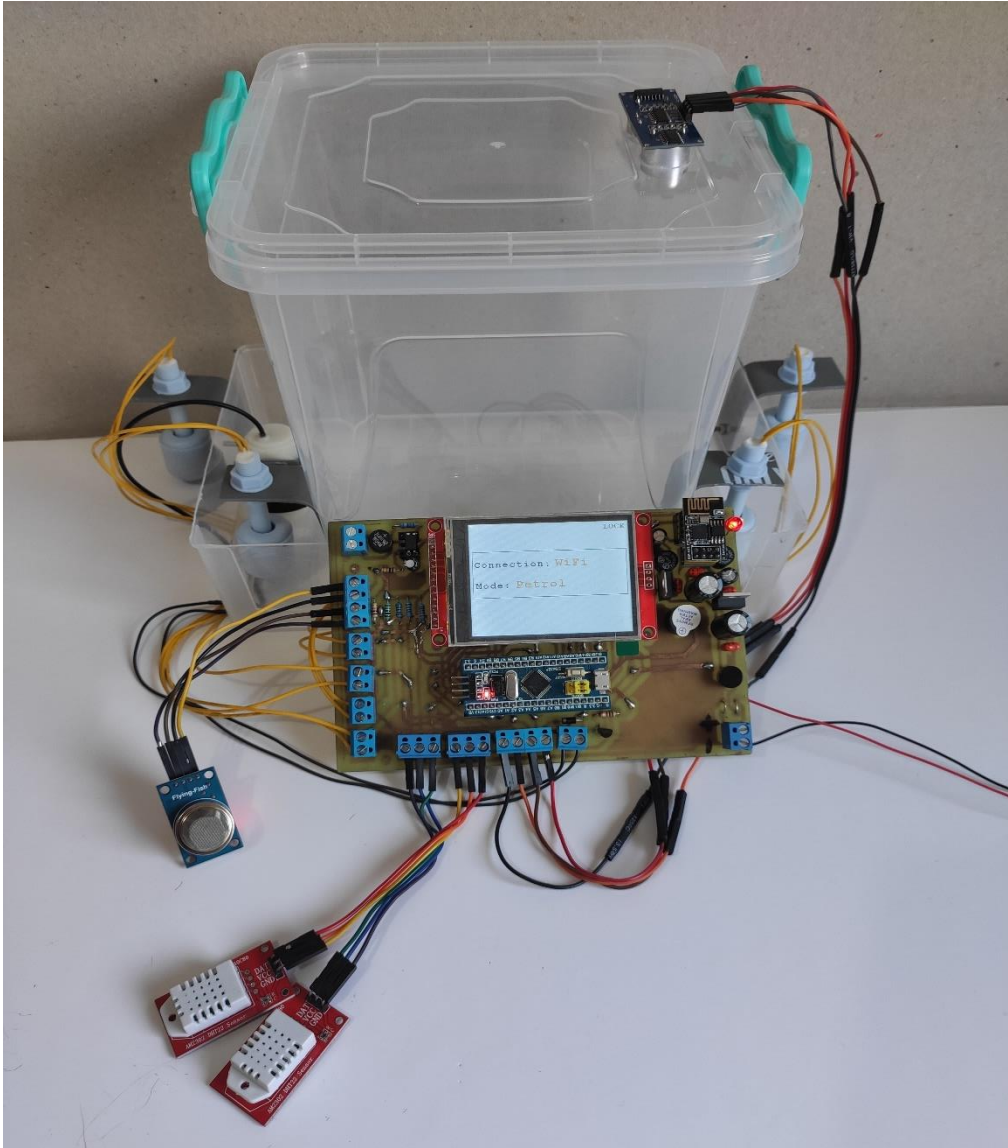
Εικόνα 2.32: Πάνω όψη της τυπωμένης πλακέτας



Εικόνα 2.33: 3D εμφάνιση της πλακέτας μέσω του προγράμματος Proteus



Εικόνα 2.34: Η κατασκευασμένη πλακέτα (τύπου μητρικής)



Εικόνα 2.35: Η συνολική κατασκευή

## 2.17 Επίλογος

Στο κεφάλαιο αυτό, έγινε μια εκτενή αναφορά στα αισθητήρια και τα υλικά που χρησιμοποιήθηκαν, για το σκοπό υλοποίησης της συσκευής καταγραφής. Μέσα από εικόνες και παραδείγματα, προσπάθησε να γίνει πιο κατανοητή η αρχή λειτουργίας των αισθητηρίων. Ακόμα, παρουσιάστηκαν τα κυκλώματα συνδεσμολογίας αυτών με τον μικροελεγκτή και οι βασικές προγραμματίστηκες ρυθμίσεις. Το κεφάλαιο ολοκληρώθηκε με τα συνολικά σχηματικά, καθώς και τα σχέδια και τις φωτογραφίες, της πλακέτας που κατασκευάστηκε. Τέλος, κάποιο ιδιαίτερο ενδιαφέρον παρουσίασε, η κατασκευή της διπλής όψης τυπωμένης πλακέτας, όσον αφορά το επιτυχές κεντράρισμα της πάνω (top) με την κάτω (bottom) πλευρά.

## Κεφάλαιο 3ο: Αρχή λειτουργίας της συσκευής

### 3.1 Εισαγωγή

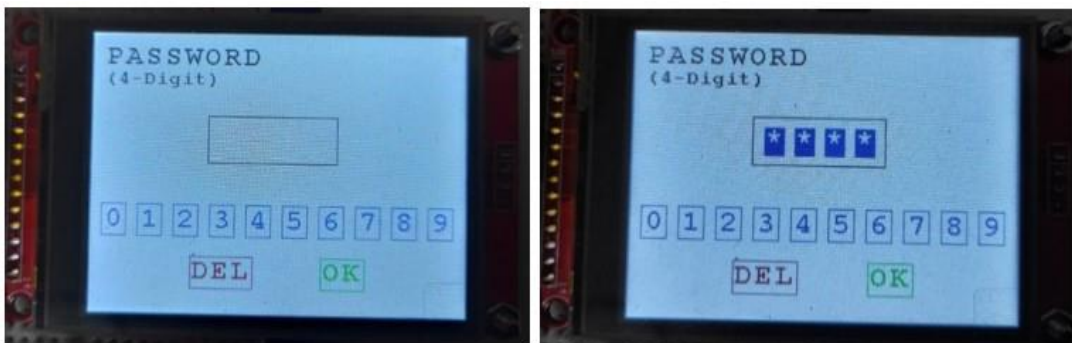
Στο δεύτερο κεφάλαιο έγινε μια εκτενή αναφορά στις ηλεκτρονικές διατάξεις και τα επιμέρους σχηματικά διαγράμματα, που συγκροτούν τη συσκευή καταγραφής. Επίσης, παρουσιάστηκαν τα αισθητήρια αυτής και έγινε αναφορά στα χαρακτηριστικά και την αρχή λειτουργίας τους. Έγινε σχολιασμός στο πως προγραμματίστηκαν και ποιες προγραμματιστικές ρυθμίσεις ήταν απαραίτητο να λάβουν χώρα. Αυτές ήταν αναγκαίες, ώστε να υπάρχει ορθή και ομαλή επικοινωνία των αισθητηρίων και των διατάξεων με τον μικροελεγκτή. Όλα αυτά, κρίθηκε απαραίτητο να αναφερθούν στο προηγούμενο κεφάλαιο, καθώς αποτελούν τη βάση για το παρόν κεφάλαιο, όπου γίνεται η ανάλυση της αρχής λειτουργίας της συσκευής. Με άλλα λόγια, γίνεται προσπάθεια μέσα από τις παραγράφους και τις Εικόνες, να μεταφερθεί στο “χαρτί” το κυρίως πρόγραμμα.

### 3.2 Ανάλυση αρχής λειτουργίας της συσκευής (κυρίως πρόγραμμα)

Συνδέοντας για πρώτη φορά τη συσκευή στη πηγή τροφοδοσίας, απαιτείται να οριστεί ένας 4ψήφιος κωδικός πρόσβασης για τη συσκευή (συνάρτηση *set\_password()*), μέσω της οθόνης αφής (Εικόνα 3.1). Αυτό συμβαίνει ώστε να διασφαλίζεται, ότι ο χειρισμός της θα μπορεί να γίνεται, μόνο από εξουσιοδοτημένα άτομα (εν τη προκύπτουσα περίπτωση, από τους διαχειριστές). Τα ψηφία που πληκτρολογήθηκαν, αποθηκεύονται σε ένα πρώτο πίνακα. Στη συνέχεια, ζητείται να εισαχθεί ως επιβεβαίωση, ο κωδικός που ορίστηκε προηγουμένως (Εικόνα 3.2) (συνάρτηση *password()*). Πρέπει να



Εικόνα 3.1: Ορισμός 4ψήφιου κωδικού πρόσβασης



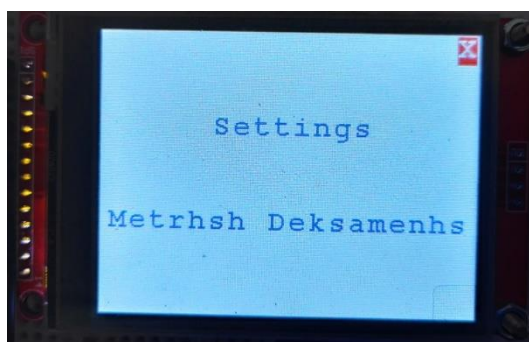
Εικόνα 3.2: Επιβεβαίωση κωδικού πρόσβασης



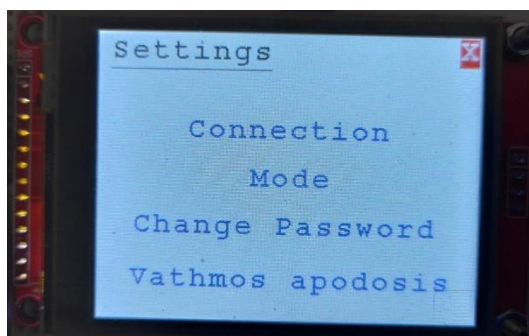


Εικόνα 3.3: Μήνυμα λανθασμένου και σωστού κωδικού πρόσβασης

εισαχθεί 4ψήφιος κωδικός, διαφορετικά το «OK» στην οθόνη, δεν μπορεί να επιλεγθεί. Με το «DEL» επιτυγχάνεται η διαγραφή ενός ή όλων των αριθμών. Τα ψηφία αυτά, αποθηκεύονται σε ένα δεύτερο πίνακα, ώστε να γίνει η σύγκριση με τα στοιχεία που δόθηκαν, στον ορισμό του κωδικού. Ο έλεγχος υλοποιείται μέσω ενός βρόχου for. Συγκρίνονται μια-μια οι θέσεις των δυο πινάκων και αν έστω και μια σύγκριση δεν υλοποιηθεί, τότε αυξάνεται κατά 1 μια μεταβλητή. Αν η μεταβλητή αυτή είναι διάφορη του μηδενός, τότε ο κωδικός είναι λανθασμένος, εμφανίζεται το μήνυμα «WRONG PASSWORD!» (Εικόνα 3.3) και το πρόγραμμα επιστρέφει εκ νέου στη σελίδα «PASSWORD» (Εικόνα 3.2). Θα μείνει εκεί έως ότου πληκτρολογηθεί ο σωστός κωδικός και πατηθεί το «OK». Τότε εμφανίζεται το μήνυμα «CORRECT PASSWORD!» (Εικόνα 3.3) και το πρόγραμμα συνεχίζει την επόμενη σελίδα. Εκεί εμφανίζεται ένα μενού με δυο επιλογές (Εικόνα 3.4). Η πρώτη ονομάζεται «Settings» και περιέχει ένα νέο μενού 4 επιλογών (Εικόνα 3.5). Η πρώτη κατά σειρά επιλογή, ονομάζεται «Connection» και μέσω αυτής γίνεται η επιλογή συνδεσιμότητας μεταξύ «Ethernet» και



Εικόνα 3.4: Μενού επιλογών



Εικόνα 3.5: Μενού επιλογής "Settings"



Εικόνα 3.6: Επιλογή σύνδεσης στο δίκτυο

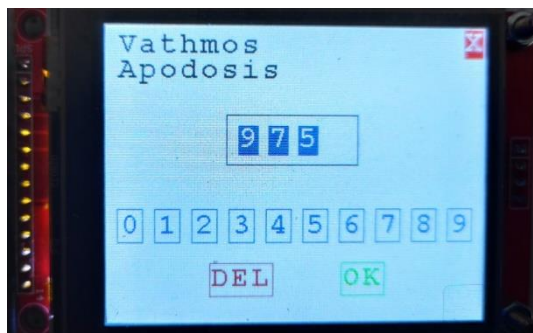
«WiFi» (Εικόνα 3.6). Την πρώτη φορά λειτουργίας της συσκευής, δεν είναι επιλεγμένη κάποια ρύθμιση, ενώ από την στιγμή επιλογής κι έπειτα, μπορεί να γίνεται εναλλαγή (των επιλογών). Η επιλεγμένη ρύθμιση ξεχωρίζει οπτικά, με την ενδεικτική πράσινη λυχνία, στα αριστερά του κάθε ονόματος. Ακόμα, πατώντας το «X» στην πάνω δεξιά πλευρά της οθόνης, το πρόγραμμα επιστρέφει στο μενού «Settings» (Εικόνα 3.5). Η δεύτερη επιλογή του μενού «Settings» (Εικόνα 3.5), είναι η επιλογή «Mode» (Εικόνα 3.7) για το είδος καυσίμου της θέρμανσης. Υπάρχει η επιλογή μεταξύ «Petrol» και «Gas». Όπως και πριν, την πρώτη φορά ενεργοποίησης της συσκευής, οι ενδεικτικές λυχνίες είναι γκρι. Όποια επιλεγεί, εμφανίζει στα αριστερά του ονόματος της την ενδεικτική πράσινη λυχνία, ενώ έπειτα, δύναται να γίνει εναλλαγή μεταξύ τους. Παράλληλα, πατώντας το «X» πάνω δεξιά, το πρόγραμμα επιστρέφει και πάλι στο μενού της Εικόνας 3.5. Στο σημείο αυτό, πρέπει να αναφερθεί πως οι επιλογές «Ethernet» και «Gas», αν και μπορούν να επιλεγούν μέσω της οθόνης, δεν υποστηρίζονται από το κυρίως πρόγραμμα. Η ύπαρξή τους αφορά τις μελλοντικές βελτιώσεις της συσκευής, που σχολιάζονται στο κεφάλαιο πέντε.

Το μενού «Settings» (Εικόνα 3.5) περιλαμβάνει άλλες δυο επιλογές. Την αλλαγή του κωδικού πρόσβασης «Change Password», που είχε εισαχθεί στην αρχή (το πρόγραμμα οδηγείται στη συνάρτηση `set_password()` Εικόνα 3.1). Οπότε πληκτρολογώντας και πατώντας «OK», τα καινούρια ψηφία αποθηκεύονται στον πρώτο πίνακα, που υπάρχει για το σκοπό αυτό. Η τέταρτη επιλογή της Εικόνας 3.5, είναι η επιλογή καταχώρησης του βαθμού απόδοσης του λέβητα «Vathmos apodosis» (Εικόνα 3.8). Αυτή μπορεί να λάβει από δυο έως τρία ψηφία, ενώ για τα «DEL» και «OK», ισχύει ότι και στην πληκτρολόγηση του κωδικού πρόσβασης. Με την ίδια λογική όπως και στα προηγούμενα, πατώντας μία φορά το «X», το πρόγραμμα επιστρέφει στο μενού της Εικόνας 3.5, ενώ πατώντας το και μια δεύτερη φορά, εμφανίζεται το μενού της Εικόνας 3.4.

Η δεύτερη επιλογή του μενού της Εικόνα 3.4, είναι η «Metrhsh Deksamenshs» (Εικόνα 3.9). Πατώντας επάνω, εμφανίζεται μια νέα σελίδα, που ρωτάει το χρήστη αν επιθυμεί, να γίνει μέτρηση απόστασης της δεξαμενής. Αυτή η επιλογή υπάρχει, με σκοπό να μπορεί ο διαχειριστής, να πάρει μια μέτρηση πριν και μια μέτρηση μετά, την πλήρωση της δεξαμενής. Με τον τρόπο αυτό, δεν χρειάζεται να καταφεύγει σε παλαιότερες μεθόδους (π.χ. μέτρηση με την κλασσική μεζούρα). Οι δυο μετρήσεις αποθηκεύονται

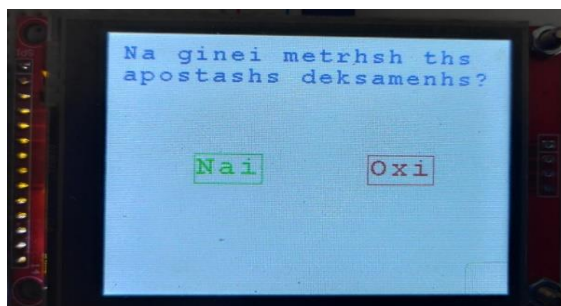


Εικόνα 3.7: Επιλογή καυσίμου κεντρικής θέρμανσης

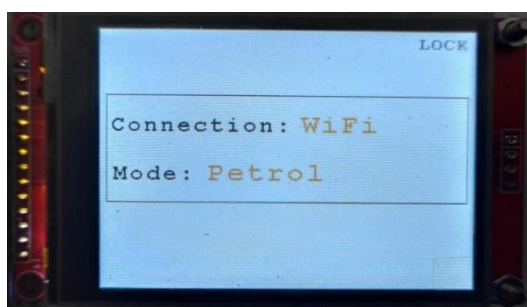


Εικόνα 3.8: Καταχώρηση βαθμού απόδοσης του λέβητα

με την ημερομηνία και ώρα που πάρθηκαν, στη βάση δεδομένων, όπως θα αναλυθεί στο επόμενο κεφάλαιο. Έτσι, πατώντας το «Ναί» γίνεται η μέτρηση ενώ πατώντας το «Όχι» δεν διενεργείται κάποια μέτρηση. Και στις δυο περιπτώσεις, το πρόγραμμα επιστρέφει στο μενού της Εικόνας 3.4. Ολοκληρώνοντας τις ρυθμίσεις, κι ενώ το πρόγραμμα βρίσκεται στο μενού της Εικόνας 3.4, πατώντας το «X» επάνω δεξιά, αυτό οδηγείται στην οθόνη αναμονής της Εικόνας 3.10. Εκεί προβάλλονται οι ρυθμίσεις του «Connection» και «Mode» και τη θέση του «X» παίρνει το «LOCK», υποδεικνύοντας ότι η οθόνη είναι κλειδωμένη. Πατώντας σε οποιοδήποτε σημείο πάνω στην οθόνη, αυτή ξεκλειδώνει και πριν επιστρέψει στο μενού τη Εικόνας 3.4, απαιτείται να εισαχθεί ο κωδικό πρόσβασης (Εικόνα 3.2). Έπειτα μπορούν να γίνουν εκ νέου όλες οι ρυθμίσεις, όπως αυτές παρουσιάστηκαν στις προηγούμενες παραγράφους. Όλα όσα αναφέρθηκαν μέχρι στιγμής, αφορούν ό,τι σχετίζεται με την οθόνη της συσκευής. Στις επόμενες παραγράφους, περιγράφονται όλα όσα σχετίζονται με τα αισθητήρια και τις μετρήσεις, καθώς και την αποστολή όλων των δεδομένων στο server.



Εικόνα 3.9: Μέτρηση δεξαμενής πριν και μετά την πλήρωση



Εικόνα 3.10: Οθόνη αναμονής

```

// Connect or reconnect to WiFi
if(WiFi.status() != WL_CONNECTED)
{
  Serial.println("Attempting to connect");
  while(WiFi.status() != WL_CONNECTED)
  {
    WiFi.begin(ssid, password);
    for (int i=0; i<40; i++)
    {
      digitalWrite(LED, ledState);
      ledState = !ledState;
      delay(250);
    }
  }

  if (WiFi.status() == WL_CONNECTED)
  {
    Serial.println("Connected.");
    digitalWrite(LED, HIGH);
  }
}

```

Εικόνα 3.11: Σύνδεση ή επανασύνδεση του ESP-01 στο δίκτυο

Τα δεδομένα από τους αισθητήρες, μεταφέρονται σειριακά στο ESP-01 όπου είναι υπεύθυνο, αρχικά να τα αναγνωρίσει κι έπειτα από κατάλληλη επεξεργασία, να τα στείλει μέσω του δικτύου στο server. Από εκεί αναλαμβάνει αυτός, ώστε να τα προωθήσει στη βάση δεδομένων. Συνεπώς, αρχικά γίνεται η προσπάθεια σύνδεσης ή επανασύνδεσης στο δίκτυο (Εικόνα 3.11). Αν για οποιοδήποτε λόγο δεν υπάρξει σύνδεση, το πρόγραμμα μένει σε έναν ατέρμον βρόχο έως ότου αυτή αποκατασταθεί, αναβοσβήνοντας παράλληλα την κόκκινη ενδεικτική λυχνία. Αυτή είναι τοποθετημένη στην μπροστινή πλευρά της συσκευής και με τον τρόπο αυτό, παρέχεται η οπτική ειδοποίηση της δυσλειτουργίας. Έτσι με το πέρας της αποκατάστασης, το πρόγραμμα του ESP-01 συνεχίζει, περιμένοντας να δεχθεί και να γεμίσει ένα πίνακα με τα σειριακά δεδομένα (Εικόνα 3.12). Οι πρώτες δυο ή τρεις θέσεις του πίνακα, υποδεικνύουν τι είδους δεδομένα έφτασαν, ώστε να γίνει και η κατάλληλη επεξεργασία αυτών. Στο τέλος, δημιουργείται προς αποστολή, ένα JSON αρχείο σαν αυτό της Εικόνας 3.13. Το πρώτο ζεύγος κλειδιού-τιμής είναι το token, που αφορά τον server και σχολιάζεται στο επόμενο κεφάλαιο, ενώ τα επόμενα ζεύγη κλειδιού-τιμής, περιέχουν τα αντίστοιχα δεδομένα.

```

if (Serial.available() > 0)
{
  // Read the incoming bytes:
  rlen = Serial.readBytes(str, 11);
  delay(100);
  // Print the received data
  Serial.print("STR-> ");

  for(int i=0; i<rlen; i++) // gemisma tou pinaka
  {
    str[i];
    Serial.print(String(str[i])+" ");
  }

  x = 1;
  Serial.println("\nLength "+String(rlen));
}

```

Εικόνα 3.12: Γέμισμα του πίνακα με τα εισερχόμενα δεδομένα

```

StaticJsonDocument<200> doc;
String json;

doc["token"] = "bH5GxzFoj4";
doc["temp"] = temp;
doc["hum"] = hum;

serializeJson(doc, json);

HTTPClient http;
http.begin(client, serverName);
http.addHeader("Content-Type", "application/json");
http.POST(json);
Serial.print(http.getString());
http.end();

```

Εικόνα 3.13: Δημιουργία JSON αρχείου προς αποστολή

Η μέτρηση της δεξαμενής (συνάρτηση *HCSR04\_Read()*) πραγματοποιείται τέσσερις φορές το 24ωρο, δηλαδή κάθε έξι ώρες. Επιλέχθηκε αυτή η λογική, ώστε να μην γεμίζει άσκοπα η βάση δεδομένων με πληροφορίες. Ο χρόνος προκύπτει από τον timer 3 και τις αντίστοιχες μεταβλητές, όπως αυτές αποτυπώνονται στην Εικόνα 3.14. Τα δεδομένα είναι ένας δεκαδικός αριθμός, με ακρίβεια ενός ψηφίου και στην πρώτη θέση υπάρχει το γράμμα «D», για γνωρίζει το ESP-01 τι δεδομένα έλαβε. Όσον αφορά τη μέτρηση της δεξαμενής πριν και μετά την πλήρωση, προφανώς και χρησιμοποιείται ο ίδιος κώδικας, με μόνη αλλαγή αυτή του πρώτου γράμματος σε «f» (Εικόνα 3.15). Από την πλευρά του, το ESP-01 αφού ξεχωρίσει ποια μέτρηση απόστασης ήρθε και γνωρίζοντας το μήκος των δεδομένων, μετατρέπει τα δεδομένα σε εκατοντάδες, δεκάδες, μονάδες και δέκατα (Εικόνα 3.16). Έπειτα τα αθροίζει και τα φορτώνει σε μια μεταβλητή. Ακολουθεί η διαδικασία της δημιουργίας του JSON αρχείου και τέλος η αποστολή του. Ακριβώς τα ίδια ισχύουν και για τη απόδοση του λέβητα, χρησιμοποιώντας ως πρώτο γράμμα το «a».

```

void HAL_TIM_PeriodElapsedCallback(TIM_HandleTypeDef *htim) // timers
{
  if(htim->Instance == TIM4) // gia metrsh me to DHT22 kathe 5 lepta otan leitourgei o kausthras
  {
    tim4_sec++; // every second
  }

  if (tim4_sec == 60)
  {
    tim4_sec=0;
    tim4_min++; // every minute
  }

  if(htim->Instance == TIM3) // main counter gia Distance kai UIC
  {
    count_sec++; // every second
  }

  if (count_sec == 60)
  {
    count_sec=0;
    count_min++; // every minute gia distance
    count_min_uic++; // every minute gia uic
    tim5_min++;
  }

  if(count_min == 360) // 6*60 -> 360 gia 6 wres
  {
    time_to_read = 1;
    count_min=0;
    count_min_uic=0; // mhdnizei giati tha kansi post apo to distance kai etsi sto backend tha parsi last timestamp
    uic_send = 0;
  }

  if(count_min_uic == 5) // gia dummy json gia uic kathe 5 lepta
  {
    uic_send = 1; // set f "1"
    count_min_uic=0;
  }
}

```

Εικόνα 3.14: Χρόνοι και μεταβλητές για τις μετρήσεις

```

if ((start > 0 && start != 3) && Distance >= 2)
{
    HAL_UART_Transmit(&huart3, (uint8_t *)&buffer, sprintf(buffer, "D%.1f", Distance),1000);
}

if (start == 3 && Distance >= 2) // plhrwsh deksamenhs prinmai meta
{
    HAL_UART_Transmit(&huart3, (uint8_t *)&buffer, sprintf(buffer, "f%.1f", Distance),1000);
}

for (int i=0; i<11; i++)
{
    buffer[i]=0;
}

```

Εικόνα 3.15: Διαφοροποίηση πρώτου γράμματος κατά τη σειριακή μεταφορά των δεδομένων

```

else if (str[0] == 'D' && x==1) // metrsh deksamenhs
{
    Serial.print("I received: ");

    for(int i=1; i<rlen; i++)
    {
        Serial.print(String(str[i])+" ");
    }
    Serial.println("\nlen  "+String(rlen));

    if (rlen == 4) //0.1
    {
        float da = ((str[1])-48);
        float dz = ((str[3])-48);
        float db = dz/10;
        dist = da+db;
    }

    else if (rlen == 5) //10.1
    {
        float da = ((str[1])-48)*10;
        float db = ((str[2])-48);
        float dz = ((str[4])-48);
        float dc = dz/10;
        dist = da+db+dc;
    }

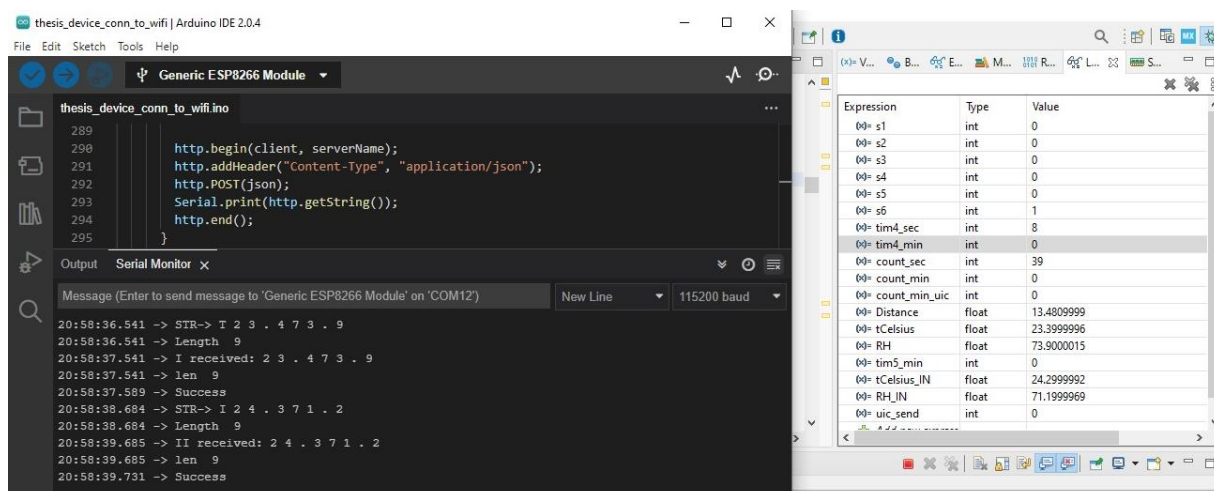
    else if (rlen == 6) //100.1
    {
        float da = ((str[1])-48)*100;
        float db = ((str[2])-48)*10;
        float dc = ((str[3])-48);
        float dz = ((str[5])-48);
        float dd = dz/10;
        dist = da+db+dc+dd;
    }
}

```

Εικόνα 3.16: Διαδικασία επεξεργασίας των δεδομένων μέτρησης της δεξαμενής από το ESP-01

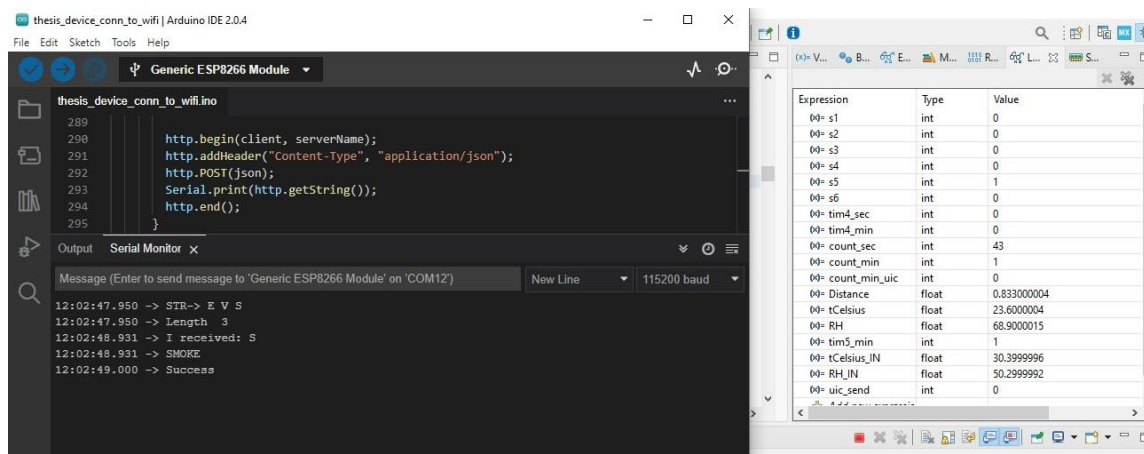
Η μέτρηση της εξωτερικής και εσωτερικής θερμοκρασίας πραγματοποιείται κάθε 5 λεπτά και όσο ο λέβητας βρίσκεται σε λειτουργία. Αυτές οι μετρήσεις είναι από τις πιο σημαντικές, καθώς οι χαμηλές θερμοκρασίες επηρεάζουν στη μεγαλύτερη κατανάλωση του καυσίμου. Οι συναρτήσεις που καλούνται είναι οι *Temp\_Hum()* και *Temp\_Hum\_IN()* αντίστοιχα και οι χρόνοι υπολογίζονται μέσω της Εικόνας 3.14. Η μεταφορά και επεξεργασία των δεδομένων στο ESP-01, ακολουθεί την ίδια λογική όπως αυτή της μέτρησης απόστασης. Για την εξωτερική θερμοκρασία το πρώτο γράμμα είναι το «T» ενώ για την εσωτερική το «I». Τα δεδομένα έχουν ακρίβεια ενός δεκαδικού ψηφίου και τα πρώτα στη σειρά αφορούν τη θερμοκρασία, ενώ αυτά που έπονται είναι της υγρασίας. Ένα παράδειγμα που αναδεικνύει την λήψη και σειριακή αποστολή των δεδομένων, από το μικροελεγκτή στο ESP-01, καθώς και την αποστολή στο server, παρουσιάζεται στην Εικόνα 3.17.

Τα αισθητήρια για την ασφάλεια του χώρου, κινούνται στο ίδιο μήκος κύματος με τα προηγούμενα, όσον αφορά τη μεταφορά του αντίστοιχου συμβάντος ή συμβάντων, καθώς και την αποστολή. Για οποιοδήποτε συμβάν μέσα από τη *while(1)*, καλείται η συνάρτηση *event()*, έχοντας “σηκωθεί” και η αντίστοιχη ή οι αντίστοιχες σημαίες. Στην αποστολή, τα πρώτα δυο γράμματα είναι τα «EV» και ανάλογα το τι ακολουθεί, υποδηλώνει και το συμβάν. Στην Εικόνα 3.18, αποτυπώνεται το παράδειγμα για την ανίχνευση καπνού στο χώρο. Όταν ενεργοποιηθεί ο αισθητήρας, καλείται η συνάρτηση και η σειριακή αποστολή, θα μεταφέρει στο ESP-01 τα γράμματα «EVS». Το ESP-01 θα τα επεξεργαστεί και θα στείλει στο server το συμβάν, για την ύπαρξη καπνού. Στον Πίνακα 3.1 παρατίθεται ο συνδυασμός των 8 καταστάσεων που μπορεί να υπάρξουν. Στην τελευταία στήλη αποτυπώνεται και η ενεργοποίηση της αντλίας (ενεργοποιείται με “1”). Εκκινεί όταν ανιχνευθεί διαρροή από τη δεξαμενή πετρελαίου, αντλώντας και επιστρέφοντας το πετρέλαιο προς τα πίσω, στη δεξαμενή. Η χρήση της στοχεύει στο να μη χυθεί το πετρέλαιο στο έδαφος, άρα και χαθεί, έως ότου αντιμετωπιστεί η βλάβη. Ο τρόπος αυτός, φιλοδοξεί την επιπλέον οικονομία, μέσα από μια απρόσμενη κατάσταση, που δημιουργήθηκε. Το μόνο που χρειάζεται σε επίπεδο κατασκευής, είναι η τοποθέτηση ενός πλαισίου ακριβώς κάτω από την δεξαμενή. Μέσα του θα τοποθετηθούν, τα αισθητήρια ελέγχου διαρροής. Ταυτόχρονα στην εξωτερική πλευρά του πλαισίου, θα τοποθετηθούν τα αισθητήρια για τον έλεγχο της πλημμύρας του χώρου (Εικόνα 3.19). Τα αισθητήρια διαρροής και πλημμύρας, αποτελούνται από ζεύγος των δυο, με σκοπό το ένα να ελέγχει το άλλο. Δηλαδή, δίνεται εντολή για το αντίστοιχο συμβάν, όταν και τα δυο ενεργοποιηθούν.



Εικόνα 3.17: Από δεξιά τα δεδομένα προς σειριακή αποστολή από το μικροελεγκτή ενώ από αριστερά δεδομένα που λήφθηκαν από το ESP-01 και στάλθηκαν στο server

## Κεφάλαιο 3



Εικόνα 3.18: Από δεξιά η ενεργοποίηση του αισθητήρα καπνού μέσω του STM32CubeIDE 1.8.0 και αριστερά τα ληφθέντα δεδομένα στο ESP-01 και η αποστολή τους

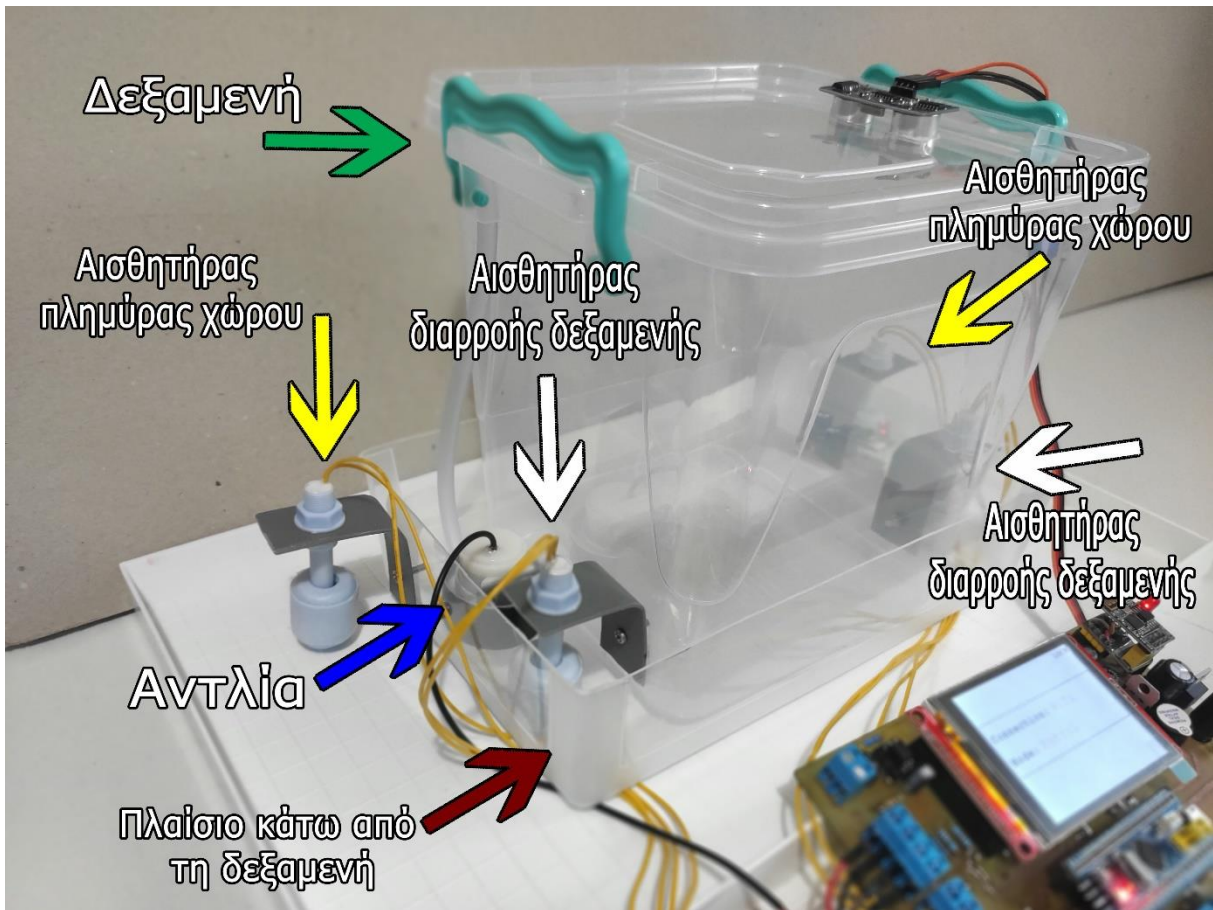
Στα αισθητήρια ασφάλειας του χώρου, συγκαταλέγεται και το DHT22, που μετρά την εσωτερική θερμοκρασία, όταν ο λέβητας δεν βρίσκεται σε λειτουργία. Γίνεται μια μέτρηση κάθε 5 λεπτά (Εικόνα 3.14) και αν αυτή είναι μεγαλύτερη από 40°C, αποστέλλεται ειδοποίηση, που καταλήγει στο server. Σε όλες τις περιπτώσεις που υπάρχει κάποιο συμβάν, ενεργοποιείται και το buzzer που βρίσκεται πάνω στην πλακέτα. Αυτό σκοπό έχει, την επιπλέον ηχητική ειδοποίηση, πέρα από το email που στέλνεται από την πλευρά του back-end (αναλύεται στο επόμενο κεφάλαιο).

Πίνακας 3.1: Συνδυασμός συμβάντων

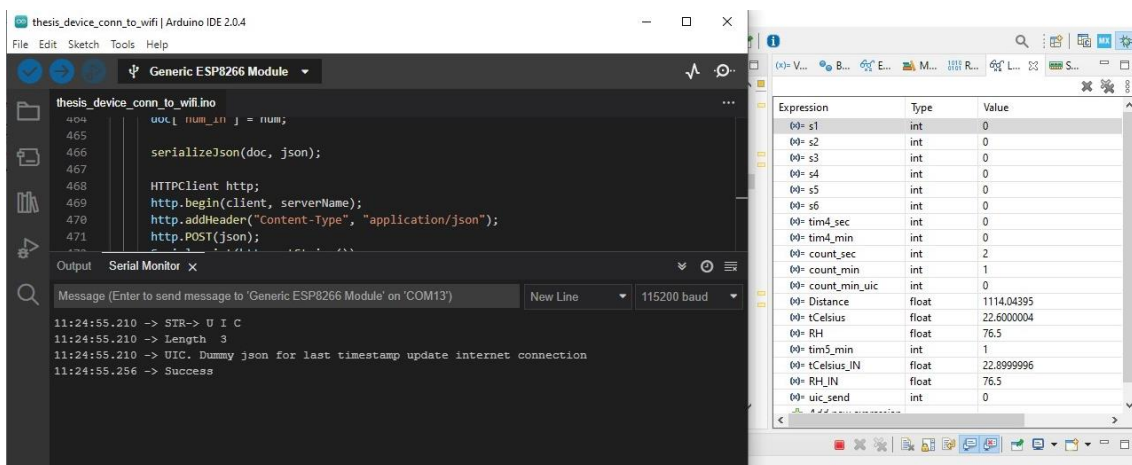
A/A	Αισθητήρας καπνού (S)	Αισθητήρας πλημμύρα χώρου (O)	Αισθητήρας διαρροής δεξαμενής (L)	Ενεργοποίηση αντλίας
1	0	0	0	0
2	0	1	0	0
3	1	0	0	0
4	1	1	0	0
5	0	0	1	1
6	0	1	1	1
7	1	0	1	1
8	1	1	1	1

Ολοκληρώνοντας, μιας και η συσκευή θα τοποθετείται σε χώρο που δεν υπάρχει η καθημερινή διέλευση ανθρώπων, δημιουργείται η ανάγκη επιτήρησης της ίδιας της συσκευής. Αυτή αφορά στην τροφοδοσία και την σύνδεση της στο δίκτυο. Έτσι, έχει ρυθμιστεί να στέλνει ένα κενό JSON αρχείο στο server, όταν δεν διενεργείται κάποια διεργασία (μέτρηση δεξαμενής, μέτρηση θερμοκρασίας ή κάποιο συμβάν) με χρόνο αποστολής, κάθε 5 λεπτά (Εικόνα 3.14). Ο λόγος ύπαρξης του κενού POST, είναι να δημιουργεί ένα last timestamp, ώστε από εκεί και πέρα να το χειριστεί ο server κατάλληλα. Τα αρχικά αναγνώρισης για το ESP-01 είναι τα «UIC» και στην Εικόνα 3.20 προβάλλεται το αντίστοιχο παράδειγμα. Τέλος, στο Σχήμα (3.1) αποτυπώνεται μέσω του μπλοκ διαγράμματος, η λειτουργία της συσκευής, όπως αυτή παρουσιάστηκε στις προηγούμενες παραγράφους, του κεφαλαίου αυτού.

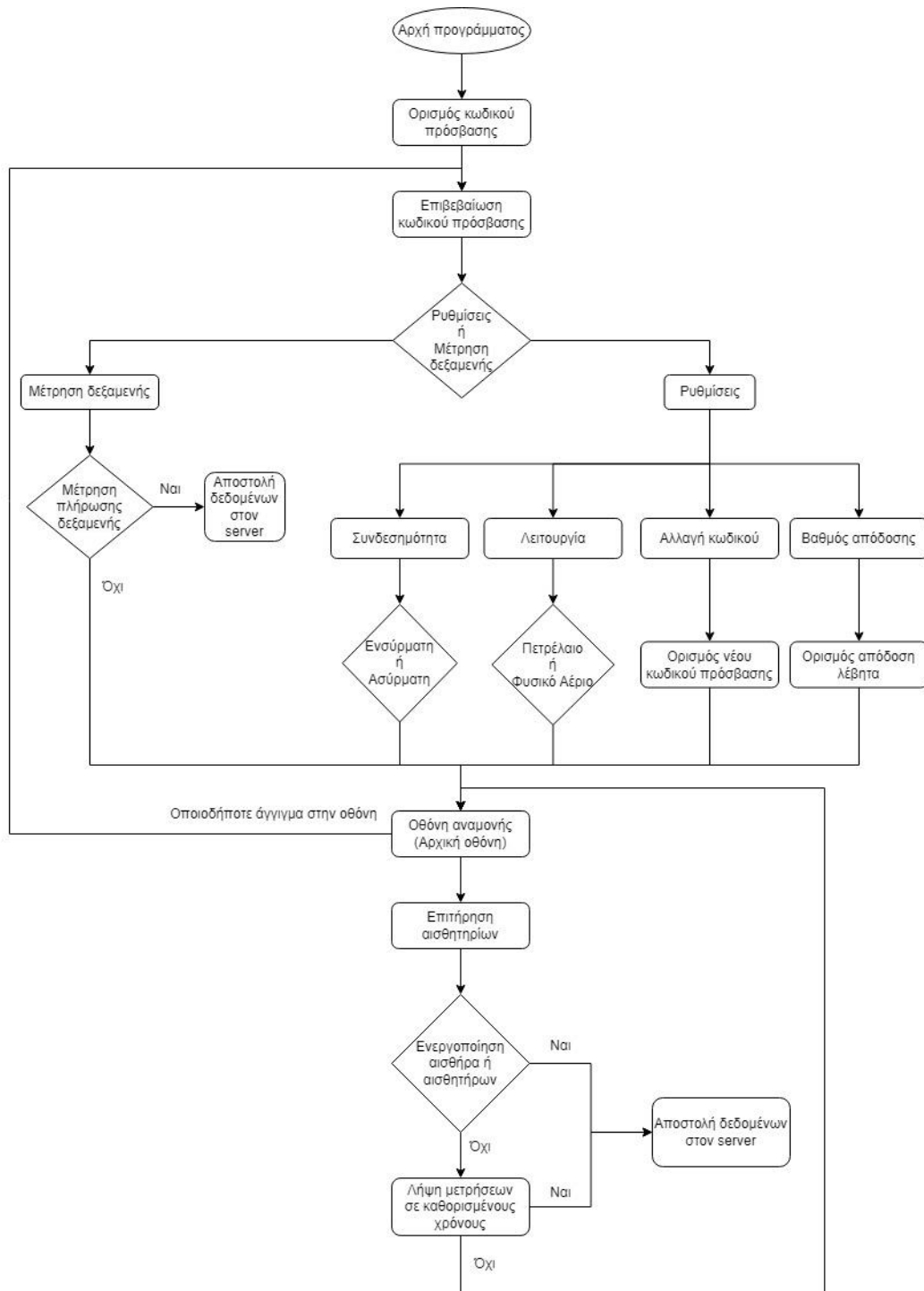




Εικόνα 3.19: Η κατασκευή με την προσομοίωση της δεξαμενής και την τοποθέτηση του πλαισίου κάτω από αυτή καθώς και η τοποθέτηση των αισθητήρων ελέγχου πλημύρας και διαρροής δεξαμενής



Εικόνα 3.20: Αποστολή κενού POST



Σχήμα 3.1: Μπλοκ διάγραμμα λειτουργίας της συσκευής

### 3.3 Επίλογος

Στις ενότητες του κεφαλαίου αυτού, παρουσιάστηκε το κυρίως πρόγραμμα της συσκευής καταγραφής και αφορούσε το STM32F103C8T6 και το ESP-01. Τα σχέδια και οι Εικόνες που προβλήθηκαν, σκοπό είχαν την ενδυνάμωση της κατανόησης, της αρχής λειτουργίας. Έτσι σε συνδυασμό με όσα ειπώθηκαν στο δεύτερο κεφάλαιο, διαλευκάνθηκε πλήρως πως λειτουργεί η συσκευή και που στοχεύει. Τέλος, στα Παραρτήματα Α και Β υπάρχουν οι κώδικες του μικροελεγκτή και του ESP-01, αντίστοιχα.

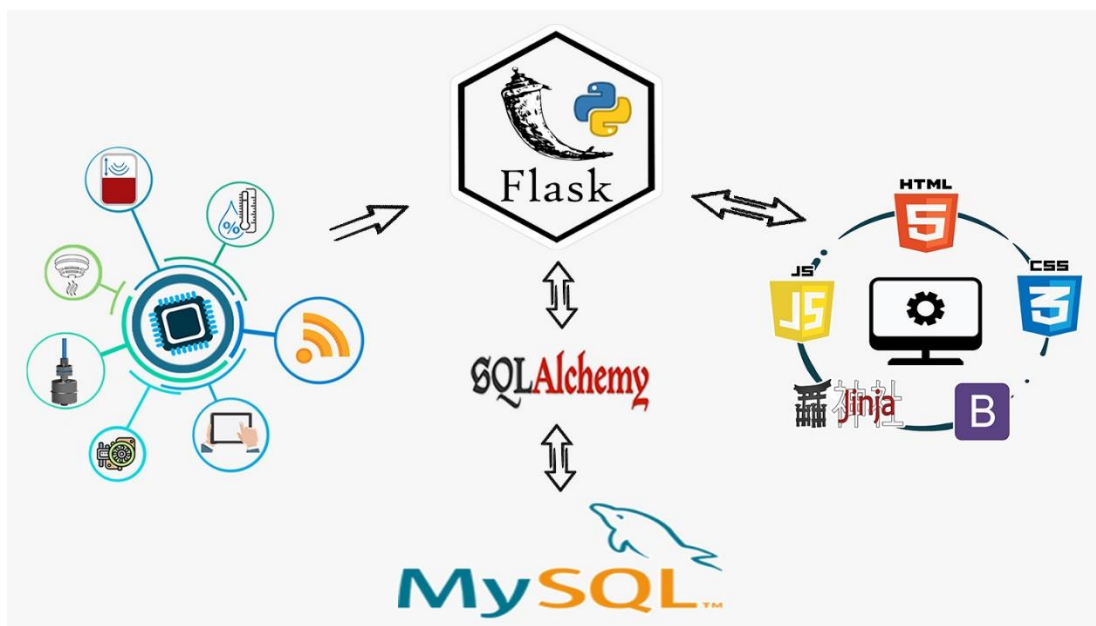
## Κεφάλαιο 4ο: Λογισμικό του συστήματος

### 4.1 Εισαγωγή

Αν η ραχοκοκαλιά του συστήματος είναι η συσκευή με τα αισθητήρια, τότε σίγουρα το λογισμικό (software) είναι η καρδιά του. Είναι αυτό που δίνει την απαιτούμενη λειτουργικότητα, καθώς επεξεργάζεται και χειρίζεται τα δεδομένα, για λογαριασμό του χρήστη, εξυπηρετώντας ταυτόχρονα τα αιτήματά του.

Το λογισμικό διαρθρώνεται σε τρία μέρη. Το πρώτο αποτελείται από τη βάση δεδομένων, όπου εκεί καταχωρούνται όλα τα δεδομένα, που στέλνει η συσκευή και ο χρήστης, με σκοπό τη μακροχρόνια αποθήκευση. Χρησιμοποιείται η βάση δεδομένων MySQL και η γλώσσα προγραμματισμού SQL. Το δεύτερο μέρος είναι το back-end, το οποίο δεν είναι ορατό στο χρήστη και είναι υπεύθυνο για την οργάνωση, διαχείριση και ομαλή λειτουργία της εφαρμογής. Η υλοποίηση του έγινε με τη γλώσσα προγραμματισμού Python και το εργαλείο Flask. Έχει προγραμματιστεί με σκοπό να διαχειρίζεται τα δεδομένα και να επικοινωνεί με τη βάση, γι' αυτά. Ακόμα, είναι υπεύθυνο για το χειρισμό των αιτημάτων που καταφθάνουν από τρίτο μέρος, το front-end, όπου εκεί αλληλοεπιδρούν οι χρήστες. Από το κομμάτι αυτό, ο server δέχεται δεδομένα προς αποθήκευση (στη Β.Δ.) ή αντλεί από τη Β.Δ. και προωθεί δεδομένα, ώστε να είναι ορατά στους ένοικους (χρήστες) της πολυκατοικίας. Η αλληλοεπίδραση γίνεται μέσω μιας διαδικτυακής εφαρμογής, κοινώς μιας ιστοσελίδας. Για την κατασκευή της (front-end) απαιτήθηκε η συνεργασία των γλωσσών προγραμματισμού, HTML για το περιεχόμενο της, CSS για την απόδοση ενός οπτικά όμορφου αποτελέσματος, στο περιεχόμενο και η JavaScript ώστε να ενσωματώσει τη λειτουργικότητα. Ακόμα, έγινε η χρήση του εργαλείου Bootstrap, όπου μέσω αυτού χρησιμοποιήθηκαν και προσαρμόστηκαν κατάλληλα, κάποια έτοιμα στοιχεία (π.χ. το μενού της σελίδας, τα κουμπιά κ.α.). Για τη συγγραφή όλων των κομματιών του κώδικα, χρησιμοποιήθηκε το προγραμματιστικό περιβάλλον του PyCharm 2023.1.

Τέλος, η Εικόνα 4.1 έρχεται να ενώσει όσα ειπώθηκαν στα κεφάλαια 2 και 3, με αυτά που θα αναφερθούν στο κεφάλαιο 4, παρουσιάζοντας οπτικά την αρχιτεκτονική του συστήματος.



Εικόνα 4.1: Αρχιτεκτονική του συστήματος [62-81]

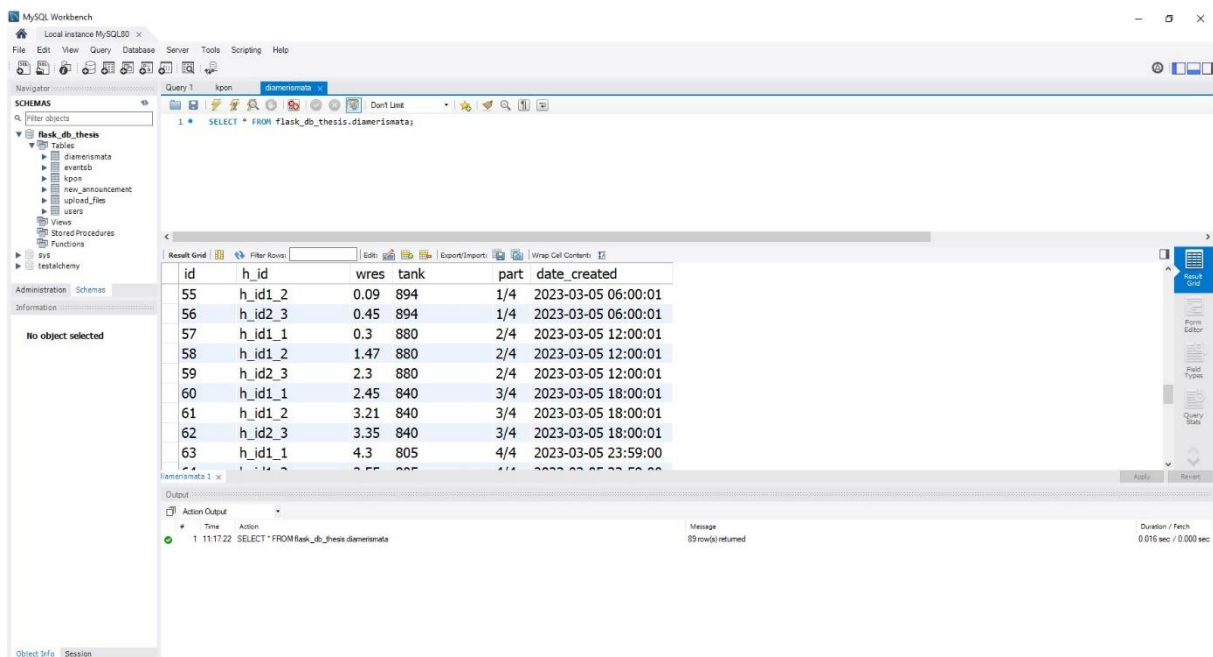
## 4.2 Βάση δεδομένων MySQL

Για το σκοπό της μακροχρόνιας καταγραφής (αποθήκευσης) των δεδομένων, χρησιμοποιείται η MySQL βάση δεδομένων, της εταιρίας ORACLE. Αποτελεί ένα σύστημα διαχείρισης σχεσιακών βάσεων (RDBMS) και για την επεξεργασία των δεδομένων, χρησιμοποιείται η γλώσσα SQL (Structured Query Language). Η εισαγωγή των πληροφοριών γίνεται κατά γραμμές του πίνακα, στις αντίστοιχες στήλες και για τις ανάγκες της διπλωματικής εργασίας, ο χειρισμός της έγινε μέσω του προγράμματος MySQL Workbench 8.0 [18]. Προτιμήθηκε η συγκεκριμένη Β.Δ., καθώς είναι ανοιχτού κώδικα και συνδυάζει την ταχύτητα, με την ευελιξία και αξιοπιστία.

Αρχικά απαιτείται να γίνει σύνδεση στον MySQL Server (Εικόνα 4.2) και στη συνέχεια εμφανίζεται η αρχική οθόνη του προγράμματος (Εικόνα 4.3). Μέσα από το μενού *Server* και *Users and Privileges*, γίνεται η δημιουργία του χρήστη και από την καρτέλα *Schema Privileges* επιλέγονται οι ενέργειες που θα μπορεί να πράττει (εν τη προκυμμένη περίπτωση όλες). Μερικές από αυτές είναι η δημιουργία ή



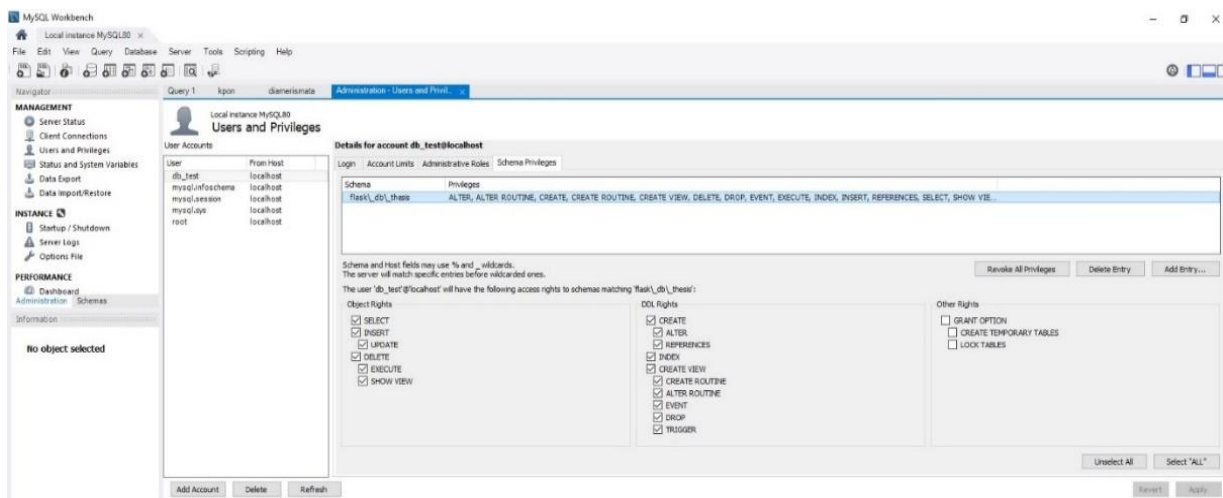
Εικόνα 4.2: Σύνδεση στον MySQL Server



Εικόνα 4.3: Αρχική οθόνη του MySQL Workbench και εμφάνιση του πίνακα diamerismata

τροποποίηση πίνακα, επεξεργασία δεδομένων κ.α. (Εικόνα 4.4). Έτσι, επιστέφοντας και πάλι στην αρχική οθόνη (Εικόνα 4.3), στα αριστερά υπάρχει το πλαίσιο του *SCHEMAS*, όπου εκεί εμφανίζεται η βάση δεδομένων (*flask\_db\_thesis*), που δημιουργήθηκε για να εξυπηρετήσει την εφαρμογή. Πατώντας επάνω και αναπτύσσοντας, εμφανίζεται η καρτέλα *Table* όπου μέσα της περιέχονται οι έξι πίνακες της βάσης. Αυτοί είναι με σειρά εμφάνισης, ο *diamerismata*, ο *eventsb*, ο *kron*, ο *new\_announcement*, ο *upload\_files* και ο *users*. Ο πρώτος περιέχει τα δεδομένα των ωρών κάθε διαμερίσματος και τη στάθμη της δεξαμενής. Ο δεύτερος περιέχει όποια συμβάντα έχει καταγράψει η συσκευή και ο τρίτος την εσωτερική και εξωτερική θερμοκρασία-υγρασία. Επιπρόσθετα, ο τέταρτος περιέχει τις ανακοινώσεις που δημοσιεύουν οι διαχειριστές, ενώ ο πέμπτος τα αρχεία που ανεβάζουν και τέλος, ο έκτος περιέχει όλους τους χρήστες που δημιούργησαν λογαριασμό, για την πρόσβαση στην ιστοσελίδα. Στη συνέχεια, στο κέντρο της οθόνης εμφανίζεται η καρτέλα *Query 1*, όπου εκεί πληκτρολογώντας εντολές σε γλώσσα *SQL*, είναι δυνατή η επεξεργασία του συνόλου της βάσης. Επίσης σε διπλανές καρτέλες, μπορούν να εμφανίζονται τα δεδομένα, που περιέχουν οι πίνακες. Στο Παράρτημα Γ, παρατίθενται όλα τα *Scripts* που χρησιμοποιήθηκαν, για τη δημιουργία της βάσης και των πινάκων, ενώ στην Εικόνα 4.5 παρουσιάζεται ενδεικτικά ένα παράδειγμα, από τη δημιουργία του πίνακα *diamerismata*. Στη πρώτη σειρά του παραδείγματος ορίζεται ο *a/a* κάθε γραμμής, να είναι θετικός ακέραιος με εύρος (0 ~ 4294967295) [19]. Επίσης θα είναι πρωτεύον κλειδί και μοναδικός αριθμός (καθώς κάθε πίνακας πρέπει να έχει ένα μοναδικό στοιχείο εγγραφής, σε κάθε γραμμή) και τέλος θα αυξάνεται αυτόματα με κάθε εγγραφή στον πίνακα. Οι επόμενες τέσσερις σειρές του *Script*, έχουν τύπο δεδομένων *VARCHAR* (50), δηλαδή μπορούν να δεχθούν το πολύ μέχρι 50 χαρακτήρες, ενώ τα πεδία δεν μπορεί να είναι κενά. Η τελευταία γραμμή είναι τύπου *Datetime*, ώστε να δέχεται ημερομηνία-ώρα και επίσης το πεδίο δεν μπορεί να είναι κενό [20-21].

Ολοκληρώνοντας, πρέπει να επισημανθεί ότι όσον αφορά τις ώρες ενεργοποίησης, του ατομικού θερμοστάτη κάθε διαμερίσματος (πίνακας *diamerismata* στήλη *wres* (Εικόνα 4.3)), τα δεδομένα προστέθηκαν χειροκίνητα. Αυτό έγινε, καθώς όπως σχολιάστηκε και στην αρχή του δεύτερου κεφαλαίου, η συσκευή δεν καταγράφει αυτές τις ώρες. Επίσης, όλα τα δεδομένα του πίνακα, διακατέχονται από μια γενναία δόση υπερβολής. Η υπερβολή στα νούμερα σκοπό έχει, να κάνει πιο ευδιάκριτες τις διακυμάνσεις στα αντίστοιχα σχήματα, για την καλύτερη κατανόηση.



Εικόνα 4.4: Ρυθμίσεις των ενεργειών που θα μπορεί να εκτελεί ο χρήστης

```
CREATE TABLE diamerismata
(
  id INT UNSIGNED PRIMARY KEY NOT NULL UNIQUE AUTO_INCREMENT,
  h_id VARCHAR(50) NOT NULL,
  wres VARCHAR(50) NOT NULL,
  tank VARCHAR(50) NOT NULL,
  part VARCHAR(50) NOT NULL,
  date_created DATETIME NOT NULL
);
```

Εικόνα 4.5: Δημιουργία του πίνακα diamerismata

### 4.3 Flask framework

Το Flask framework (ή micro framework όπως πολλές φορές καλείται, από την προγραμματιστική κοινότητα) είναι ένα εργαλείο που βρίσκει εφαρμογή, για την κατασκευή διαδικτυακών εφαρμογών και χρησιμοποιεί τη γλώσσα προγραμματισμού Python [22]. Εξ' αρχής περιέχει βασικά εργαλεία για την κατασκευή (γι' αυτό ονομάζεται και micro), συγκριτικά με κάποιο άλλο αντίστοιχο εργαλείο, όπως π.χ. το Django, που περιέχει περισσότερα και πιο εξειδικευμένα [23]. Αυτό βέβαια δεν δημιουργεί κανένα πρόβλημα, καθώς μπορεί να υπάρξει ευελιξία, μιας και είναι επεκτάσιμο. Υπάρχει η δυνατότητα προσθήκης βιβλιοθηκών και εξωτερικών εργαλείων, που μπορούν να αποδώσουν την απαιτούμενη επιπλέον λειτουργικότητα. Μάλιστα πολλά από αυτά, έχουν σχεδιαστεί ειδικά για το Flask (π.χ. Flask SQLAlchemy [24]).

Ο κώδικας οργανώνεται σε μεθόδους, όπου κάθε μέθοδος επιστρέφει και την αντίστοιχη HTML σελίδα, για προβολή. Μια άλλη λειτουργία, που προσδίδει ευελιξία και υπάρχει στο αρχικό πακέτο του Flask, είναι τα Jinja templates. Πρόκειται ουσιαστικά για HTML αρχεία, με επιπλέον λειτουργικότητα, καθώς επιτρέπουν τη σύνταξη κώδικα, παρόμοιου με αυτόν της Python. Χρησιμοποιούν code blocks για συνθήκες, βρόχους, επεκτάσεις ή την οργάνωση ενός τμήματος του κώδικα. Είναι της μορφής «{% (block όνομα, συνθήκη, επέκταση ή βρόχος) όνομα ή τιμή %}» {% end(block όνομα, συνθήκη ή βρόχος) %}. Δηλαδή, άγκιστρα που ανοίγουν και κλείνουν, έχοντας στη αρχή και στο τέλος %. Ακόμα υπάρχει η δυνατότητα, να περαστούν μεταβλητές, από το back-end στο HTML αρχείο, μέσω των Jinja templates, χρησιμοποιώντας το όνομα μέσα σε διπλά άγκιστρα (π.χ. {{ myvar }}). Ένα χαρακτηριστικό παράδειγμα εμφανίζεται στην Εικόνα 4.6. Αν η τιμή της μεταβλητής είναι ίση με 1, στην ιστοσελίδα θα εμφανιστεί η τιμή της μεταβλητής, διαφορετικά θα εμφανιστεί το None. Τονίζεται, ότι στα code blocks, υπάρχει η ανάγκη εκκίνησης και τερματισμού του μπλοκ, ώστε να γίνεται αντιληπτό περί τίνος πρόκειται. Για παράδειγμα, {% for ...%} {% endfor %} ή {% if ... %} {% endif %} ή {% block somename %} {% endblock somename %}. Αντίθετα, η επέκταση δεν χρειάζεται το end (π.χ. {% extends "base.html" %}). Τα Jinja templates (HTML αρχεία) αποθηκεύονται σε φάκελο με όνομα templates, μέσα στο φάκελο της εφαρμογής. Ένας δεύτερος φάκελος που πρέπει να δημιουργηθεί στο φάκελο της εφαρμογής, είναι ο static, που θα περιέχει περιεχόμενο που δεν αλλάζει, όπως π.χ. σταθερές εικόνες και το CSS αρχείο [25-28].

```
{% if myvar == 1 %}
  {{ myvar }}
{% else %}
  {{ None }}
{% endif %}
```

Εικόνα 4.6: Παράδειγμα ενός Jinja template

```

from flask import Flask, render_template
app = Flask(__name__)

@app.route('/index_example/')
def index_example():
    myvar = "Paradeigma web efarmoghs"
    # myvar = 1
    return render_template('index.html', myvar=myvar)

if __name__ == '__main__':
    app.run(host='0.0.0.0')

```

Εικόνα 4.7: Παράδειγμα απλής web εφαρμογής με το Flask

Οι Εικόνες 4.7, 4.8 και 4.9 έρχονται να συνδυαστούν με όσα αναφέρθηκαν μέχρι στιγμής, παρουσιάζοντας ένα απλό παράδειγμα, μιας διαδικτυακής εφαρμογής. Αρχικά μέσω του τερματικού (terminal) στο PyCharm, γίνεται η εγκατάσταση του Flask (*pip install Flask*) και στη συνέχεια γίνεται εισαγωγή από το Flask, του Flask και render template. Έπειτα αρχικοποιείται το Flask (που εισήχθη), μέσω του αντικειμένου app, δημιουργώντας μια αναφορά για την συνολική Flask εφαρμογή. Το «Flask(\_\_name\_\_)» παίρνει αυτόματα, το όνομα του αρχείου που τρέχει. Εν συνέχεια, υλοποιείται μια μέθοδος με όνομα *def index\_example()*, όπου μέσα της περιέχει μια μεταβλητή με όνομα myvar. Στην πρώτη περίπτωση θα έχει κείμενο, ενώ στην άλλη την τιμή 1. Αυτό γίνεται καθώς παρουσιάζονται δυο παραδείγματα. Ακόμα, περιέχει το *return render\_template*, που σαν όρισμα παίρνει το HTML αρχείο, καθώς και τη μεταβλητή που θα μεταβιβάσει στο Jinja template (η μεταβλητή με τα μπλε γράμματα). Επιπρόσθετα, στη μέθοδο αποδίδεται επιπλέον λειτουργικότητα, μέσω του decorator *@app.route*. Αυτός, σαν όρισμα δέχεται το μονοπάτι από το οποίο θα κληθεί, για να εμφανίσει το περιεχόμενο στην ιστοσελίδα. Επίσης δέχεται και τη μέθοδο, με την οποία θα κληθεί. Αν αυτή δεν αναφέρεται, τότε by default είναι η GET, ωστόσο αποδεκτό είναι και το «*methods=["GET", "POST"]*».

Μέσω του *app.run*, γίνεται η εκκίνηση του sever και η διαθέσιμη πόρτα εξ ορισμού είναι η 5000. Στο πρώτο παράδειγμα η μεταβλητή myvar, θα περιέχει τον αριθμό 1 ενώ στο δεύτερο παράδειγμα θα περιέχει το κείμενο. Έτσι, στο φυλλομετρητή πληκτρολογώντας *http://127.0.0.1:5000/index\_example/* μέσω του *return render\_template*, θα επιστραφεί κάθε φορά, το αντίστοιχο παράδειγμα (Εικόνας 4.9). Ο έλεγχος για το τι περιεχόμενο θα εμφανίσει η σελίδα, γίνεται μέσω του code block και του Jinja template της Εικόνας 4.8. Στο code block περιέχεται η σύγκριση if, ενώ σε διπλά άγκιστρα η τιμή της μεταβλητής (το κείμενο της).

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Web app example</title>
</head>
<body>
  {% if myvar == 1 %}
    H metablhth einai ish me 1
  {% else %}
    {{ myvar }}
  {% endif %}
</body>
</html>

```

Εικόνα 4.8: HTML αρχείο και Jinja template



Εικόνα 4.9: Εμφάνιση των δυο αποτελεσμάτων του παραδείγματος στην ιστοσελίδα

Σε αυτή λογική βασίστηκε και η ανάπτυξη του κώδικα, για το σύστημα της παρούσα εργασίας, εμπλουτίζοντας τον φυσικά με περισσότερα εργαλεία. Ο στόχος ήταν η υλοποίηση μιας πιο λειτουργικής εφαρμογής, προσαρμοσμένης στα ζητούμενα και τις ανάγκες που τέθηκαν εξ αρχής.

#### 4.4 Σύνδεση της εφαρμογής με τη Βάση Δεδομένων

Η σύνδεση της εφαρμογής με τη βάση δεδομένων, έγινε μέσω του εργαλείου SQLAlchemy [29] και συγκεκριμένα του Flask SQLAlchemy [24]. Χρησιμοποιώντας το εργαλείο αυτό, δύναται να συνταχθούν SQL ερωτήματα, χωρίς τον κλασικό κώδικα SQL. Το SQLAlchemy θα μετατρέψει τα γραφόμενα σε Python, σε SQL ερωτήματα. Τέτοιου είδους εργαλεία, ονομάζονται ORM (Object-Relational Mapping) και αυτό που κάνουν, είναι να αντιστοιχίζουν (mapping) τις κλάσεις που έχουν δημιουργηθεί με κάποια αντικείμενα σε κάποια άλλα αντικείμενα (relational), δηλαδή κάποιο πίνακα. Η χρήση του απαιτεί αρχικά την εγκατάσταση του, κι εν συνεχεία κάποιες αρχικοποιήσεις. Αυτές αφορούν την αποκατάσταση της σύνδεσης με τη βάση, μέσω της εντολής «*app.config['SQLALCHEMY\_DATABASE\_URI']*» και την επικοινωνία με τη βάση, μέσω της εντολής «*my\_db = SQLAlchemy(app)*». Αφού γίνει αυτό, το επόμενο βήμα αφορά τη δημιουργία κλάσεων, μια για κάθε πίνακα. Η κάθε κλάση, πρέπει να έχει το όνομα του πίνακα και για να λειτουργήσει σε συνδυασμό με το υπόλοιπο πρόγραμμα, χρειάζεται μια πολλαπλή κληρονομικότητα [30-32]. Για το λόγο αυτό, κληρονομεί από τις κλάσεις «*my\_db.Model*» και «*UserMixin*». Στην Εικόνα 4.10 παρουσιάζεται ενδεικτικά η κλάση για τον πίνακα *diamerismata*, ενώ στο Παράρτημα Δ παρατίθενται όλες οι υλοποιηθείσες κλάσεις.

Όπως παρουσιάστηκε και στην ενότητα 4.2, η κλάση θα έχει ίδια ονόματα και τύπους δεδομένων με του εκάστοτε πίνακα. Ωστόσο, αναφορά πρέπει να γίνει για το πεδίο *date\_created*, καθώς αυτό παίρνει ως default τιμή, την συγκεκριμένη ημερομηνία και ώρα, όταν καλείται η κλάση. Για τον προσδιορισμό της τρέχουσας ημερομηνίας-ώρας, χρησιμοποιείται από τη βιβλιοθήκη *datetime* το *datetime.now* [33].

Στις Εικόνες 4.11 έως 4.14, παρουσιάζονται κάποια παραδείγματα για το πώς γίνεται η αποστολή και η λήψη των δεδομένων (εγγραφών) με τη Β.Δ., μέσω της κλάσης *users* και του SQLAlchemy.

```
class Diamerismata(my_db.Model, UserMixin):
    id = my_db.Column(INTEGER(unsigned=True), primary_key=True, unique=True, nullable=False)
    date_created = my_db.Column(my_db.DateTime, default=datetime.now)
    wres = my_db.Column(my_db.String(50), nullable=False)
    h_id = my_db.Column(my_db.String(50), nullable=False)
    tank = my_db.Column(my_db.String(50), nullable=False)
    part = my_db.Column(my_db.String(50), nullable=False)
```

Εικόνα 4.10: Κλάση *Diamerismata* για τον πίνακα *diamerismata*



```
class Users(my_db.Model, UserMixin):
    id = my_db.Column(INTEGER(unsigned=True), primary_key=True, unique=True, nullable=False)
    username = my_db.Column(my_db.String(30), unique=True, nullable=False)
    email = my_db.Column(my_db.String(30), nullable=False)
    password = my_db.Column(my_db.String(100), nullable=False)
    first_name = my_db.Column(my_db.String(50), nullable=False)
    last_name = my_db.Column(my_db.String(50), nullable=False)
    floor = my_db.Column(TINYINT(unsigned=True), nullable=False) #unable not null
    house_number = my_db.Column(TINYINT(unsigned=True), nullable=False)
    date_created = my_db.Column(my_db.DateTime, default=datetime.now)
    admin_key = my_db.Column(SMALLINT(unsigned=True), nullable=False)
```

Εικόνα 4.11: Κλάση Users για τον πίνακα users

```
new_user = Users(username=username, email=email, password=encrypt_password, first_name=first_name,
                 last_name=last_name, floor=floor, house_number=house_number, admin_key=admin_key)
my_db.session.add(new_user)
my_db.session.commit()
```

Εικόνα 4.12: Νέα εγγραφή στον πίνακα users της βάσης δεδομένων

```
user = Users.query.filter_by(email=email).first_or_404()
```

Εικόνα 4.13: Ερώτημα στη Β.Δ για τον πίνακα users με κριτήριο το email και επιστροφή της πρώτης εγγραφής που θα βρεθεί

```
admin = Users.query.filter_by(admin_key=12123).all()
```

Εικόνα 4.14: Ερώτημα στη Β.Δ για τον πίνακα users με κριτήριο το admin\_key και επιστροφή όλων των εγγραφών που θα βρεθούν

Έγινε ειδική μνεία στις ενότητες 4.3 και 4.4 όσον αφορά τα Jinja templates, την επιστροφή αυτών μέσω του `return render_template` καθώς και τον τρόπο λειτουργίας του SQLAlchemy, μέσω των κλάσεων. Ακολουθήθηκε αυτή η πρακτική, ώστε να μην αναλύονται συνεχώς, μιας και τα συγκεκριμένα εμφανίζονται αρκετές φορές, στις περισσότερες (αν όχι όλες) μεθόδους, που θα αναλυθούν στη συνέχεια. Εκεί, θα γίνεται ειδική αναφορά μόνο αν προκύπτει κάποια καινούρια λειτουργία, διαφορετικά θα γίνεται μια απλή επισήμανση. Έτσι οι επόμενες ενότητες που ακολουθούν, αναλύουν βήμα-βήμα την διαδικτυακή εφαρμογή και πως αυτή υλοποιήθηκε προγραμματιστικά. Η παρουσίαση γίνεται όπως ο χρήστης (ο ένοικος της πολυκατοικίας), θα βιώσει την εμπειρία μέσω της περιήγησης του στον ιστότοπο (αρχικά πρέπει να δημιουργήσει λογαριασμό, έπειτα να συνδεθεί κ.ο.κ.).

#### 4.5 Δημιουργία λογαριασμού, είσοδος-έξοδος και επαναφορά κωδικού πρόσβασης

Το πρώτο πράγμα που πρέπει να γίνει, ώστε να είναι διαθέσιμη η εφαρμογή, είναι να “σηκωθεί στον αέρα” ο sever. Αυτό γίνεται με την εντολή της Εικόνας 4.15 και η πρόσβαση γίνεται μέσω της πόρτας 5000. Στο τερματικό του PyCharm εμφανίζεται η Εικόνα 4.16, που περιέχει κάποια στοιχεία για τον server. Αυτά είναι, το Debug mode σε on, η προειδοποίηση ότι είναι ένας server για ανάπτυξη (σχολιάζεται στο πέμπτο κεφάλαιο) και οι διευθύνσεις απ’ όπου μπορεί να κληθεί η εφαρμογή. Η πρώτη αφορά τον τοπικό υπολογιστή που εκκίνησε ο server και η δεύτερη τη διεύθυνση που μπορεί κάποιος να έχει πρόσβαση, μέσα από το τοπικό δίκτυο. Ακόμα, μιας και είναι ρυθμισμένος σε Debug mode, αυτό σημαίνει ότι θα εμφανίζονται τα λάθη, που προκύπτουν από την πλευρά του server, στην οθόνη. Κατά τη διάρκεια ανάπτυξης του κώδικα, αυτό είναι αποδεκτό και συνδυάζεται άρδην με το Debugger PIN. Ο κωδικός αυτός μπορεί να χρησιμοποιηθεί στο φυλλομετρητή και να ενεργοποιήσει ένα τερματικό, όπου πληκτρολογώντας εκεί τις εντολές, να γίνεται επί τόπου η αποσφαλμάτωση του κώδικα. Ωστόσο, με την ολοκλήρωση της συγγραφής του κώδικα της εφαρμογής, το Debug mode πρέπει να αλλάξει σε False, καθώς ο χρήστης δεν είθισται να βλέπει τυχόν σφάλματα.

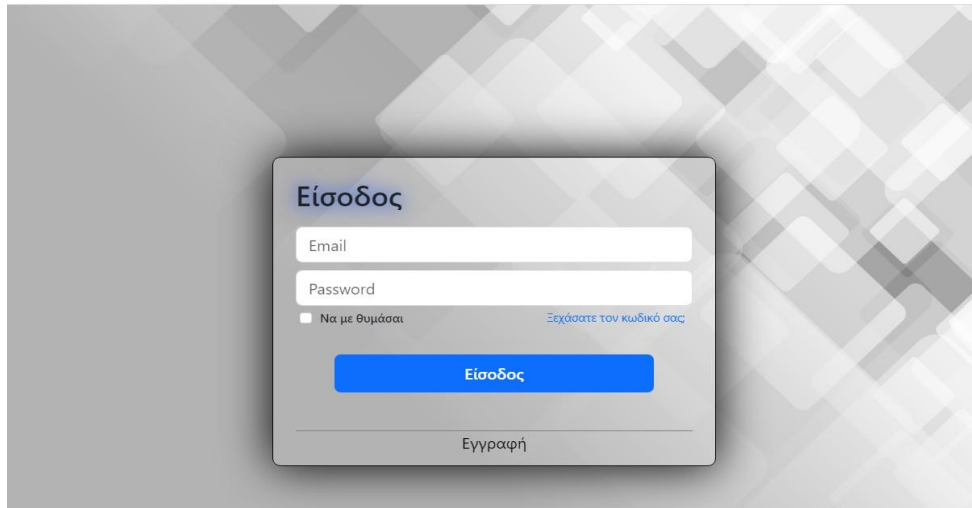
Πληκτρολογώντας την διεύθυνση `http://192.168.2.11:5000/` στο φυλλομετρητή, εμφανίζεται η σελίδα (Εικόνα 4.17) που περιέχει την Είσοδο στην εφαρμογή, πληκτρολογώντας το email και τον κωδικό πρόσβασης. Ακόμα, υπάρχει η επιλογή «Να με θυμάσαι» και η επιλογή, «Ξεχάσατε τον κωδικό σας;». Όλα αυτά έχουν ουσία, αφού πρώτα κάνει την «Εγγραφή» (Εικόνα 4.18). Η κλήση της σελίδας εγγραφής γίνεται με GET, ενώ η αποστολή των στοιχείων με POST. Όλα τα πεδία είναι απαραίτητο να συμπληρωθούν (Εικόνα 4.19), εκτός από το Admin\_Key. Το πεδίο αφορά το διαχειριστή, όπου πληκτρολογώντας τον κωδικό (δόθηκε ο 12123) αποκτά επιπλέον δικαιώματα στη σελίδα. Η μέθοδος που χειρίζεται την Εγγραφή, είναι η `def signup()` (Εικόνα 4.20) και το HTML αρχείο ονομάζεται `signup.html`. Για τη δημιουργία των πεδίων προς πληκτρολόγηση, χρησιμοποιήθηκαν από τη βιβλιοθήκη Flask\_wtf [34] το FlaskForm και από τη βιβλιοθήκη wtforms [35] τα StringField, IntegerField, BooleanField, SubmitField, TextAreaField, SelectField, PasswordField. Ακόμα, από τη βιβλιοθήκη wtforms.validators τα DataRequired, NumberRange, Email, Length, EqualTo και ValidationError [35]. Αρχικά στην πλευρά του back-end, δημιουργείται η κλάση με όνομα `SignupForm(FlaskForm)` και όρισμα το FlaskForm. Μέσα ορίζονται τα αντίστοιχα πεδία, που θα εμφανίζονται στη σελίδα και γίνεται το validation (επικύρωση) των στοιχείων, όπου χρειάζεται. Όλες

```
if __name__ == '__main__':
    app.run(debug=True, host='0.0.0.0')
```

Εικόνα 4.15: Εκκίνηση του server

```
* Serving Flask app 'Thesis_Build_Flask'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on all addresses (0.0.0.0)
* Running on http://127.0.0.1:5000
* Running on http://192.168.2.11:5000
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 367-937-250
```

Εικόνα 4.16: Τερματικό του PyCharm κατά την εκκίνηση του server



Εικόνα 4.17: Παράθυρο εισόδου – εγγραφής στην εφαρμογή

οι κλάσεις για τα πεδία της εφαρμογής, παρατίθενται στο Παράρτημα Ε. Στην HTML σελίδα, δημιουργείται μια HTML φόρμα, που θα χρησιμοποιεί τη μέθοδο POST, για την αποστολή των δεδομένων. Δεν θα χρειαστεί να κάνει επικύρωση της εγκυρότητας των δεδομένων, μιας και αυτό γίνεται στο back-end. Μέσα στην HTML φόρμα, εισάγονται τα πεδία της κλάσης *SignupForm* και το *signupform.csrf\_token* που είναι μια μέθοδος του *FlaskForm*. Χρησιμοποιείται ώστε να περάσει στη σελίδα το *Secret Key* (Παράρτημα ΣΤ). Αυτό είναι μια δικλείδα ασφαλείας, όπου με κάποιο εσωτερικό

## Σελίδα εγγραφής νέου χρήστη

Όνομα	Επώνυμο
<input type="text" value="First name"/>	<input type="text" value="Last name"/>
Email	
<input type="text" value="Email"/>	
Όνομα χρήστη	
<input type="text" value="Username"/>	
Κωδικός	Επιβεβαίωση κωδικού
<input type="text" value="Password"/>	<input type="text" value="Confirm password"/>
Όροφος	Αριθμός διαμερίσματος
<input type="text" value="Floor"/>	<input type="text" value="House_id"/>
Admin_Key	
<input type="text" value="Admin_key"/>	
<input type="button" value="Αποστολή"/>	

Εικόνα 4.18: Καρτέλα εγγραφής νέου χρήστη

**Σελίδα εγγραφής νέου χρήστη**

<b>Όνομα</b>	<b>Επώνυμο</b>
<input type="text" value="First name"/>	<input type="text" value="Last name"/>
• Υποχρεωτικό πεδίο	• Υποχρεωτικό πεδίο
<b>Email</b>	
<input type="text" value="Email"/>	
• Υποχρεωτικό πεδίο	
<b>Όνομα χρήστη</b>	
<input type="text" value="Username"/>	
• Υποχρεωτικό πεδίο	
<b>Κωδικός</b>	<b>Επιβεβαίωση κωδικού</b>
<input type="text" value="Password"/>	<input type="text" value="Confirm password"/>
• Υποχρεωτικό πεδίο	• Υποχρεωτικό πεδίο
<b>Όροφος</b>	<b>Αριθμός διαμερίσματος</b>
<input type="text" value="Floor"/>	<input type="text" value="House_id"/>
• Υποχρεωτικό πεδίο	• Υποχρεωτικό πεδίο
<b>Admin_Key</b>	
<input type="text" value="Admin_key"/>	
<input type="button" value="Αποστολή"/>	

Εικόνα 4.19: Υποχρεωτικά πεδία της φόρμας εγγραφής

τρόπο, επιβεβαιώνει ότι τα στοιχεία που στάλθηκαν προς το back-end, προήλθαν από τη συγκεκριμένη σελίδα και όχι από κάπου αλλού. Αφού πληκτρολογηθούν και σταλούν, πριν περάσουν στη βάση δεδομένων, πρέπει να γίνει το validation. Αυτό γίνεται σε όλα τα πεδία, ενώ για τα username και email, έχει οριστεί ότι θα πρέπει να είναι και μοναδικά. Για το σκοπό αυτό γίνεται ένα ερώτημα στη βάση (όπως περιεγράφηκε στην ενότητα 4.4), με τα στοιχεία που δόθηκαν και αν αυτά βρεθούν, τότε δεν ολοκληρώνεται η εγγραφή εμφανίζοντας το αντίστοιχο μήνυμα (Εικόνα 4.21). Αν όλα είναι σωστά τότε η εγγραφή ολοκληρώνεται, γίνεται ανακατεύθυνση στην σελίδα Εισόδου (Εικόνα 4.17) και στην πάνω πλευρά εμφανίζεται το μήνυμα «Επιτυχής εγγραφή!» (Εικόνα 4.22). Τα δεδομένα προς εγγραφή χρησιμοποιούν την κλάση Users και τον πίνακα Users. Στο σημείο αυτό πρέπει να ειπωθεί, ότι ο κωδικός που πληκτρολογήθηκε περνάει στη βάση, αφού πρώτα γίνει η κρυπτογράφηση του. Δεν πρέπει να είναι ορατός ούτε μέσα στη Β.Δ., για ευνόητους λόγους. Η διαδικασία κρυπτογράφησης, απαιτεί τη χρήση του Bcrypt από τη βιβλιοθήκη Flask-bcrypt [36].

```
@app.route('/signup/', methods=["GET", "POST"])
def signup():
```

Εικόνα 4.20: Μέθοδος def signup()

## Σελίδα εγγραφής νέου χρήστη

Όνομα	Επώνυμο
<input type="text" value="Επαμεινώνδας"/>	<input type="text" value="Κάκαλος"/>
Email	
<input type="text" value="cmmf211@hotmail.com"/>	
<ul style="list-style-type: none"> <li>To email <b>cmmf211@hotmail.com</b> χρησιμοποιείται!</li> </ul>	
Όνομα χρήστη	
<input type="text" value="smhnarxos_kakalos"/>	
<ul style="list-style-type: none"> <li>To username <b>smhnarxos_kakalos</b> χρησιμοποιείται!</li> </ul>	

Εικόνα 4.21: Μήνυμα μη επικύρωσης (validation) κάποιων πεδίων

Αν ο χρήστης δεν θυμάται τον κωδικό πρόσβασης του, από τη σελίδα της Εικόνας 4.17 μπορεί να επιλέξει το «Ξεχάσατε τον κωδικό σας;» και να ακολουθήσει τα βήματα. Αρχικά πρέπει να εισαγάγει το email, με το οποίο είχε κάνει την εγγραφή του (Εικόνα 4.23) και να πατήσει Αποστολή. Στη συνέχεια γίνεται ανακατεύθυνση στη σελίδα της Εικόνας 4.17 και εμφανίζεται το μήνυμα επιτυχούς αποστολής. Αν το email που δόθηκε, δεν υπάρχει στη Β.Δ., εμφανίζεται το αντίστοιχο μήνυμα της Εικόνας 4.24. Στο καταχωρημένο email, αποστέλλεται ένα link με διάρκεια ισχύς μισής ώρας (Εικόνα 4.25). Ακολουθώντας το σύνδεσμο, εμφανίζεται η σελίδα της Εικόνας 4.26, ώστε να εισαχθεί ο νέος κωδικός πρόσβασης.

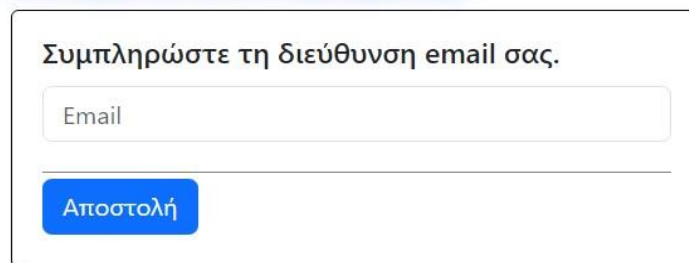
The image shows a login form titled "Είσοδος" (Login) overlaid on a background with a geometric pattern. At the top, a green notification box says "Επιτυχής εγγραφή!" (Successful registration!) with a close button. The login form contains the following elements:

- Input field for "Email"
- Input field for "Password"
- Checkbox labeled "Να με θυμάσαι" (Remember me)
- Link labeled "Ξεχάσατε τον κωδικό σας" (Forgot your password?)
- A blue button labeled "Είσοδος" (Login)
- A link at the bottom labeled "Εγγραφή" (Registration)

Εικόνα 4.22: Μήνυμα επιτυχούς εγγραφής στην εφαρμογή

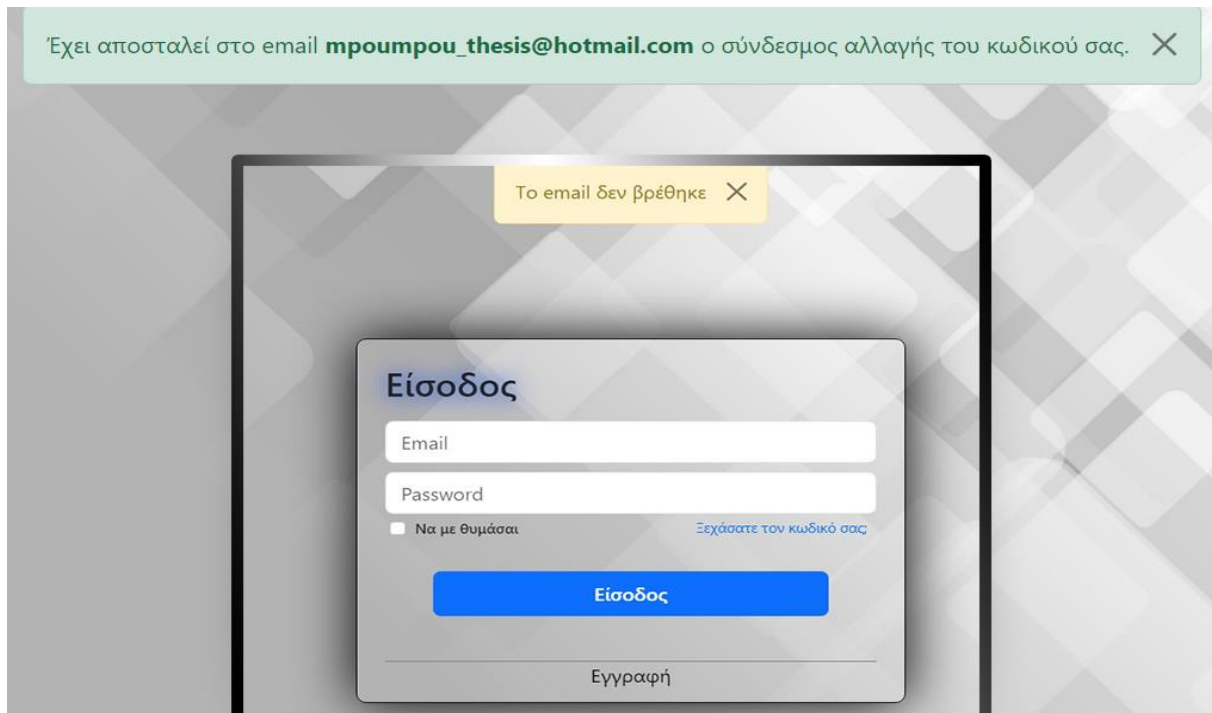
Όσα αναφέρθηκαν για τον ορισμό του νέου κωδικού πρόσβασης, προγραμματιστικά υποστηρίζονται από τις μεθόδους `def reset_password()` και `def reset_password_token(token)` και από τις HTML σελίδες `reset_passw_email.html` και `reset_password.html`. Μέσω της πρώτης μεθόδου, γίνεται η αποστολή του email με το σύνδεσμο. Το link που δημιουργείται περιέχει τυχαία κωδικοποιημένη συμβολοσειρά, που ένα τμήμα της είναι το `salt='7c2189ed0c264d49'` και το υλοποιεί το `URLSafeTimedSerializer` μέσω της βιβλιοθήκης `itsdangerous` [37]. Έχει οριστεί το `salt`, ώστε στη δεύτερη μέθοδο να γίνεται η σύγκριση και αν το link ταυτίζεται στο συγκεκριμένο κομμάτι, τότε να ακολουθεί ο ορισμός του νέου κωδικού. Τα πεδία ακολουθούν την ίδια λογική, σαν αυτά της εγγραφής. Το ίδιο ισχύει και για την αλλαγή της εγγραφής στη βάση, μόνο που τώρα δεν χρειάζεται να γίνει `add` αλλά μόνο `commit` (Εικόνα 4.12). Αυτό συμβαίνει μιας και η εγγραφή υπάρχει, κι έτσι θα γίνει ενημέρωση των δεδομένων της, όχι νέα εγγραφή.

### Αλλαγή κωδικού πρόσβασης

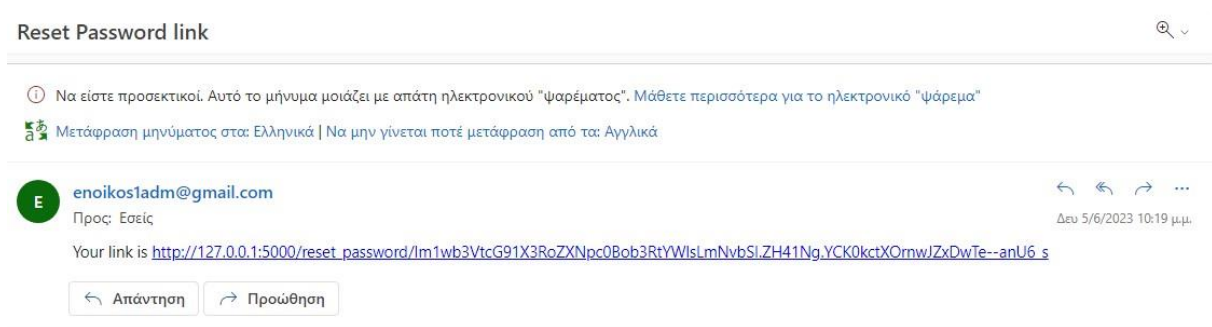


The screenshot shows a web form titled "Αλλαγή κωδικού πρόσβασης". The form contains a text input field labeled "Email" and a blue button labeled "Αποστολή". The text above the input field reads "Συμπληρώστε τη διεύθυνση email σας."

Εικόνα 4.23: Εισαγωγή του email για την αλλαγή του κωδικού πρόσβασης



Εικόνα 4.24: Μήνυμα επιτυχούς ή μη αποστολής του email για τον ορισμό νέου κωδικού πρόσβασης



Εικόνα 4.25: Email που περιέχει το σύνδεσμο για τον ορισμό νέου κωδικού πρόσβαση

### Ορισμός νέου κωδικού πρόσβασης

**Κωδικός**

**Επιβεβαίωση κωδικού**

[Αποστολή](#)

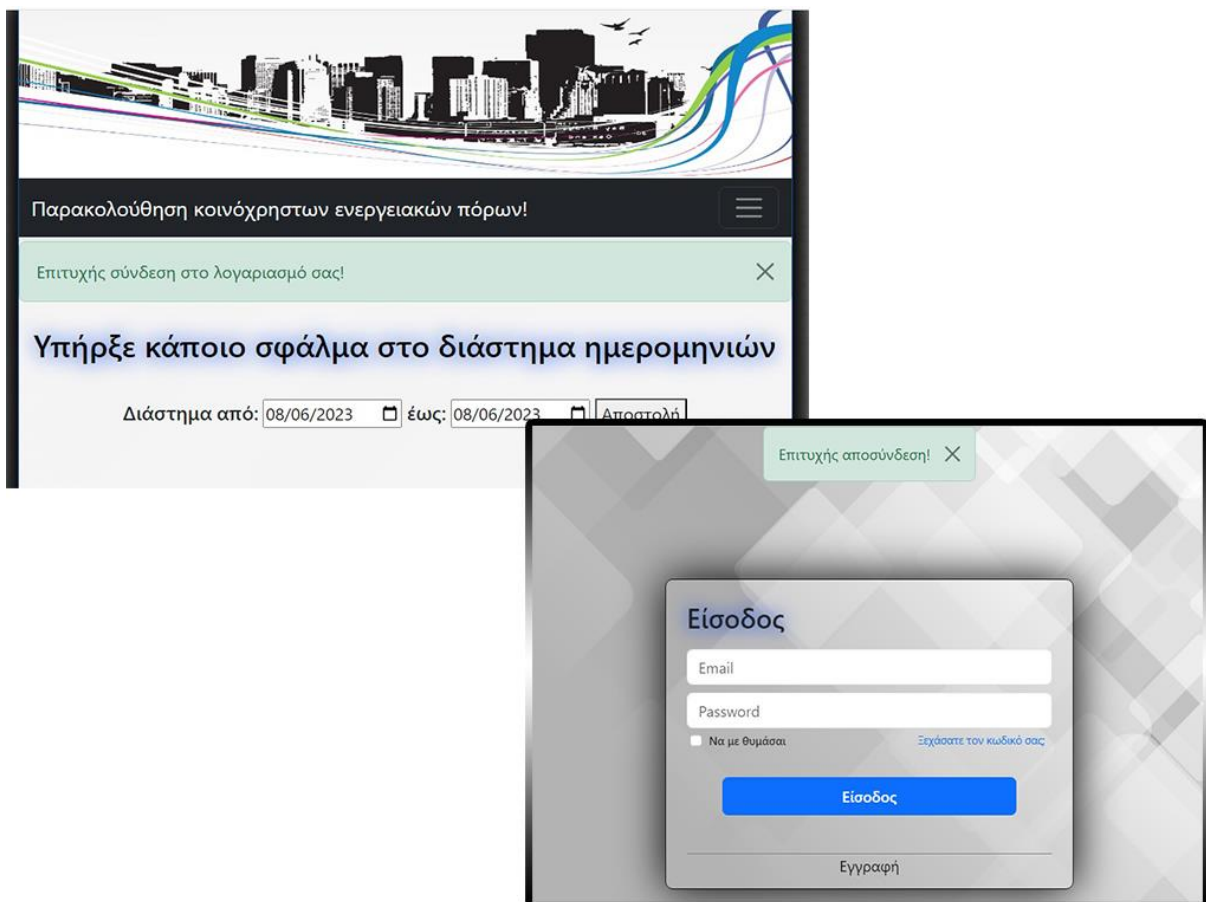
Εικόνα 4.26: Ορισμός νέου κωδικού πρόσβασης

Όσον αφορά την αποστολή του email, αυτή γίνεται με τη βοήθεια του Flask-email [38-40]. Απαιτείται να γίνουν οι απαραίτητες αρχικοποιήσεις (Παράρτημα ΣΤ) και αμέσως μετά είναι έτοιμο προς χρήση. Η κάθε διαδικασία αποστολής, περιέχει το κυρίως μέρος «msg» και την αποστολή «mail.send(msg)». Το κυρίως μέρος αποτελείται από το θέμα «subject» του email, τον αποστολέα «sender», τον παραλήπτη «recipients» και φυσικά το κείμενο «body». Στην περίπτωση που οι παραλήπτες είναι περισσότεροι του ενός, τότε απαιτείται η διεργασία να ξεκινάει με «with app.app\_context()». Στην Εικόνα 4.27 αποτυπώνονται όσα αναφέρθηκαν.

```
with app.app_context(): # polloi paralhptes
    for a in admin:
        msg = Message(subject="Thema",
                      sender=app.config.get("MAIL_USERNAME"),
                      recipients=[a.email], #paralhptes
                      body="Keimeno tou email")
        mail.send(msg)
```

Εικόνα 4.27: Διαδικασία αποστολής email με πολλαπλούς παραλήπτες

Τέλος, εφόσον ο χρήστης χρησιμοποιήσει τα σωστά διαπιστευτήρια του και πατήσει «Είσοδος», εισέρχεται στην Αρχική σελίδα της εφαρμογής (Εικόνα 4.28 και 4.38). Αν δε, επιλέξει και το «Να με θυμάσαι», τότε η εφαρμογή θα τον “θυμάται” για μια ώρα. Αυτό, σε περίπτωση που κλείσει καταλάθος, το πρόγραμμα περιήγησης του. Το εργαλείο που επιλέχθηκε να διευθετεί αυτό το ζήτημα, είναι το Flask-login [41]. Χειρίζεται τα Sessions στο παρασκήνιο με ένα πιο αποδοτικό τρόπο. Οργανώνει τις συνδέσεις μεταξύ του browser και του server, ενώ παράλληλα δίνει την ευελιξία στον προγραμματιστή, να χρησιμοποιεί απλές εντολές, ώστε να επιτυγχάνεται ο σκοπός. Χαρακτηριστικό παράδειγμα είναι ότι, για να θυμάται το χρήστη (μέθοδος *def login()*) αρκεί να χρησιμοποιηθούν τα στοιχεία του και το `remember_me` «`login_user(user,remember_me)`». Ενώ όταν ο χρήστης κάνει αποσύνδεση από τη σελίδα (Εικόνα 4.28) (μέθοδος *def logout()*), στο προγραμματιστικό κομμάτι, αρκεί η εντολή «`logout_user`». Το Flask-login αναλαμβάνει να διακόψει τη σύνδεση, καθαρίζοντας τα sessions κλπ.

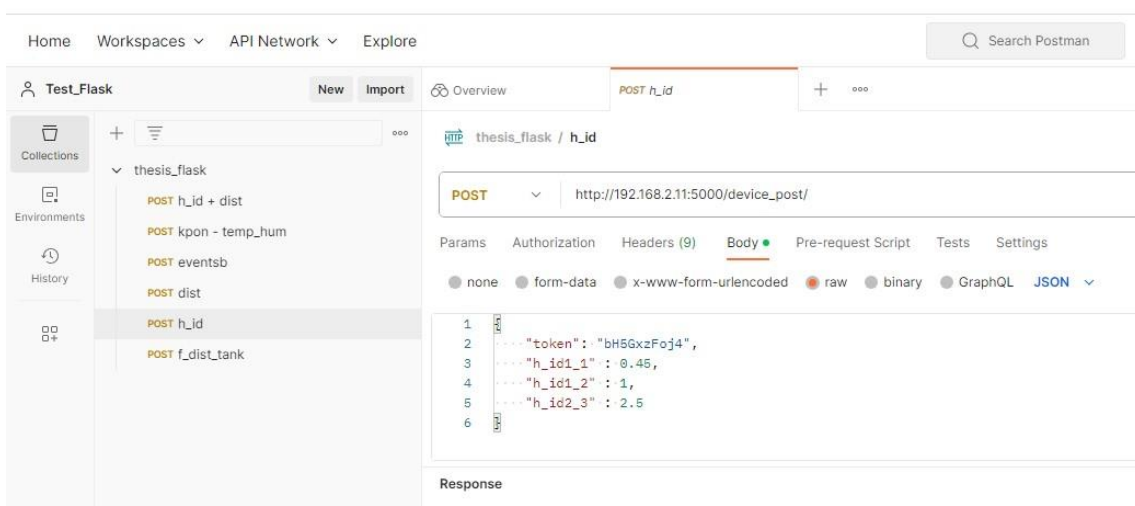


Εικόνα 4.28: Επιτυχής σύνδεση και αποσύνδεση από την εφαρμογή



#### 4.6 Αποστολή δεδομένων από τη συσκευή και επιτήρηση της

Η μέθοδος *def device\_post*, καλείται μόνο με τη μέθοδο POST και είναι υπεύθυνη για την λήψη και επεξεργασία των δεδομένων από τη συσκευή, καθώς και για την εγγραφή αυτών στη Β.Δ. Δέχεται ένα JSON αρχείο και αρχικά ελέγχει αν μέσα του περιέχεται το «token : bH5GxzFoj4». Το token είναι ένα ζεύγος κλειδιού-τιμής, κάτι σαν ταυτότητα, που ορίστηκε με σκοπό να αυξήσει την ασφάλεια, στη λήψη των δεδομένων. Έτσι μόνο όταν αυτό περιέχεται, το πρόγραμμα μπαίνει μέσα στη συνθήκη if, ώστε να γίνουν οι κατάλληλες ενέργειες στα δεδομένα. Πριν όμως γίνει αυτό, στη ροή του κώδικα υπάρχουν δυο μεταβλητές, η «check\_uic\_h» και «check\_uic\_m». Κάθε φορά που γίνεται POST με το σωστό token, σε αυτές φορτώνονται η ώρα και τα λεπτά, αντίστοιχα, της συγκεκριμένης χρονικής στιγμής. Οι μεταβλητές αυτές, χρησιμοποιούνται για την επιτήρηση της συσκευής. Όσον αφορά τα δεδομένα, ανάλογα το ζεύγος κλειδιού-τιμής, γίνονται και οι ανάλογες ενέργειες. Για την μέτρηση της δεξαμενής πριν και μετά την πλήρωση, τα δεδομένα που φθάνουν, στρογγυλοποιούνται σε ένα δεκαδικό ψηφίο και γίνεται η εγγραφή τους στη βάση. Για τη μέτρηση της δεξαμενής κάθε έξι ώρες, όταν φθάσουν τα δεδομένα, στρογγυλοποιούνται και πάλι σε ένα δεκαδικό και σηκώνεται μια σημαία. Η ίδια λογική ακολουθείται και όταν φτάσουν τα δεδομένα των ωρών. Τότε, όταν και οι δυο σημαίες γίνουν “1”, πραγματοποιείται η εγγραφή στη βάση. Στις Εικόνες 4.29 έως 4.31 παρουσιάζονται τα δεδομένα των ωρών των διαμερισμάτων και των λίτρων πετρελαίου. Πως φθάνουν στο server και πως εμφανίζονται στη βάση δεδομένων. Για το συγκεκριμένο παράδειγμα (αποστολή ωρών) αλλά και όπου αλλού απαιτήθηκε για την αποσφαλμάτωση του κώδικα (σε επίπεδο λήψης POST JSON αρχείων), χρησιμοποιήθηκε το εργαλείο Postman [42]. Τα ίδια ισχύουν και για την εσωτερική-εξωτερική θερμοκρασία-υγρασία. Όσον αφορά τα συμβάντα, αυτά καταγράφονται στη βάση και την αμέσως επόμενη στιγμή, αποστέλλεται ένα email σε όλους τους χρήστες, ενημερώνοντας τους για το αντίστοιχο συμβάν (Εικόνα 4.32). Επιπρόσθετα, πάλι μέσω email, ενημερώνονται αυτή τη φορά μόνο οι διαχειριστές, για την ανάγκη πλήρωσης της δεξαμενής πετρελαίου, όταν η στάθμη πέσει κάτω από ένα συγκεκριμένο όριο (Εικόνα 4.33).



Εικόνα 4.29: Αποστολή ωρών μέσω του Postman

## Κεφάλαιο 4

```
routes.py [C:\Users\Chriss\Desktop\Thesis_building] - Local
Local
Local x
(venv) PS C:\Users\Chriss\Desktop\Thesis_building\venv\src> python .\app_run.py
* Serving Flask app 'Thesis_Build_Flask'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on all addresses (0.0.0.0)
* Running on http://127.0.0.1:5000
* Running on http://192.168.2.11:5000
Press CTRL+C to quit
Λίτρα δεξαμενής= 660.0 λίτρα
192.168.2.3 - - [18/May/2023 12:21:51] "POST /device_post/ HTTP/1.1" 200 -
Όρες: Διαμέρισμα 1_1= 0.45, Διαμέρισμα 1_2= 1, Διαμέρισμα 2_3= 2.5
192.168.2.11 - - [18/May/2023 12:21:57] "POST /device_post/ HTTP/1.1" 200 -

```

Εικόνα 4.30: Εμφάνιση δεδομένων των λίτρων δεξαμενής και των ωρών των διαμερισμάτων στο τερματικό του PyCharm

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator

Query 1 insert\_row\_diame... Create\_eventsb eventsb Create\_diamerismata diamerismata x

1 • SELECT \* FROM flask\_db\_thesis.diamerismata;

id	h_id	wres	tank	part	date_created
1	h_id1_1	0.45	660	1/4	2023-05-18 12:21:58
2	h_id1_2	1	660	1/4	2023-05-18 12:21:58
3	h_id2_3	2.5	660	1/4	2023-05-18 12:21:58
	NULL	NULL	NULL	NULL	NULL

Result Grid Filter Rows: Edit: Export/Import: Wrap Cell Content:

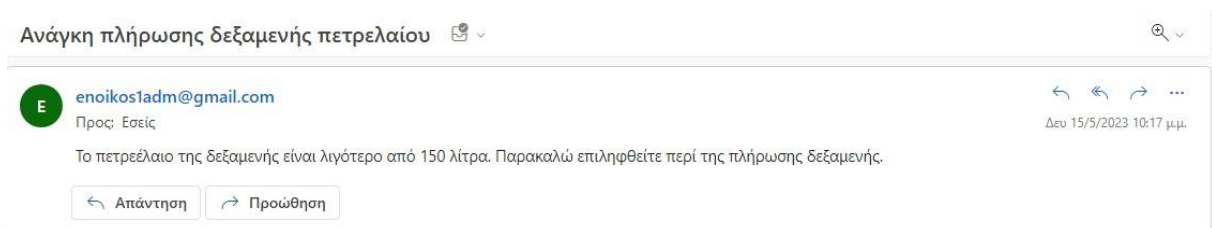
Table: diamerismata

Columns: id int UN AI PK

Εικόνα 4.31: Εγγραφή και εμφάνιση των δεδομένων στη MySQL βάση

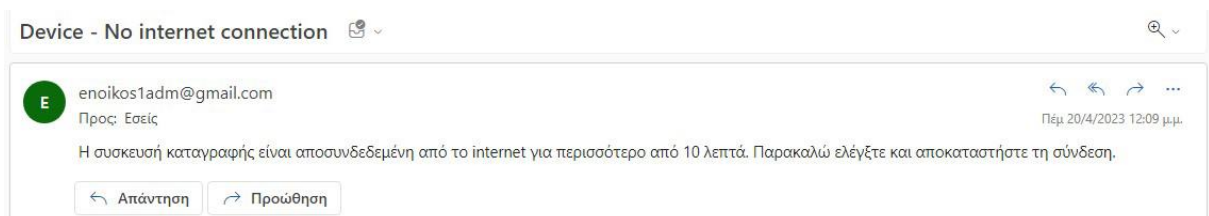


Εικόνα 4.32: Εισερχόμενο email μετά την ενεργοποίηση του αισθητήρα καπνού



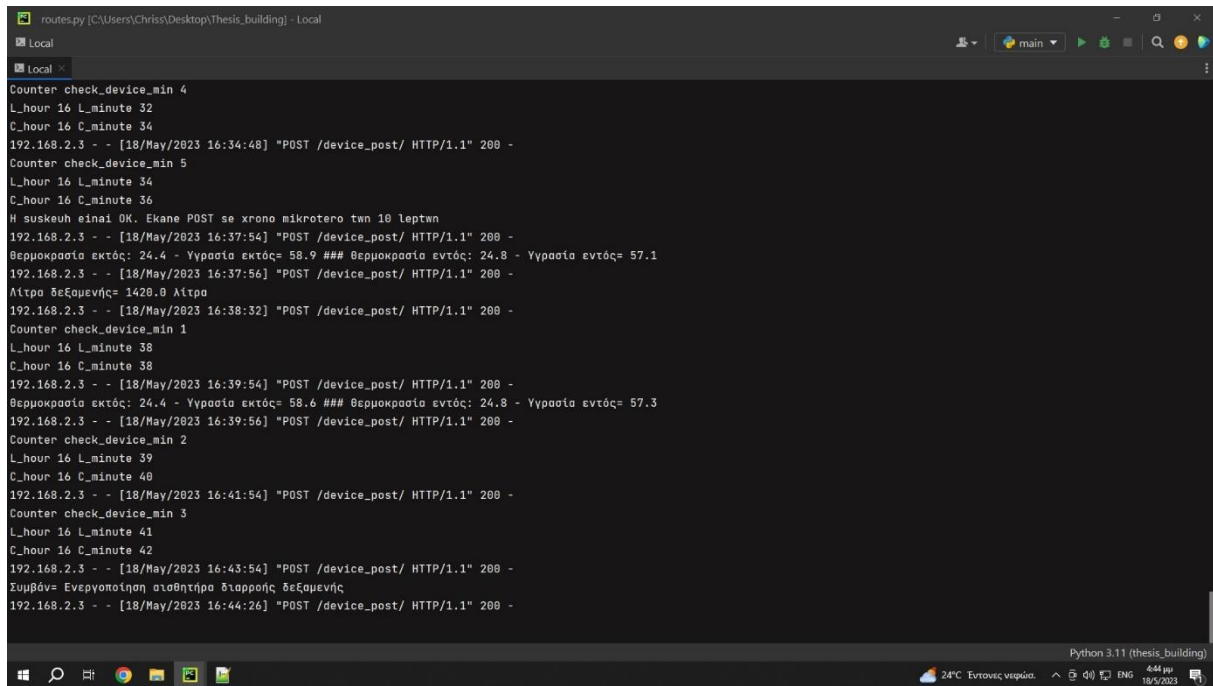
Εικόνα 4.33: Email προς τους διαχειριστές για την ανάγκη πλήρωσης της δεξαμενής

Η μέθοδος `def interval_task`, βοηθά στην επιτήρηση της συσκευής καταγραφής. Αποκτά επιπλέον λειτουργικότητα μέσω του decorator `@scheduler.task`, όπου ένα εκ των ορισμάτων του, είναι τα λεπτά (`minutes=5`). Μπορεί να γίνει η παρομοίωση του με ένα timer, όπου κάθε 5 λεπτά κάνει υπερχείλιση. Όταν λοιπόν κάνει και μπει στη μέθοδο, αυξάνεται μια μεταβλητή κατά 1. Όταν αυτή γίνει ίση με δυο, συγκρίνεται η τρέχουσα χρονική στιγμή με τις μεταβλητές «`check_uic_h`» και «`check_uic_m`», που περιέχουν τις χρονικές πληροφορίες της τελευταίας φοράς που έκανε POST, η συσκευή. Έτσι αν η διαφορά είναι μεγαλύτερη των 10 λεπτών, τότε θεωρείται ότι η συσκευή βρίσκεται εκτός σύνδεσης και αποστέλλεται το αντίστοιχο email στους διαχειριστές (Εικόνα 4.34) [43-45]. Τέλος στις Εικόνες 4.35 και 4.36, αποτυπώνονται μέσα από το τερματικό του PyCharm, κάποιες διεργασίες που έλαβαν χώρα.



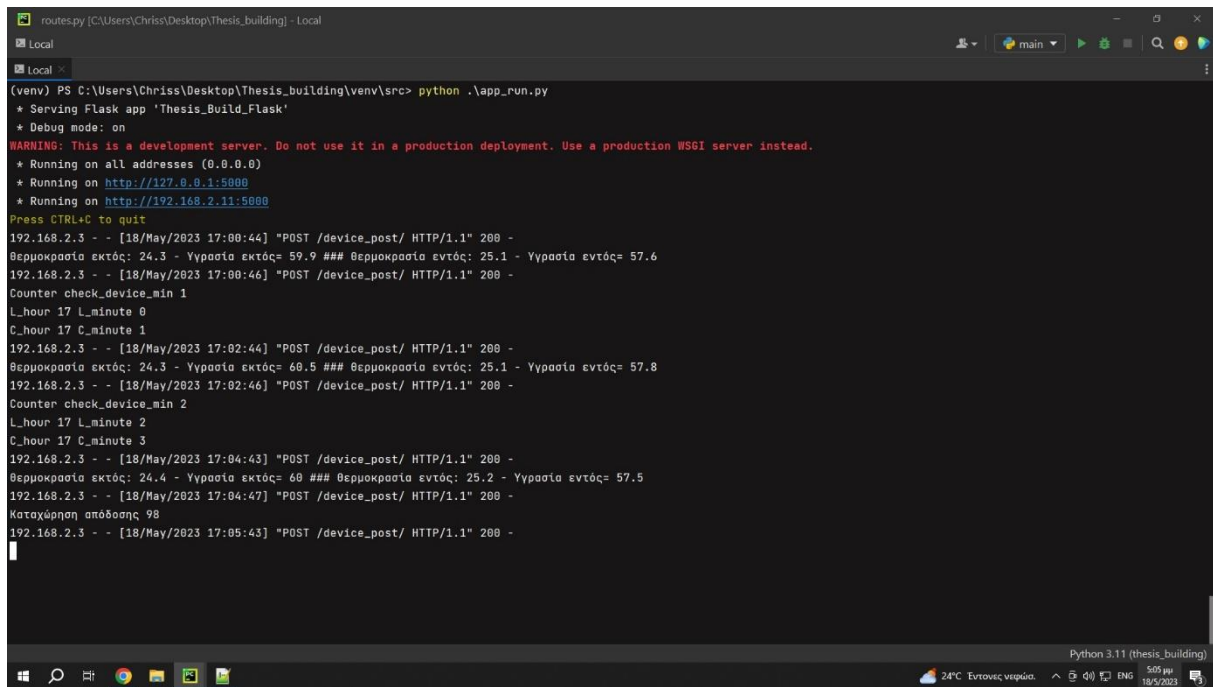
Εικόνα 4.34: Email προς τους διαχειριστές ότι η συσκευή βρίσκεται εκτός σύνδεσης

## Κεφάλαιο 4



```
routes.py [C:\Users\Chris\Desktop\Thesis_building] - Local
Local
Counter check_device_min 4
L_hour 16 L_minute 32
C_hour 16 C_minute 34
192.168.2.3 - - [18/May/2023 16:34:48] "POST /device_post/ HTTP/1.1" 200 -
Counter check_device_min 5
L_hour 16 L_minute 34
C_hour 16 C_minute 36
Η συσκευή είναι OK. Έκανε POST σε χρόνο μικρότερο των 10 λεπτών
192.168.2.3 - - [18/May/2023 16:37:54] "POST /device_post/ HTTP/1.1" 200 -
Θερμοκρασία εκτός: 24.4 - Υγρασία εκτός= 58.9 ### Θερμοκρασία εντός: 24.8 - Υγρασία εντός= 57.1
192.168.2.3 - - [18/May/2023 16:37:56] "POST /device_post/ HTTP/1.1" 200 -
Αίτριο δεξαμενής= 1420.0 λίτρα
192.168.2.3 - - [18/May/2023 16:38:32] "POST /device_post/ HTTP/1.1" 200 -
Counter check_device_min 1
L_hour 16 L_minute 38
C_hour 16 C_minute 38
192.168.2.3 - - [18/May/2023 16:39:54] "POST /device_post/ HTTP/1.1" 200 -
Θερμοκρασία εκτός: 24.4 - Υγρασία εκτός= 58.6 ### Θερμοκρασία εντός: 24.8 - Υγρασία εντός= 57.3
192.168.2.3 - - [18/May/2023 16:39:56] "POST /device_post/ HTTP/1.1" 200 -
Counter check_device_min 2
L_hour 16 L_minute 39
C_hour 16 C_minute 40
192.168.2.3 - - [18/May/2023 16:41:54] "POST /device_post/ HTTP/1.1" 200 -
Counter check_device_min 3
L_hour 16 L_minute 41
C_hour 16 C_minute 42
192.168.2.3 - - [18/May/2023 16:43:54] "POST /device_post/ HTTP/1.1" 200 -
Συμβόλε= Ενεργοποίηση αισθητήρα διαρροής δεξαμενής
192.168.2.3 - - [18/May/2023 16:44:26] "POST /device_post/ HTTP/1.1" 200 -
Python 3.11 (thesis_building)
484 μμ
18/5/2023
```

Εικόνα 4.35: Εμφάνιση κάποιων διεργασιών μέσω του τερματικού (1)



```
routes.py [C:\Users\Chris\Desktop\Thesis_building] - Local
Local
(venv) PS C:\Users\Chris\Desktop\Thesis_building\venv\src> python .\app_run.py
* Serving Flask app 'Thesis_Build_Flask'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on all addresses (0.0.0.0)
* Running on http://127.0.0.1:5000
* Running on http://192.168.2.11:5000
Press CTRL+C to quit
192.168.2.3 - - [18/May/2023 17:00:44] "POST /device_post/ HTTP/1.1" 200 -
Θερμοκρασία εκτός: 24.3 - Υγρασία εκτός= 59.9 ### Θερμοκρασία εντός: 25.1 - Υγρασία εντός= 57.6
192.168.2.3 - - [18/May/2023 17:00:46] "POST /device_post/ HTTP/1.1" 200 -
Counter check_device_min 1
L_hour 17 L_minute 0
C_hour 17 C_minute 1
192.168.2.3 - - [18/May/2023 17:02:44] "POST /device_post/ HTTP/1.1" 200 -
Θερμοκρασία εκτός: 24.3 - Υγρασία εκτός= 60.5 ### Θερμοκρασία εντός: 25.1 - Υγρασία εντός= 57.8
192.168.2.3 - - [18/May/2023 17:02:46] "POST /device_post/ HTTP/1.1" 200 -
Counter check_device_min 2
L_hour 17 L_minute 2
C_hour 17 C_minute 3
192.168.2.3 - - [18/May/2023 17:04:43] "POST /device_post/ HTTP/1.1" 200 -
Θερμοκρασία εκτός: 24.4 - Υγρασία εκτός= 60 ### Θερμοκρασία εντός: 25.2 - Υγρασία εντός= 57.5
192.168.2.3 - - [18/May/2023 17:04:47] "POST /device_post/ HTTP/1.1" 200 -
Καταχώρηση απόδοσης 98
192.168.2.3 - - [18/May/2023 17:05:43] "POST /device_post/ HTTP/1.1" 200 -
Python 3.11 (thesis_building)
505 μμ
18/5/2023
```

Εικόνα 4.36: Εμφάνιση κάποιων διεργασιών μέσω του τερματικού (2)

## 4.7 Αρχική σελίδα

Στην Αρχική σελίδα της εφαρμογής, κάθε φορά που αυτή καλείται, εμφανίζεται το διάγραμμα με την τρέχουσα ημερήσια καταγραφή ωρών, του διαμερίσματος για το οποίο έκανε εγγραφή ο χρήστης [46]. Επίσης, δύναται να υποστηρίξει και διάστημα ημερών. Η επιλογή γίνεται από τη φόρμα τύπου date στην HTML. Οι ημερομηνίες που επιλέγονται, επιστρέφουν στο back-end ώστε να γίνει το ερώτημα στη βάση, έχοντας ως συνθήκη τις ημερομηνίες και το h\_id (διακριτικό) του διαμερίσματος. Αυτό (h\_id) δημιουργείται δυναμικά, μέσα από τη σύνδεση του χρήστη στην εφαρμογή. Είναι ο συνδυασμός του ορόφου και του αριθμού διαμερίσματος, που εισήγαγε κατά την εγγραφή του. Έτσι, πρώτα γίνεται ένα ερώτημα στη βάση, ώστε να επιστραφεί ο όροφος και το διαμέρισμα. Για να γίνει αυτό, χρησιμοποιείται το «current\_user» που το προσφέρει το Flask-login. Για παράδειγμα, το ερώτημα:

```
x = Users.query.filter_by(floor=current_user.floor,
house_number=current_user.house_number).first_or_404()
```

ζητάει από τη βάση, να επιστρέψει την πρώτη εγγραφή (ή error 404 εφόσον δεν βρεθεί εγγραφή), με τον όροφο και τον αριθμό διαμερίσματος, του τρέχοντος χρήστη. Μετά τη συνένωση, θα προκύψει ένας συνδυασμός της μορφής «h\_id1\_1». Γνωρίζοντας λοιπόν το h\_id και το διάστημα ημερών, γίνεται ένα ερώτημα στον πίνακα Diamerimsta. Για κάθε ημερομηνία μέσω ενός βρόχου for, τα δεδομένα που επιστρέφουν, αυτά των ωρών μαζί με την ημερομηνία, προστίθενται ως tuple σε μια λίστα. Για το άθροισμα των συνολικών ωρών και του ποσοστού επί του συνόλου, χρησιμοποιούνται δυο μεταβλητές ως μετρητές. Εκεί προστίθενται σε κάθε ερώτημα, τα αντίστοιχα δεδομένα. Για το ποσοστό, γίνεται η διαίρεση των ωρών του ενός διαμερίσματος, προς τις συνολικές ώρες όλων των διαμερισμάτων. Με το πέρας του βρόχου, όλα τα δεδομένα μεταφέρονται στο front-end κομμάτι (Jinja templates) μέσω των μεταβλητών, από το return render\_template. Επίσης δυναμικά μεταφέρεται και ο τίτλος της σελίδας, ανάλογα αν θα είναι για μια ημερομηνία ή διάστημα ημερών. Στην Εικόνα 4.37 παρατίθενται τα SQL ερωτήματα και τα αποτελέσματα στη Β.Δ για μια ημερομηνία (1/3/2023).

```
1
2
3 • SELECT * FROM flask_db_thesis.diamerimsta where DATE(date_created) between "2023-03-01" and "2023-03-01" and h_id="h_id1_1";
4
5
```

id	h_id	wres	tank	part	date_created
4	h_id1_1	0.3	500	1/4	2023-03-01 06:00:01
7	h_id1_1	2	490	2/4	2023-03-01 12:00:01
10	h_id1_1	1	470	3/4	2023-03-01 18:00:01
13	h_id1_1	4	430	4/4	2023-03-01 23:59:00

```
2
3 • SELECT sum(wres) FROM flask_db_thesis.diamerimsta where DATE(date_created)="2023-03-01" and h_id = "h_id1_1";
4
5
```

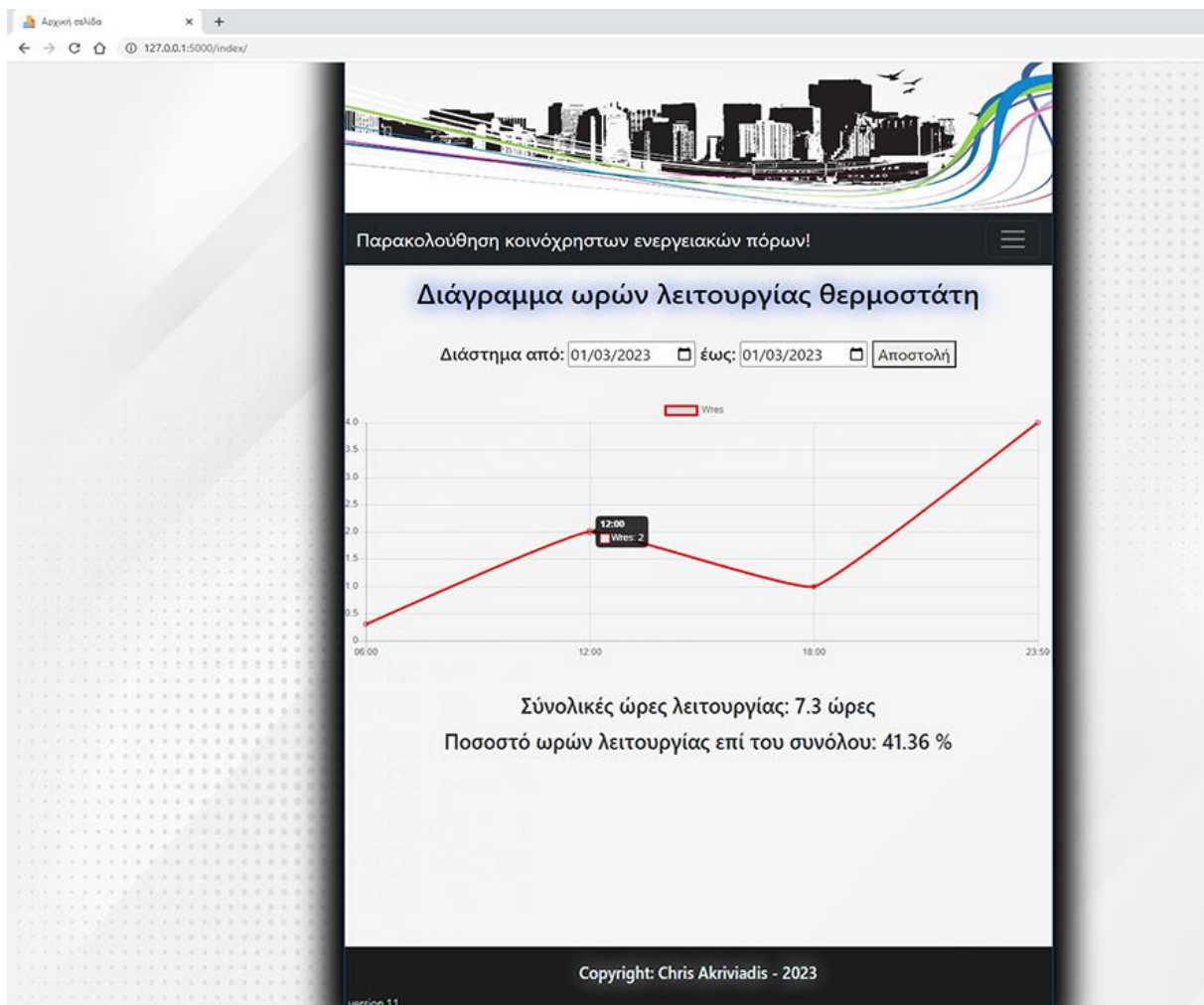
sum(wres)
7.3

```
2
3 • SELECT sum(wres) FROM flask_db_thesis.diamerimsta where DATE(date_created)="2023-03-01";
4
5
```

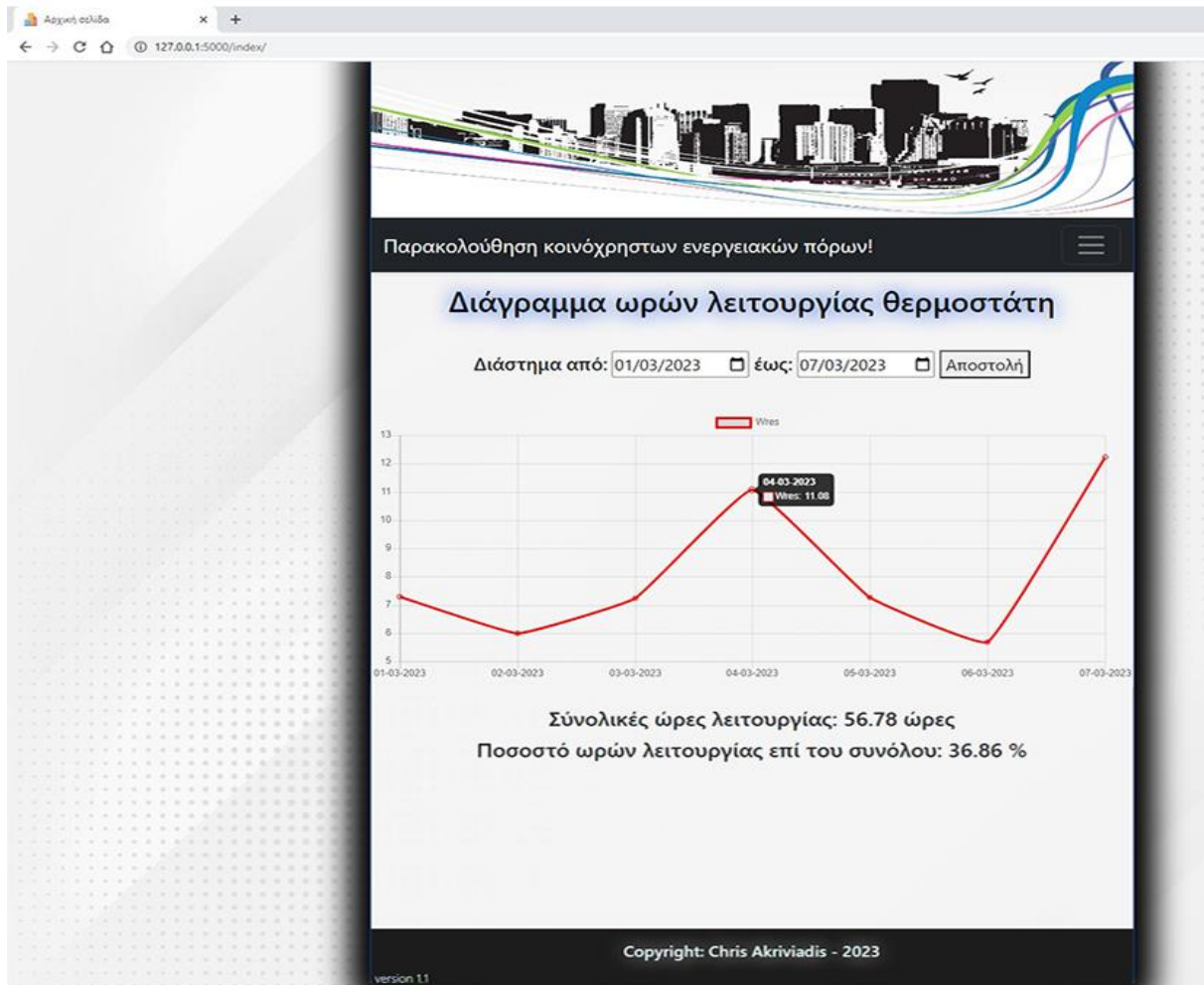
sum(wres)
17.650000000000002

Εικόνα 4.37: SQL ερωτήματα στη βάση για την 1/3/2023

Στην Εικόνα 4.38, παρουσιάζονται τα αντίστοιχα δεδομένα στην Αρχική σελίδα, σε μορφή διαγράμματος. Όπως προκύπτει υπάρχει ταύτιση των αποτελεσμάτων (Εικόνα 4.37 και Εικόνα 4.38), κάτι που σημαίνει ότι το σύστημα λειτουργεί σωστά (ποσοστό ωρών επί του συνόλου =  $(7,3/17,65)*100 = 41,36$ ). Επιπρόσθετα, στην Εικόνα 4.39, παρατίθενται τα δεδομένα, για το διάστημα ημερών από 1/3/2023 έως 7/3/2023. Να σημειωθεί πως αν υπάρξει κάποιο πρόβλημα στο διάστημα ημερομηνιών, ο server δεν “σκάει” αλλά το σφάλμα πιάνεται με try – except (Εικόνα 4.28). Η μέθοδος της Αρχικής σελίδας είναι η `def index()` και περιέχει έναν ακόμα decorator, τον `@login_required` που υποστηρίζεται από το Flask-login. Αυτός χρησιμοποιείται όταν υπάρχει η θέληση, κάποιες σελίδες να εμφανίζονται, μόνο όταν ο χρήστης συνδεθεί στην ιστοσελίδα. Το HTML αρχείο είναι το `index.html`, κι εκεί υπάρχει μια πιο εμφανή παρουσία των Jinja templates. Στην αρχή υπάρχει το «`{% extends "base_html.html" %}`» όπου επεκτείνει την `base_html.html` σελίδα, με ότι υπάρχει στη συνέχεια. Αυτή είναι μια ιδιότητα των Jinja templates, όπου ένα αρχείο μπορεί να επιδράσει σε κάποιο άλλο. Στη συγκεκριμένη περίπτωση, το `base_html.html` περιέχει τον βασικό κορμό μιας HTML σελίδας και τις απαιτούμενες συνδέσεις. Αν δεν γινόταν επέκταση του αρχείου στην `index.html`, θα έπρεπε να γραφούν όλα όσα περιέχει και το `base_html.html`. Ουσιαστικά η χρήση του `extends` έχει ως αποτέλεσμα, κοινά σημεία κώδικα να γράφονται μια φορά (να μην επαναλαμβάνονται), κι έπειτα να καλούνται σε κάποιο άλλο αρχείο. Με την ίδια λογική υπάρχουν και τα code blocks «`title`» και «`body`».



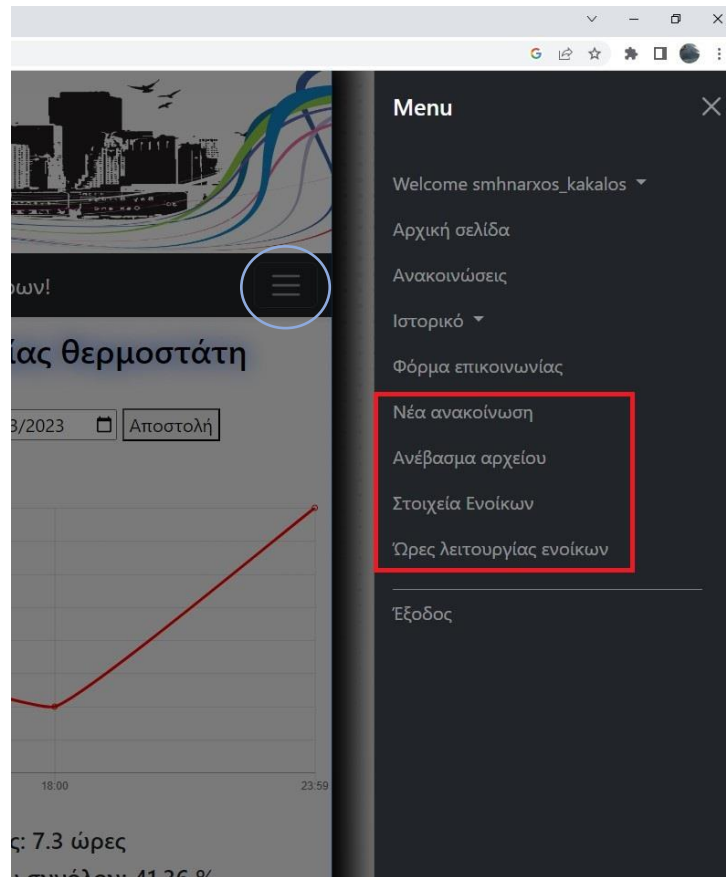
Εικόνα 4.38: Αρχική σελίδα της εφαρμογής (για την 1/3/2023)



Εικόνα 4.39: Αρχική σελίδα και εμφάνιση δεδομένων για διάστημα από 1/3/2023 έως 7/3/2023

#### 4.8 Μενού της εφαρμογής

Για το μενού της εφαρμογής, χρησιμοποιήθηκε και παραμετροποιήθηκε κατάλληλα ένα έτοιμο, από τη βιβλιοθήκη Bootstrap [47]. Εμφανίζεται στα δεξιά της σελίδας, είναι προσβάσιμο από όλες τις σελίδες, αλλά μόνο αν ο χρήστης έχει συνδεθεί. Ο έλεγχος γίνεται στην HTML από το «{% if current\_user.is\_authenticated %}», εργαλείο που συμπεριλαμβάνεται στο Flask-Login. Επίσης γίνεται και ο έλεγχος μέσω του «{% if current\_user.admin\_key == 12123 %}» για το αν ο συγκεκριμένος χρήστης είναι διαχειριστής, ώστε να του εμφανίσει επιπλέον επιλογές. Το μενού και η επιλογή του (σύμβολο στο μπλε κύκλο) παρουσιάζονται στην Εικόνα 4.40, ενώ οι επιπλέον επιλογές βρίσκονται μέσα στο κόκκινο πλαίσιο. Η κάθε επιλογή συνδέεται με μια μέθοδο στο back-end και όταν επιλεγεί ανακατευθύνεται εκεί. Όσον αφορά την επιλογή του Ιστορικού, μέσω του route ταξιδεύει κι ένας αριθμός (value\_id), που χρησιμοποιείται από το back-end για να ενεργήσει καταλλήλως, ενώ σε κάθε περίπτωση γίνεται ανακατεύθυνση στην μέθοδο που αφορά την κάθε επιλογή. Η πρώτη επιλογή είναι αυτή της Αρχικής σελίδας, που παρουσιάστηκε στην προηγούμενη ενότητα, ενώ η δεύτερη, «Ανακοινώσεις», παρουσιάζεται στην επόμενη. Τέλος, η εικόνα που βρίσκεται πάνω από τη λεζάντα «Παρακολούθηση κοινόχρηστων ενεργειακών πόρων!», αλλά και η λεζάντα, ανήκουν στο μενού και είναι ορατά σε όλες τις σελίδες. Πατώντας πάνω στην φωτογραφία [61], η εφαρμογή σε όποια σελίδα κι αν βρίσκεται, επιστρέφει στην Αρχική.



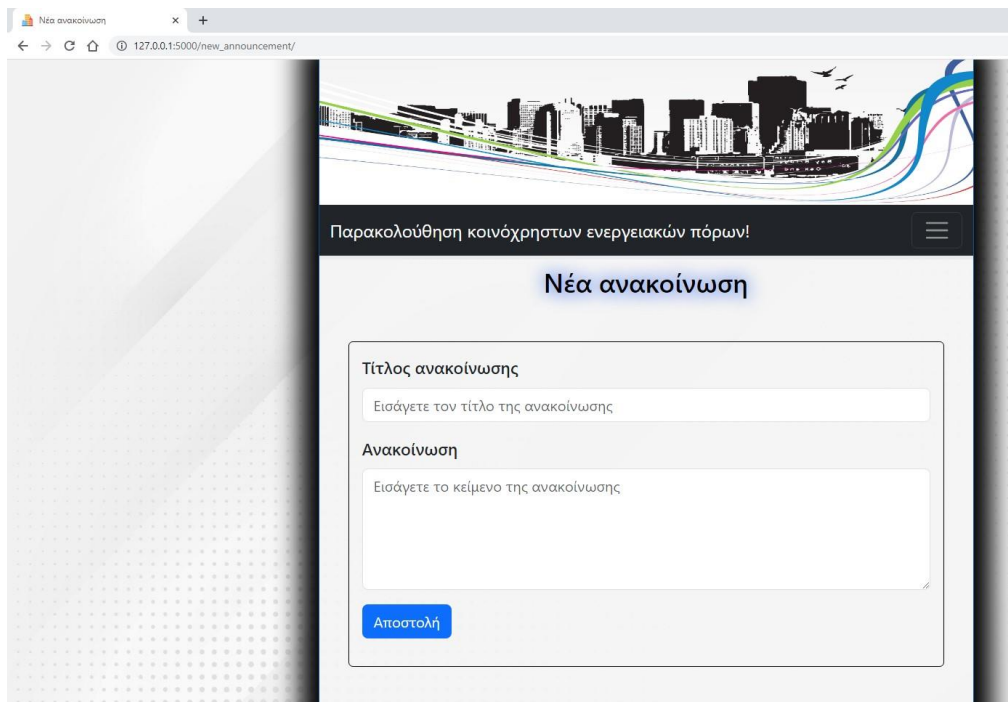
Εικόνα 4.40: Μενού της εφαρμογής

#### 4.9 Δημιουργία, τροποποίηση ή διαγραφή και προβολή Ανακοίνωσης

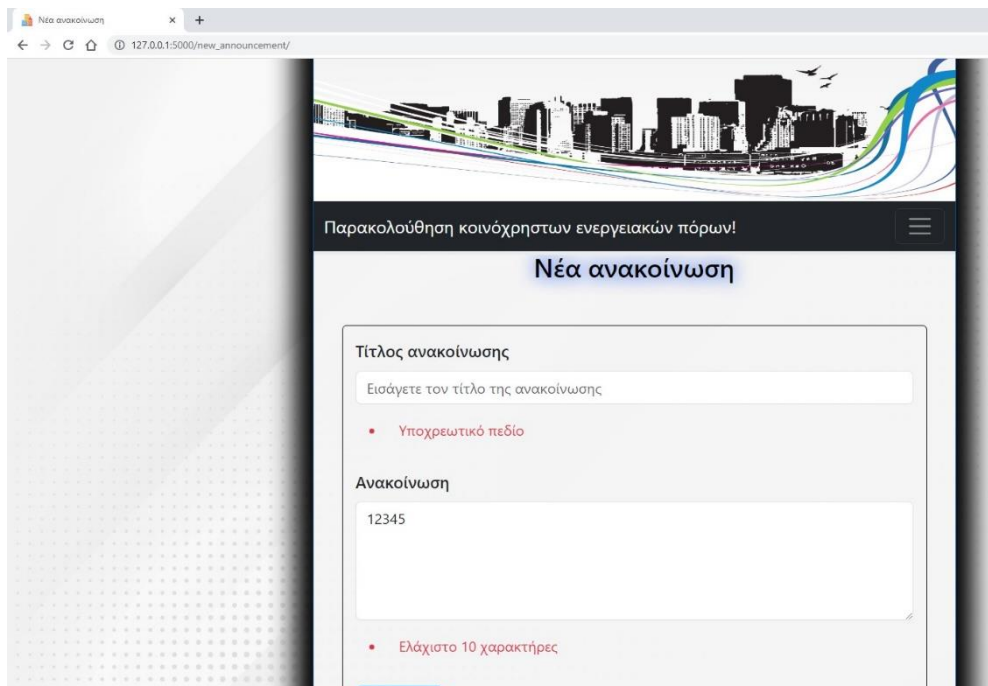
Η εφαρμογή υποστηρίζει την δημιουργία, επεξεργασία και διαγραφή μιας ανακοίνωσης μόνο από τους διαχειριστές, ενώ η προβολή της γίνεται από όλους. Η δημιουργία νέας ανακοίνωσης (Μενού και επιλογή «Νέα Ανακοίνωση») (Εικόνα 4.41), περιλαμβάνει τα πεδία του τίτλου και του κυρίως κειμένου της ανακοίνωσης. Στο HTML αρχείο *new\_announcement.html* δημιουργείται μια HTML φόρμα που θα στείλει με τη μέθοδο POST, τα δεδομένα χωρίς να τα κάνει επικύρωση. Αυτή γίνεται μέσω της κλάσης *NewAnnouncementForm(FlaskForm)*, απ' όπου και προέρχονται τα πεδία προς συμπλήρωση. Όταν πατηθεί το κουμπί «Αποστολή», στη μέθοδο *new\_announcement()*, αρχικά γίνεται ο έλεγχος αν τα δεδομένα έχουν περάσει την επικύρωση «*new\_announc\_form.validate\_on\_submit()*» (Εικόνα 4.42). Στη συνέχεια αυτά αντλούνται από τις φόρμες, μέσω του «*.. .data*» και φορτώνονται σε κάθε μεταβλητή αντίστοιχα. Έπειτα, γίνεται η εγγραφή τους στον πίνακα *new\_announcement* της Β.Δ. Πέρα από τον τίτλο και το κυρίως κείμενο, η εγγραφή περιλαμβάνει και το ποιος διαχειριστής συντάξε την ανακοίνωση. Αυτό γίνεται μέσω του «*current\_user.id*» όπου το *id* είναι ο *a/a* της εγγραφής χρήστη (Εικόνα 4.43). Για την σύνδεση της ανακοίνωσης με το χρήστη, στην κλάση *Users* (Παράρτημα Δ) στην τελευταία σειρά εμφανίζεται το *announcements*. Μέσω του «*my\_db.relationship(...)*» γίνεται μια σύνδεση σχέσης ένα προς πολλά και το *backref* είναι σαν να εμφανίζεται μια επιπλέον στήλη στον πίνακα, που να συνδέει την ανακοίνωση με το χρήστη. Αυτή η εικονική στήλη ονομάζεται *build\_admin*. Μετά την επιτυχή καταχώρηση της ανακοίνωσης, προβάλλεται το αντίστοιχο μήνυμα μέσω της ιδιότητας *flash* του *Flask*. Το παράθυρο και το χρώμα του, προέρχονται από το εργαλείο *Bootstrap*. Στη συνέχεια, γίνεται ανακατεύθυνση στη μέθοδο των ανακοινώσεων *def announcements()*, μέσω της



εντολής «return redirect(url\_for("announcements"))». Από εκεί θα επιστρέψει ο server τη σελίδα των Ανακοινώσεων (Εικόνα 4.44).



Εικόνα 4.41: Δημιουργία νέας ανακοίνωσης



Εικόνα 4.42: Έλεγχος επικύρωσης δεδομένων

```

@app.route('/new_announcement/', methods=["GET", "POST"]) # dhmiourgia neas anakoinwshs
@login_required
def new_announcement():
    new_announc_form = NewAnnouncementForm()

    if request.method == 'POST' and new_announc_form.validate_on_submit():
        annc_title = new_announc_form.announc_title.data
        annc_main = new_announc_form.announc_main.data

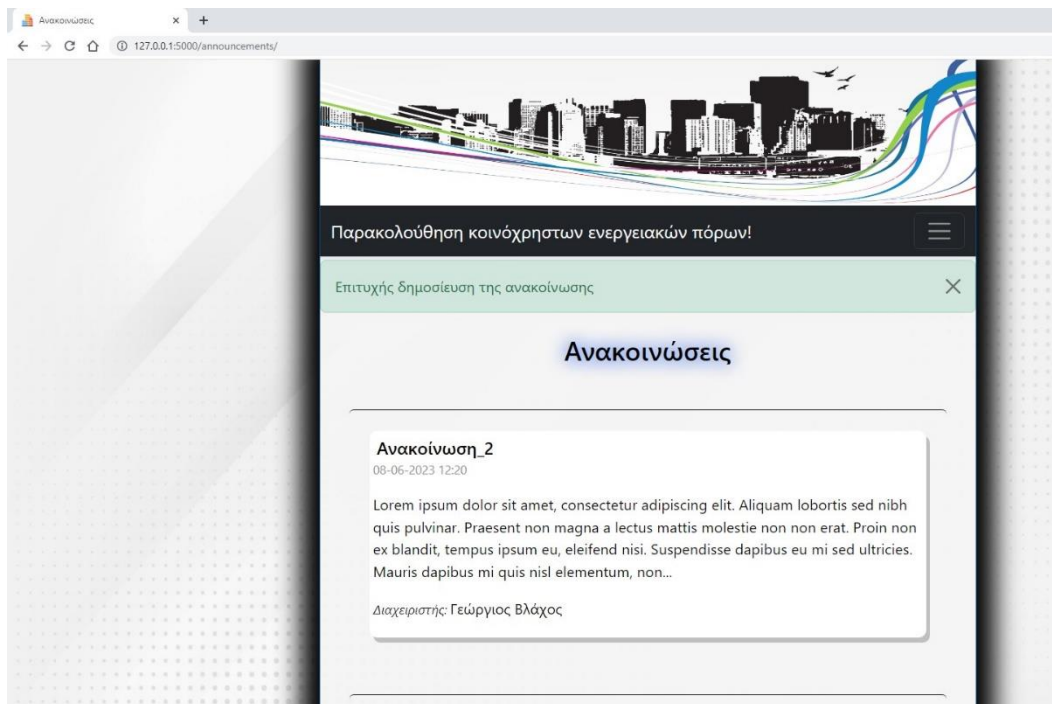
        new_full_announc = New_announcement(announc_title=annc_title, announc_main=annc_main, user_id=current_user.id)
        my_db.session.add(new_full_announc)
        my_db.session.commit()

        flash(f"Επιτυχής δημοσίευση της ανακοίνωσης", "success")
        return redirect(url_for("announcements"))

    return render_template('new_announcement.html', new_announc_form=new_announc_form,
                           page_title="Νέα ανακοίνωση")

```

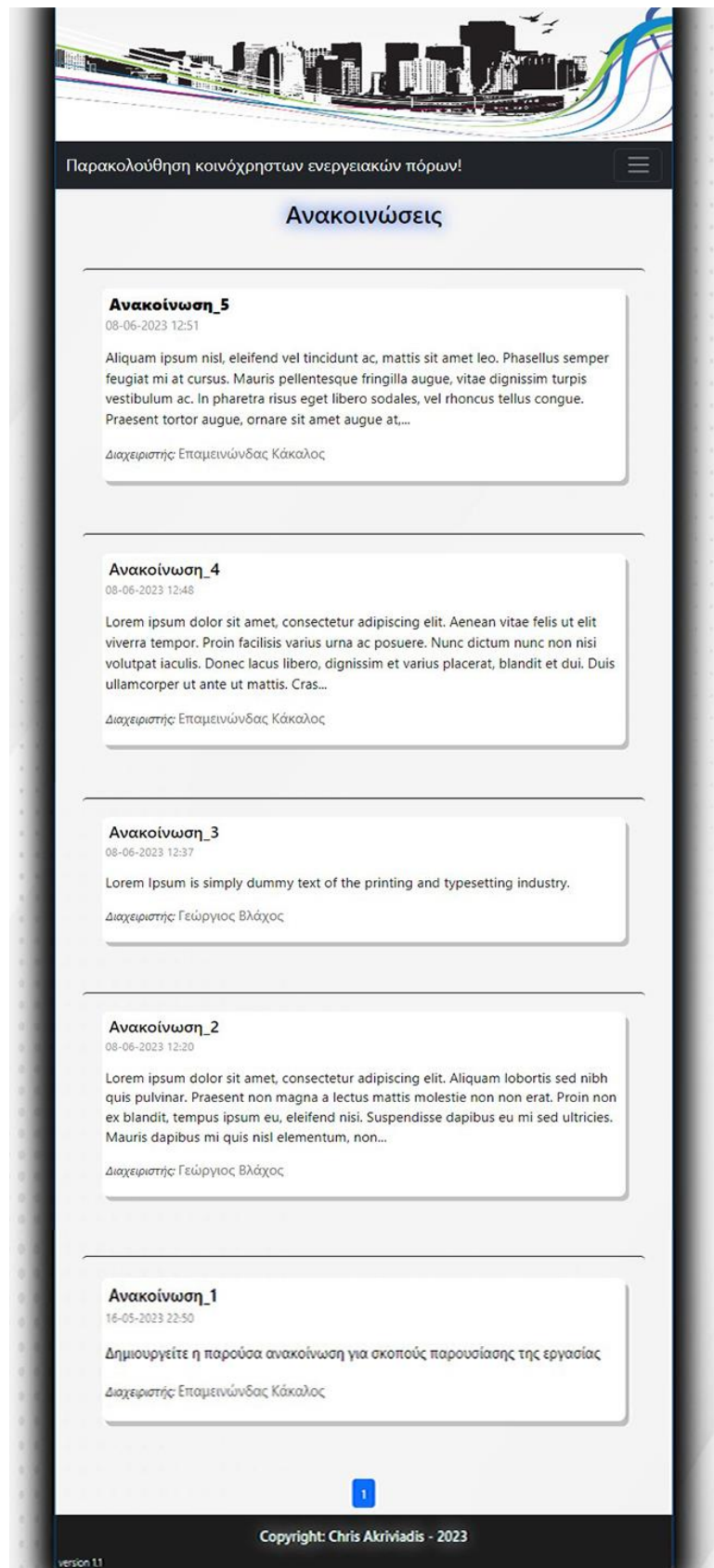
Εικόνα 4.43: Μέθοδος για τη δημιουργία νέας ανακοίνωσης



Εικόνα 4.44: Μήνυμα επιτυχούς δημοσίευσης της ανακοίνωσης και η σελίδα των Ανακοινώσεων

Στη σελίδα της Εικόνας 4.45 (HTML αρχείο *announcements.html*), εμφανίζονται συγκεντρωτικά όλες οι δημοσιευμένες ανακοινώσεις. Η σελιδοποίηση που γίνεται, επιτρέπει την ύπαρξη πέντε ανακοινώσεων ανά σελίδα. Αυτό επιτυγχάνεται με το SQLAlchemy, μέσω του `paginate` «`paginate(per_page=5, page=page)`», όταν γίνεται το ερώτημα στη βάση. Μέσω του φίλτρου «`truncate`» που υποστηρίζεται στα Jinja templates, γίνεται αποκοπή του κειμένου στους τριακοσίους χαρακτήρες, ώστε να μην εμφανίζεται όλο, αν σε περίπτωση είναι μακροσκελές. Το πλήρες κείμενο εμφανίζεται σε μια νέα σελίδα, πατώντας πάνω (`hover` στην Ανακοίνωση\_5) στην αντίστοιχη ανακοίνωση (Εικόνα

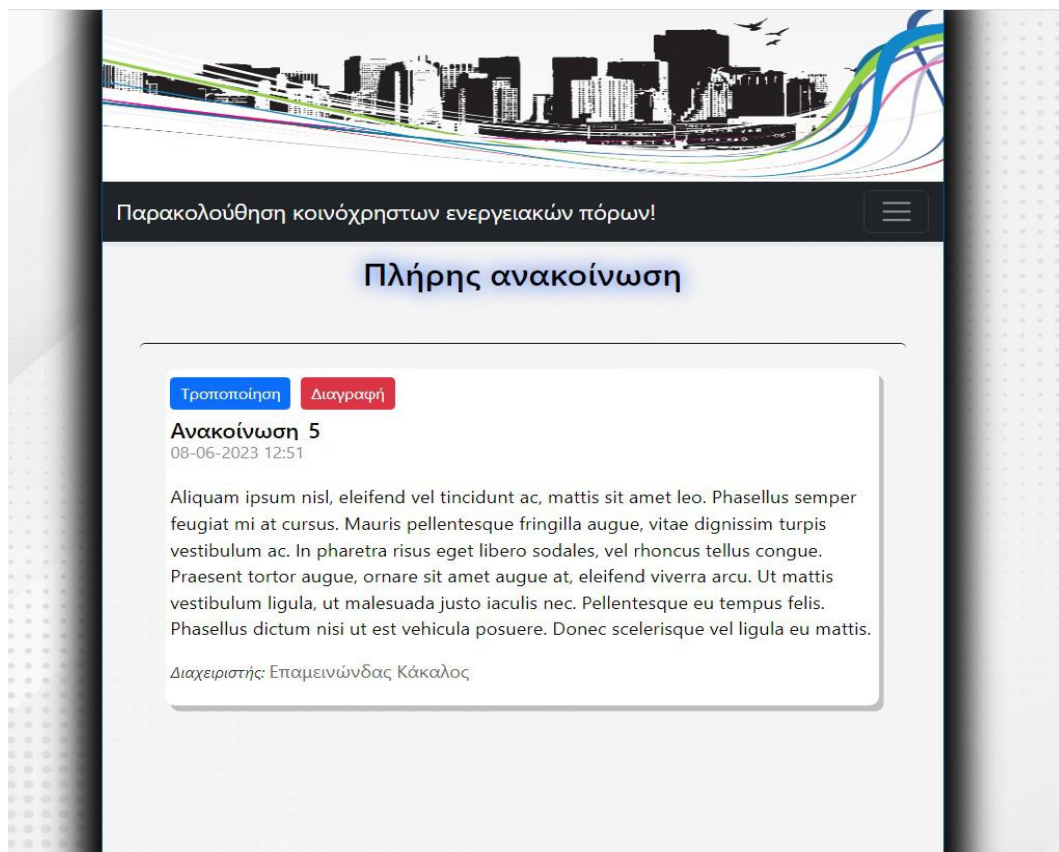
4.46). Αυτό υλοποιείται μέσω της μεθόδου `def full_announcement()` και του HTML αρχείου `full_announcement.html`.



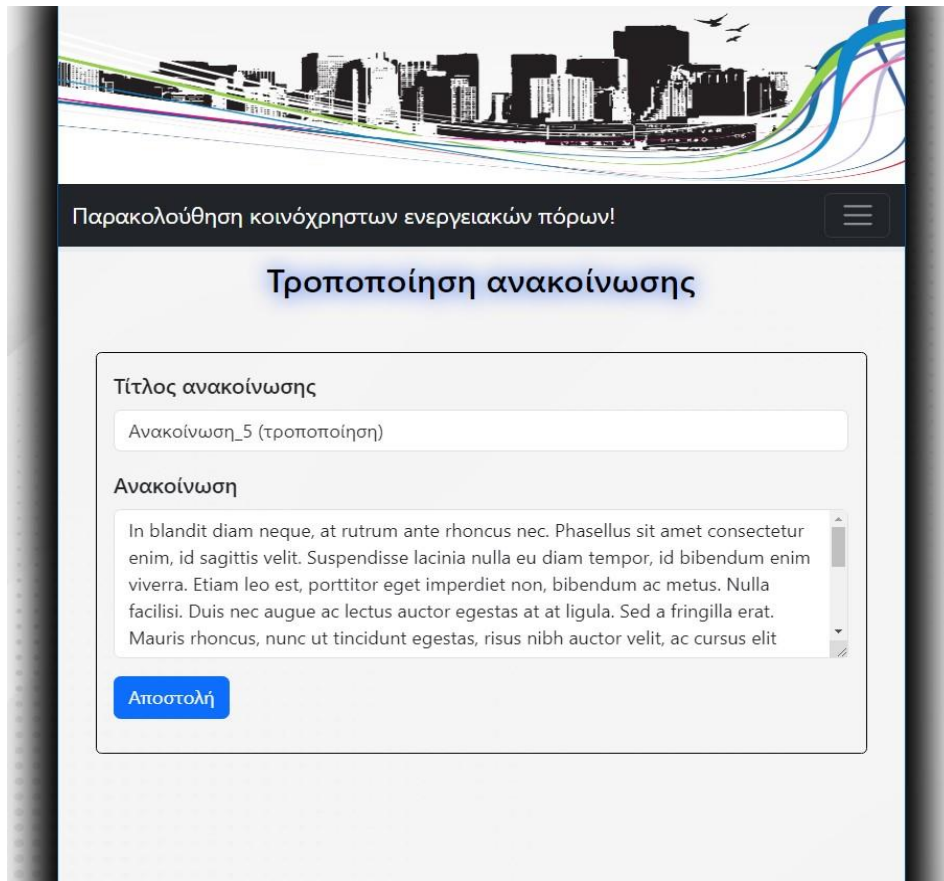
Εικόνα 4.45: Σελίδα Ανακοινώσεων

Αν ο χρήστης είναι και ο δημιουργός της ανακοίνωσης, τότε μπορεί να την τροποποιήσει ή να την διαγράψει (η τροποποίηση και διαγραφή εμφανίζονται μόνο στον αντίστοιχο διαχειριστή) (Εικόνα 4.46). Την τροποποίηση χειρίζεται η μέθοδος `def edit_announcement(announcement_id)`, που σαν όρισμα παίρνει τον α/α της ανακοίνωσης και το αντίστοιχο αρχείο HTML είναι το `new_announcement.html`. Χρησιμοποιείται η σελίδα της νέας ανακοίνωσης, καθώς στο ερώτημα που γίνεται στη βάση, ζητείται να επιστραφούν τα δεδομένα που υπάρχουν, ώστε να επεξεργαστούν και να καταχωρηθούν εκ νέου (Εικόνα 4.47). Έτσι, μετά την ολοκλήρωση της επεξεργασίας και πατώντας «Αποστολή», όντας τροποποιημένα περνάν στη Β.Δ. και γίνεται ανακατεύθυνση στην σελίδα Ανακοινώσεων (Εικόνα 4.48). Πατώντας διαγραφή, εμφανίζεται ένα παράθυρο επιβεβαίωσης (Εικόνα 4.49), πριν την οριστική διαγραφή της ανακοίνωσης (`my_db.session.delete(...)`), και την αφαίρεση της από τη βάση. Αφού διαγραφεί, εμφανίζεται το αντίστοιχο μήνυμα και γίνεται ξανά η ανακατεύθυνση στη σελίδα των Ανακοινώσεων (Εικόνα 4.50). Τα παράθυρα των μηνυμάτων, είναι προϊόν του Bootstrap και η μέθοδος για τη διαγραφή είναι η `def delete_announcement(announcement_id)`.

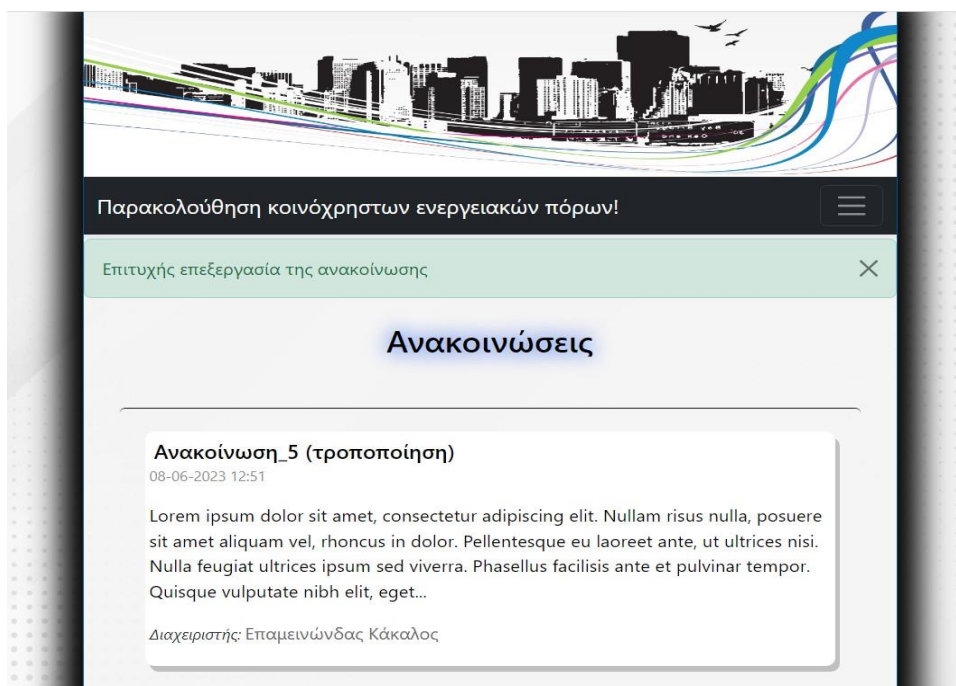
Στη σελίδα των Ανακοινώσεων, πέρα από τον τίτλο και το κείμενο της κάθε ανακοίνωσης, περιέχεται επίσης η ημερομηνία και ώρα, καθώς και ποιος διαχειριστής την δημοσίευσε. Μόνο όταν ένας χρήστης συνδεθεί με δικαιώματα διαχειριστή, μπορεί πατώντας πάνω στο όνομα αυτού που δημοσίευσε μια ανακοίνωση, να δει συγκεντρωτικά όλες τις ανακοινώσεις, του συγκεκριμένου, σε νέο παράθυρο (Εικόνα 4.51). Η υλοποίηση γίνεται μέσω της μεθόδου `def build_admin_announc(build_admin_id)` και του HTML αρχείου `build_admin_annnc.html`. Επίσης, πατώντας πάνω σε κάποια ανακοίνωση, μπορεί να δει το πλήρες κείμενο. Εάν είναι και αυτός που τη δημιούργησε, τότε και από τη συγκεκριμένη σελίδα, μπορεί να την τροποποιήσει ή να την διαγράψει. Στις Εικόνες 4.52 έως 4.54 εμφανίζονται στιγμιότυπα από τη Βάση Δεδομένων πριν και μετά την επεξεργασία και διαγραφή της Ανακοίνωσης\_5.



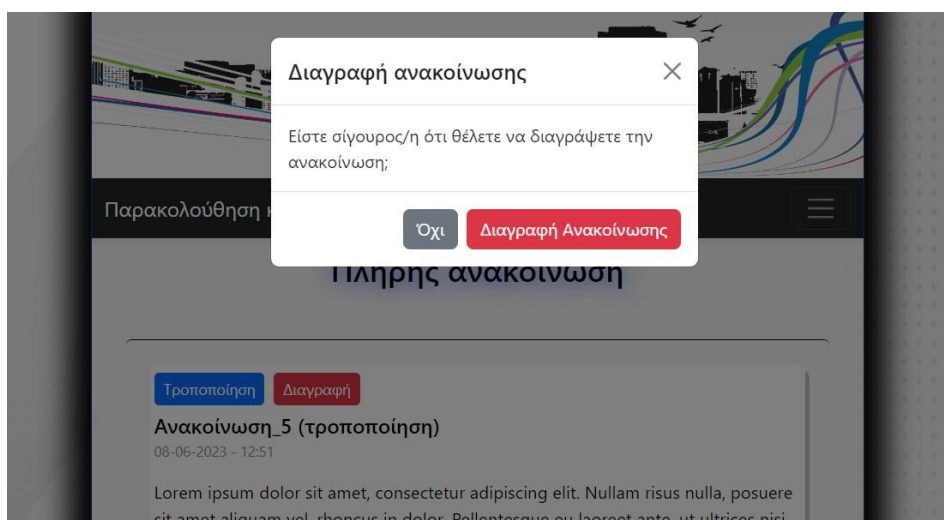
Εικόνα 4.46: Εμφάνιση πλήρους ανακοίνωσης και τροποποίηση ή διαγραφή αυτής



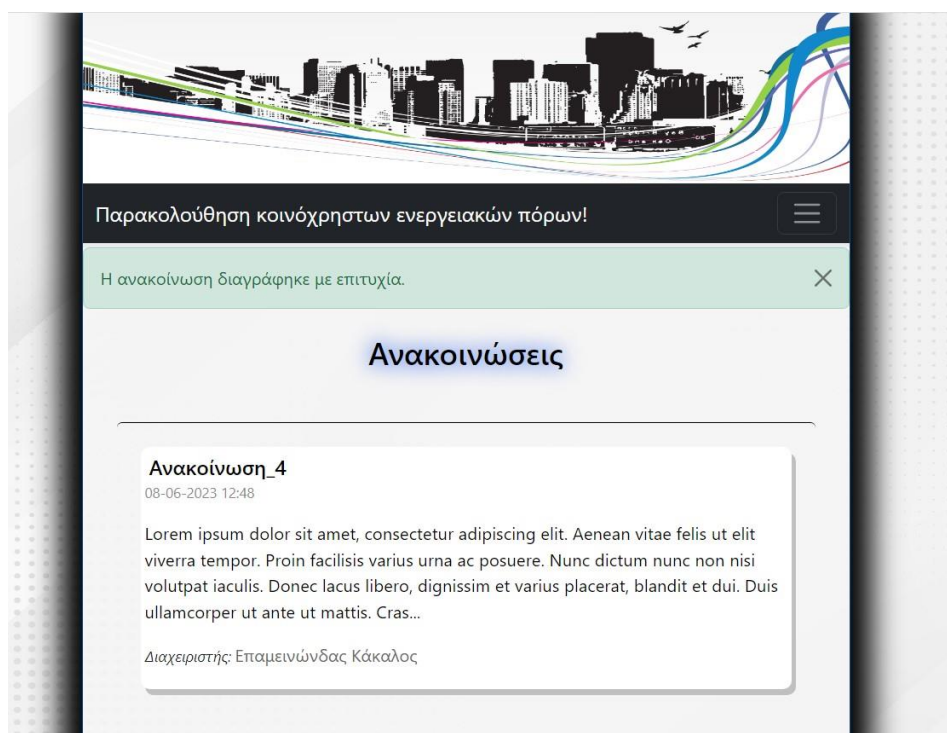
Εικόνα 4.47: Τροποποίηση υπάρχουσας ανακοίνωσης



Εικόνα 4.48: Μήνυμα επιτυχούς επεξεργασίας ανακοίνωσης



Εικόνα 4.49: Μήνυμα επιβεβαίωσης διαγραφής ανακοίνωσης



Εικόνα 4.50: Μήνυμα επιτυχούς διαγραφής της ανακοίνωσης (Ανακοίνωση\_5)



Εικόνα 4.51: Όλες οι ανακοινώσεις ενός διαχειριστή

	id	announc_title	announc_main	date_created	user_id
▶	1	Ανακοίνωση_1	Δημιουργείτε η παρούσα α...	2023-05-16 22:50:22	1
	2	Ανακοίνωση_2	Lorem ipsum dolor sit ame...	2023-06-08 12:20:15	3
	3	Ανακοίνωση_3	Lorem Ipsum is simply du...	2023-06-08 12:37:18	3
	4	Ανακοίνωση_4	Lorem ipsum dolor sit ame...	2023-06-08 12:48:56	1
	5	Ανακοίνωση_5	Lorem ipsum dolor sit ame...	2023-06-08 12:51:36	1
*	NULL	NULL	NULL	NULL	NULL

Εικόνα 4.52: Εμφάνιση των πέντε ανακοινώσεων στη Β.Δ.

	id	announc_title	announc_main	date_created	user_id
▶	1	Ανακοίνωση_1	Δημιουργείτε η παρούσα α...	2023-05-16 22:50:22	1
	2	Ανακοίνωση_2	Lorem ipsum dolor sit ame...	2023-06-08 12:20:15	3
	3	Ανακοίνωση_3	Lorem Ipsum is simply du...	2023-06-08 12:37:18	3
	4	Ανακοίνωση_4	Lorem ipsum dolor sit ame...	2023-06-08 12:48:56	1
	5	Ανακοίνωση_5 (τροποποίηση)	Lorem ipsum dolor sit ame...	2023-06-08 12:51:36	1
*					

Εικόνα 4.53: Εμφάνιση της τροποποιημένης ανακοίνωσης (Ανακοίνωση\_5) στη Β.Δ

	id	announc_title	announc_main	date_created	user_id
▶	1	Ανακοίνωση_1	Δημιουργείτε η παρούσα α...	2023-05-16 22:50:22	1
	2	Ανακοίνωση_2	Lorem ipsum dolor sit ame...	2023-06-08 12:20:15	3
	3	Ανακοίνωση_3	Lorem Ipsum is simply du...	2023-06-08 12:37:18	3
	4	Ανακοίνωση_4	Lorem ipsum dolor sit ame...	2023-06-08 12:48:56	1
*					

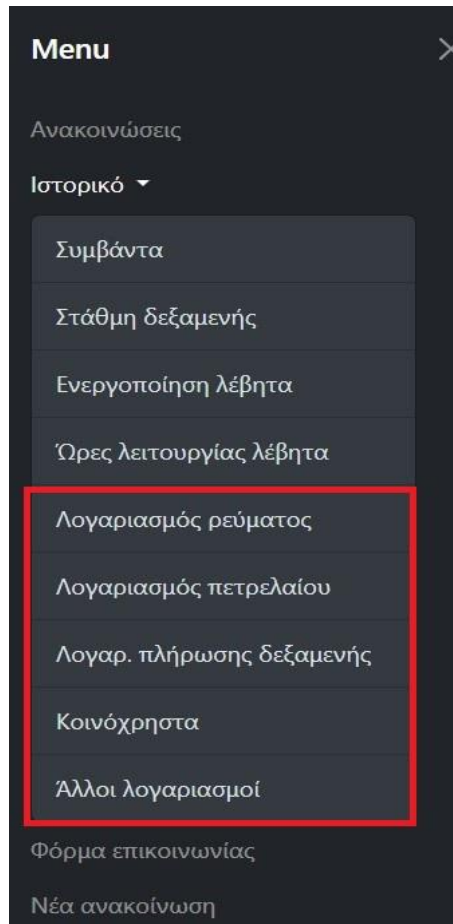
Εικόνα 4.54: Εμφάνιση των ανακοινώσεων στη Β.Δ. μετά τη διαγραφή της ανακοίνωσης (Ανακοίνωση\_5)

#### 4.10 Ανέβασμα αρχείου

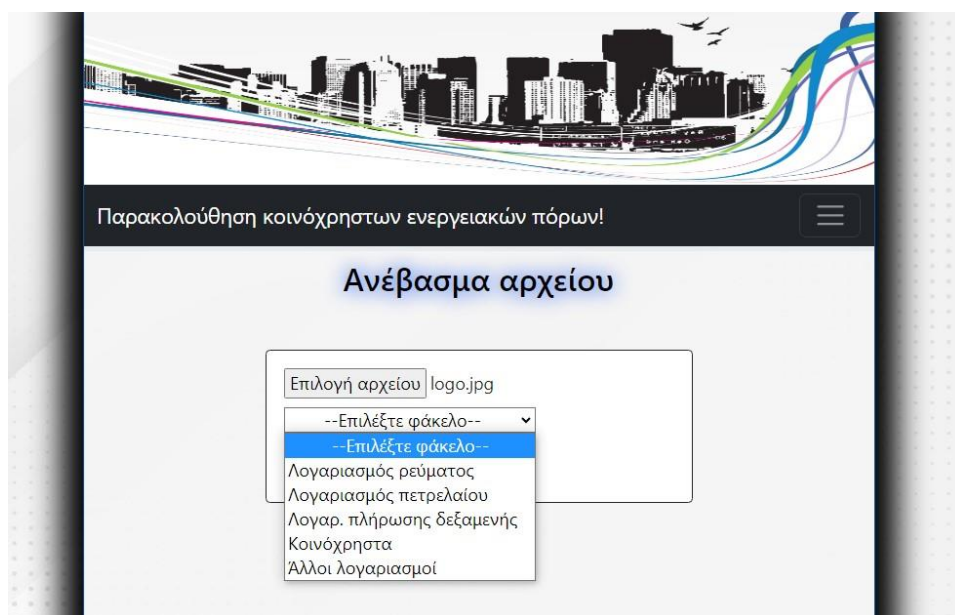
Οι διαχειριστές μπορούν να ανεβάσουν αρχεία, που αποθηκεύονται στη Β.Δ. (πίνακας `upload_files`) και είναι προσβάσιμα σε όλους, μέσω της εφαρμογής από το Ιστορικό. Αυτά μπορεί να είναι κατά βάση, ότι σχετίζεται με λογαριασμούς. Για το λόγο αυτό, έχουν δημιουργηθεί πέντε επιλογές φακέλων (από το μενού, επιλογή της καρτέλας Ιστορικού) που εμφανίζονται στο κόκκινο πλαίσιο της Εικόνας 4.55. Επιλέχθηκαν αυτές, ώστε να καλύπτουν ένα ευρύ φάσμα λογαριασμών, που είθισται να υπάρχουν σε μια πολυκατοικία. Έτσι ανάλογα την επιλογή του φακέλου, όταν το αρχείο κοινοποιηθεί, θα μπορεί να εμφανίζεται στον αντίστοιχο φάκελο.

Από το μενού επιλέγοντας «Ανέβασμα αρχείου» φορτώνει η αντίστοιχη σελίδα και ζητείται να γίνει η επιλογή αρχείου και η επιλογή φακέλου (Εικόνα 4.56). Μόνο επιλέγοντας και τα δυο και πατώντας «Αποστολή», το πρόγραμμα ανακατευθύνεται στη μέθοδο των ανακοινώσεων, εμφανίζοντας παράλληλα το αντίστοιχο μήνυμα (Εικόνα 4.57). Τα επιτρεπόμενα αρχεία μπορούν να είναι «pdf, png, jpg, jpeg» με μέγιστο μέγεθος τα 4MB, ενώ τα δεδομένα που αποθηκεύονται στη βάση είναι τύπου BLOB. Αν το μέγεθος ξεπεραστεί, εμφανίζεται το αντίστοιχο μήνυμα και το αρχείο δεν αποθηκεύεται στη Β.Δ (Εικόνα 4.58). Από πλευράς HTML κώδικα, δημιουργήθηκε μια φόρμα που θα στείλει τα δεδομένα με POST και προστέθηκε το «enctype="multipart/form-data"». Χωρίς αυτό δεν μπορούν να σταλούν αρχεία προς το back-end. Επίσης, η επιλογή του φακέλου γίνεται από τη λίστα επιλογών της HTML και η κάθε επιλογή αντιστοιχεί σε μια τιμή. Στην Εικόνα 4.59 προβάλλονται τα δεδομένα μέσα από τη Β.Δ. Όλα αυτά δρομολογούνται από τη μέθοδο `def upload_files()` και το HTML αρχείο `upload_files.html` [48-49].

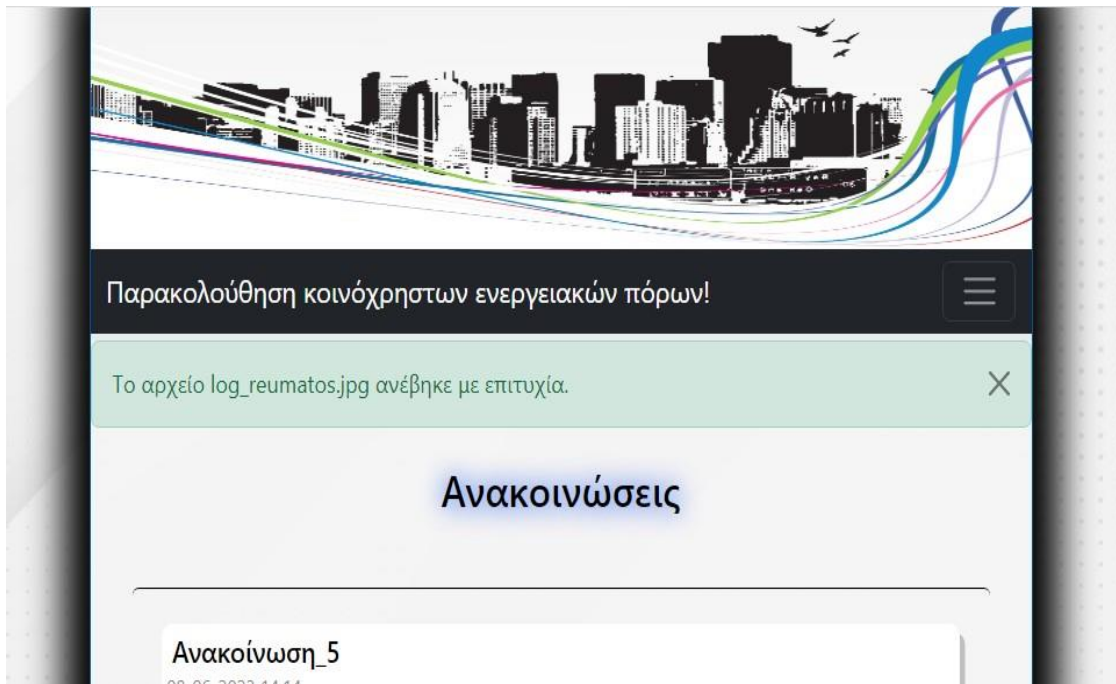




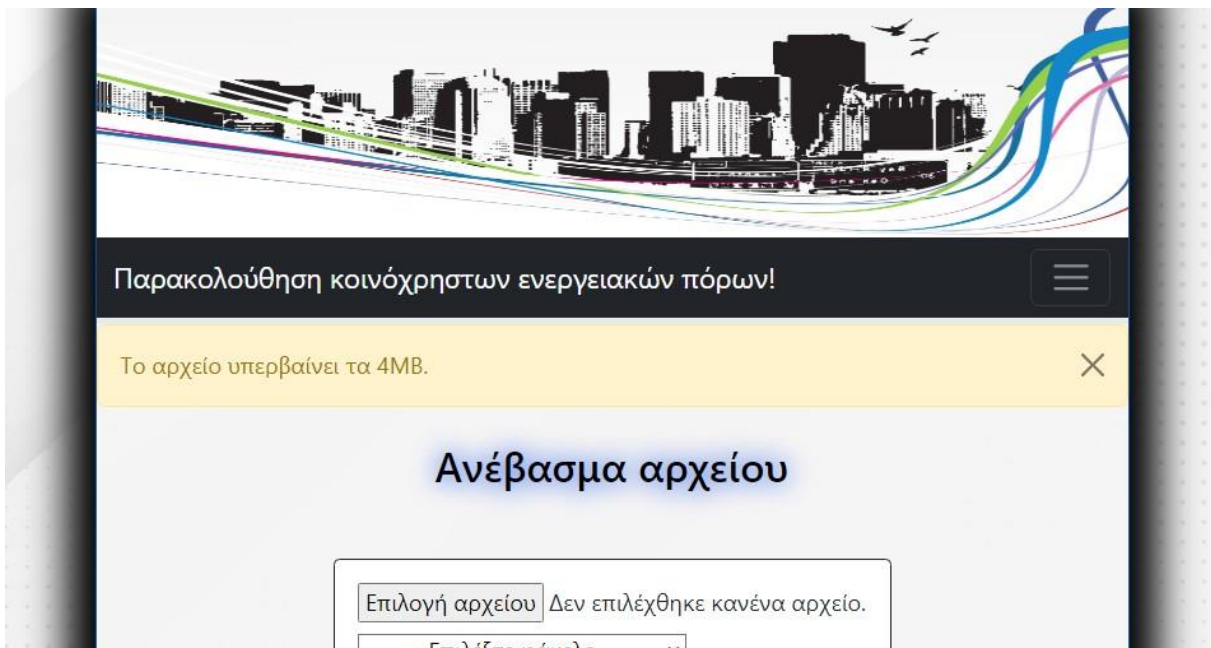
Εικόνα 4.55: Επιλογές για το ανέβασμα αρχείου



Εικόνα 4.56: Επιλογή αρχείου και φακέλου



Εικόνα 4.57: Επιτυχής μεταφόρτωση του αρχείου



Εικόνα 4.58: Μέγεθος αρχείου μεγαλύτερο από 4MB

	id	filename	category	filedata	date_created
▶	1	log_reumatosis.jpg	deh	BLOB	2023-06-08 19:48:34
	2	log_petrelaiou.jpg	petrol	BLOB	2023-06-08 19:52:01
	3	log_plhr_deksmamenhs.jpg	fuel_tank	BLOB	2023-06-08 19:55:14
	4	log_koinoxrhstwn.jpg	koinox	BLOB	2023-06-08 19:57:58
	5	log_nerou.pdf	alloi	BLOB	2023-06-08 20:02:41
*					

Εικόνα 4.59: Εμφάνιση αρχείων στη Β.Δ.

#### 4.11 Στοιχεία ενοίκων και οι συνολικές τους καταγεγραμμένες ώρες

Όσον αφορά τους διαχειριστές και τα επιπλέον δικαιώματα που έχουν στη σελίδα, πέρα από τη δημιουργία ανακοινώσεων, μπορούν να δουν και τα στοιχεία των ενοίκων (Εικόνα 4.60). Τα ίδια στοιχεία μέσα από τη Β.Δ. παρατίθενται στην Εικόνα 4.61. Επίσης, μπορούν να έχουν πρόσβαση και στην παρακολούθηση των συνολικών ωρών λειτουργίας του θερμοστάτη, κάθε διαμερίσματος (Εικόνα 4.62). Αυτό μπορεί να είναι για μια ημέρα ή για διάστημα ημερών. Η μέθοδος για την εμφάνιση των στοιχείων των χρηστών, είναι η *def people()* και το HTML αρχείο είναι το *people.html*. Να σημειωθεί πως η συγκεκριμένη σελίδα, σε πραγματικές συνθήκες θα εμφάνιζε σίγουρα περισσότερους των τριών, χρήστες. Για το λόγο αυτό, γίνεται σελιδοποίηση στους εικοσιπέντε ανά σελίδα. Ολοκληρώνοντας, όσον αφορά τις συνολικές ώρες, χρησιμοποιείται η μέθοδος *def total\_h\_hours()* και το HTML αρχείο *total\_h\_hours.html*. Η λογική του προγραμματισμού της μεθόδου, είναι ακριβώς ίδια με αυτή της Αρχικής σελίδας.

### Στοιχεία ενοίκων

A/A	Επώνυμο	Όνομα	Email	Όροφος	Διαμέρισμα
1	Κάκαλος	Επαμεινώνδας	cmmf211@hotmail.com	1	1
2	Κεραντάρη	Ελισάβετ	mpoumpou_thesis@hotmail.com	1	2
3	Βλάχος	Γεώργιος	cmmf211@outlook.com	2	3

1

Εικόνα 4.60: Στοιχεία ενοίκων μέσα από την εφαρμογή

	id	username	email	password	first_name	last_name	floor	house_nu	date_created	admin_key
▶	1	smhnarxos_kakalos	cmmf211@hotmail.com	\$2b\$12\$w.xVozg8pXifh0K...	Επαμεινώνδας	Κάκαλος	1	1	2023-05-16 22:4...	12123
	2	mpoumpou	mpoumpou_thesis@hot...	\$2b\$12\$g/ik91gXweo0O3...	Ελισάβετ	Κεραντάρ...	1	2	2023-06-05 21:5...	0
	3	o_palios	cmmf211@outlook.com	\$2b\$12\$CkgShbH7.om1cF...	Γεώργιος	Βλάχος	2	3	2023-06-08 12:0...	12123
*										

Εικόνα 4.61: Εμφάνιση στοιχείων των χρηστών στη Β.Δ.

**Συνολικές ώρες λειτουργίας ενοίκων**

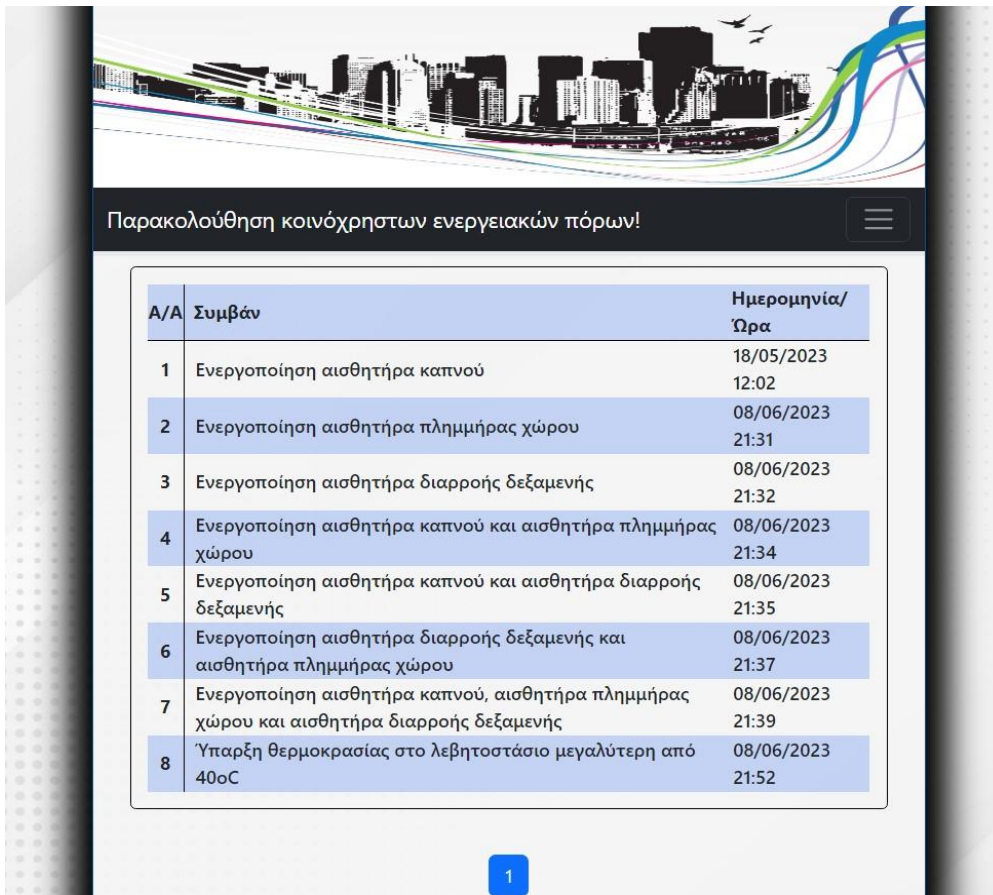
Διάστημα από: 01/03/2023  Έως: 07/03/2023  Αποστολή

Διαμέρισμα	Συνολικές ώρες
Διαμέρισμα 1_1	56.78
Διαμέρισμα 1_2	51.08
Διαμέρισμα 2_3	46.17

Εικόνα 4.62: Συνολικές ώρες λειτουργίας των ενοίκων για το διάστημα ημερών 1/3/2023 ~ 7/3/2023

#### 4.12 Ιστορικό συμβάντων

Επιλέγοντας από το μενού, Ιστορικό κι έπειτα Συμβάντα εμφανίζονται όλα τα συμβάντα που έχουν καταγραφεί και αφορούν την ασφάλεια του χώρου (Εικόνα 4.63). Αυτά βρίσκονται καταχωρημένα μέσα στην Β.Δ. (Εικόνα 4.64). Η μέθοδος από την οποία γίνεται το ερώτημα στη Β.Δ και εμφανίζονται τα αποτελέσματα, είναι η *def history(value\_id)*. Το όρισμα της, είναι η τιμή που αντιστοιχεί στην επιλογή και προέρχεται μέσα από τις επιλογές του Ιστορικού (μενού της εφαρμογής). Η ίδια τιμή χρησιμοποιείται και στο Jinja template του HTML αρχείου *people.html*, ώστε να τα εμφανίσει.



A/A	Συμβάν	Ημερομηνία/ Ωρα
1	Ενεργοποίηση αισθητήρα καπνού	18/05/2023 12:02
2	Ενεργοποίηση αισθητήρα πλημμύρας χώρου	08/06/2023 21:31
3	Ενεργοποίηση αισθητήρα διαρροής δεξαμενής	08/06/2023 21:32
4	Ενεργοποίηση αισθητήρα καπνού και αισθητήρα πλημμύρας χώρου	08/06/2023 21:34
5	Ενεργοποίηση αισθητήρα καπνού και αισθητήρα διαρροής δεξαμενής	08/06/2023 21:35
6	Ενεργοποίηση αισθητήρα διαρροής δεξαμενής και αισθητήρα πλημμύρας χώρου	08/06/2023 21:37
7	Ενεργοποίηση αισθητήρα καπνού, αισθητήρα πλημμύρας χώρου και αισθητήρα διαρροής δεξαμενής	08/06/2023 21:39
8	Υπαρξη θερμοκρασίας στο λεβητοστάσιο μεγαλύτερη από 40οC	08/06/2023 21:52

Εικόνα 4.63: Εμφάνιση καταγεγραμμένων συμβάντων στην ιστοσελίδα

id	eventb	date_created
1	Ενεργοποίηση αισθητήρα καπνού	2023-05-18 12:02:19
2	Ενεργοποίηση αισθητήρα πλημμύρας χώρου	2023-06-08 21:31:30
3	Ενεργοποίηση αισθητήρα διαρροής δεξαμενής	2023-06-08 21:32:26
4	Ενεργοποίηση αισθητήρα καπνού και αισθητήρα πλημμύρας χώρου	2023-06-08 21:34:10
5	Ενεργοποίηση αισθητήρα καπνού και αισθητήρα διαρροής δεξαμενής	2023-06-08 21:35:19
6	Ενεργοποίηση αισθητήρα διαρροής δεξαμενής και αισθητήρα πλημμύρας χώρου	2023-06-08 21:37:01
7	Ενεργοποίηση αισθητήρα καπνού, αισθητήρα πλημμύρας χώρου και αισθητήρα διαρροής δεξαμενής	2023-06-08 21:39:03
8	Ύπαρξη θερμοκρασίας στο λεβητοστάσιο μεγαλύτερη από 40οC	2023-06-08 21:52:00

Εικόνα 4.64: Καταγεγραμμένα συμβάντα στη βάση δεδομένων

### 4.13 Ιστορικό Στάθμης δεξαμενής

Επιλέγοντας από το μενού, Ιστορικό κι έπειτα Στάθμη δεξαμενής, εμφανίζονται τα δεδομένα που αφορούν τη στάθμη του πετρελαίου στη δεξαμενή. Είναι αυτά που καταγράφονται στον πίνακα *Diaperismata*, μαζί με τις ώρες του κάθε διαμερίσματος. Υποστηρίζει την εμφάνιση διαγράμματος είτε για μια ημέρα, είτε για διάστημα ημερών (Εικόνα 4.65) ενώ και στις δυο περιπτώσεις αναφέρεται η συνολική κατανάλωση λίτρων, στο κάτω μέρος της οθόνης. Όσον αφορά την πλήρωση της δεξαμενής, τα λίτρα πριν και μετά, εμφανίζονται στην αντίστοιχη ημερήσια επιλογή (Εικόνα 4.66). Η μέθοδος που ασχολείται με την εμφάνιση αυτών των δεδομένων είναι η *def index\_petrol()*, ενώ το HTML αρχείο είναι το *index\_petrol.html*. Η αρχή λειτουργίας της μεθόδου, είναι και πάλι η ίδια, με αυτή της Αρχικής σελίδας.



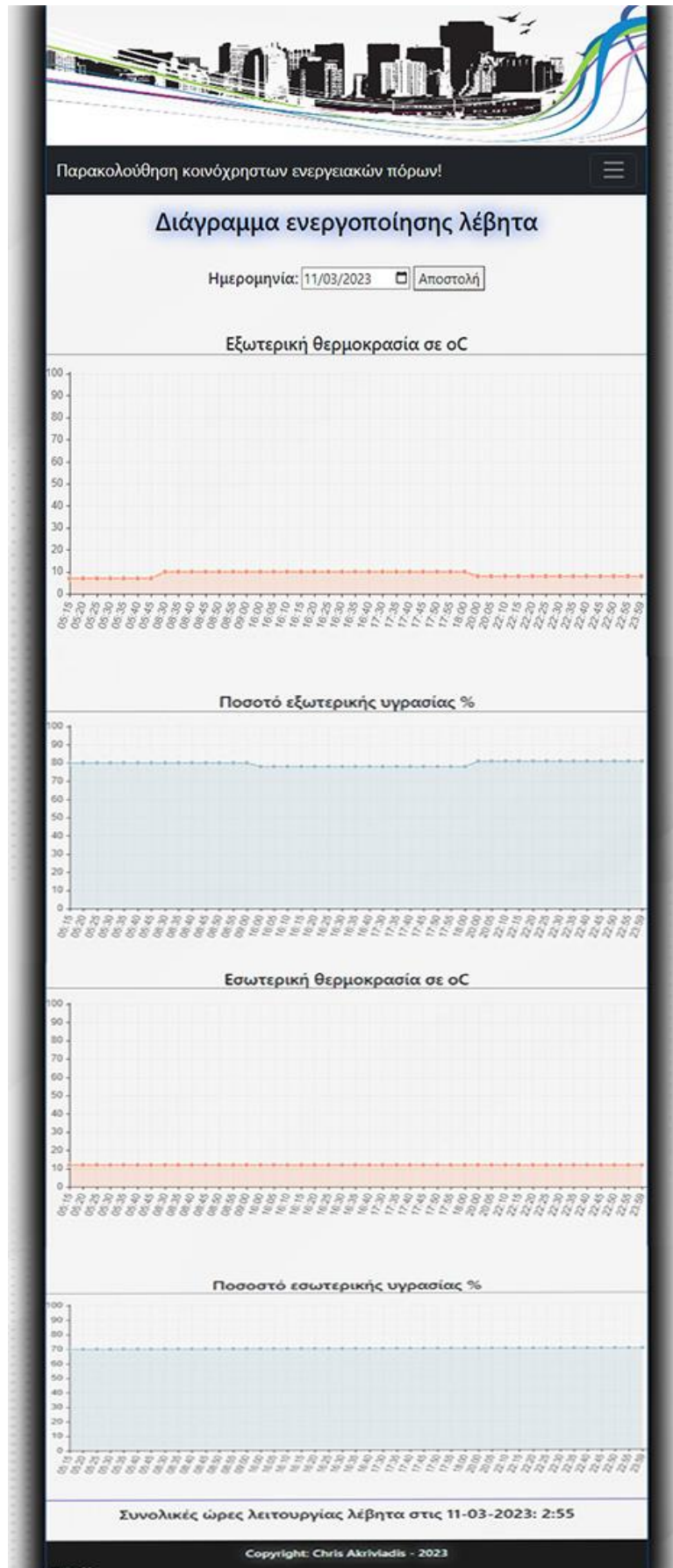
Εικόνα 4.65: Διάγραμμα κατανάλωσης λίτρων πετρελαίου για το διάστημα 1/3/2023 έως 7/3/2023



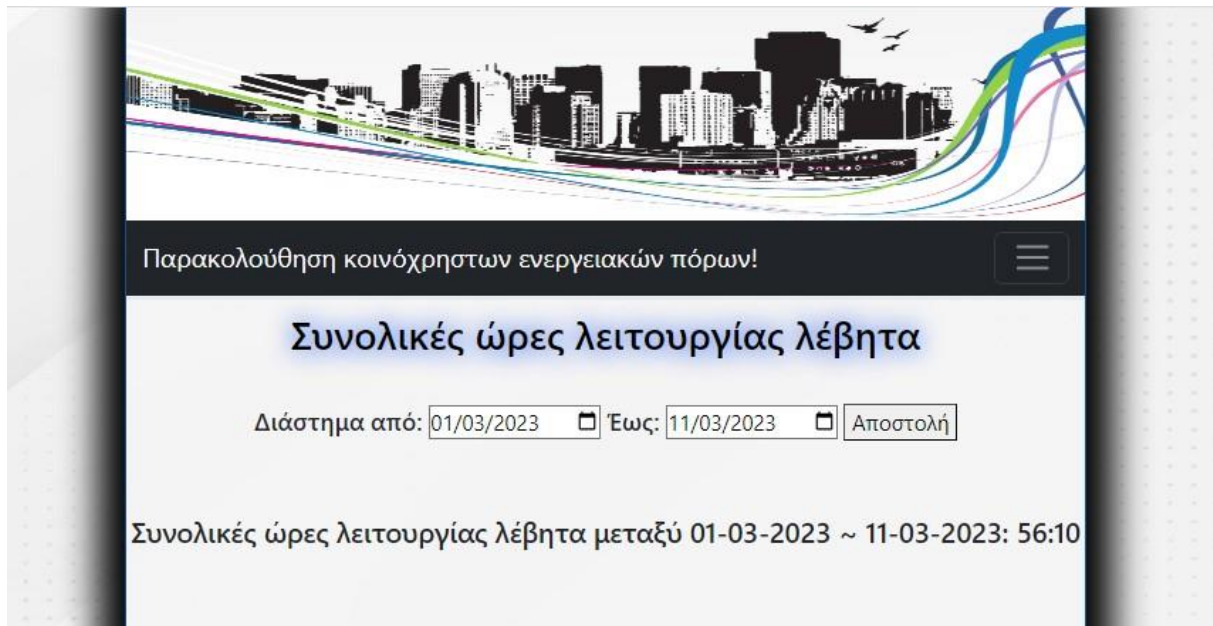
Εικόνα 4.66: Διάγραμμα ημερήσιας κατανάλωσης λίτρων πετρελαίου και λίτρα πλήρωσης της δεξαμενής (4/3/2023)

#### 4.14 Ιστορικό Ενεργοποίησης λέβητα και Ώρες λειτουργίας του λέβητα

Μέσα από το μενού, Ιστορικό κι επιλέγοντας Ενεργοποίηση λέβητα, εμφανίζονται τα διαγράμματα με τις καταγεγραμμένες εξωτερικές και εσωτερικές θερμοκρασίες-υγρασίες (Εικόνα 4.67). Ο προγραμματισμός της μεθόδου `def temp_hum()`, βασίζεται σε όσα αναλύθηκαν γι' αυτή της Αρχικής σελίδας. Στο HTML αρχείο `temp_hum.html` υπάρχουν τα αντίστοιχα διαγράμματα, που υλοποιήθηκαν μέσω της γλώσσας προγραμματισμού JavaScript και του Chart.js [46]. Σε αντίθεση με προηγούμενες περιπτώσεις, εδώ υποστηρίζεται μονό η ημερήσια προβολή, γιατί τα δεδομένα αφορούν τις ώρες ενεργοποιήσεις του λέβητα. Αυτά ποικίλουν από μέρα σε μέρα, κι έτσι η εμφάνιση ενός διστήματος ημερών, δεν παρουσιάζει κάποιο ενδιαφέρον. Στο διάγραμμα εμφανίζονται και τα λεπτά των καταγεγραμμένων τιμών, ενώ στο κάτω μέρος της σελίδας υπάρχουν οι συνολικές ώρες λειτουργίας του λέβητα, για τη συγκεκριμένη μέρα. Υπογραμμίζεται ότι όσα αναφέρθηκαν για τις συνολικές ώρες, αφορούν την ημερήσια καταγραφή και μόνο. Αυτό μιας και από το μενού, Ιστορικό κι επιλέγοντας Ώρες λειτουργίας λέβητα, υπάρχει η δυνατότητα εμφάνισης μόνο του συνόλου των ωρών, για συγκεκριμένο χρονικό διάστημα (ή και ανά ημέρα) (Εικόνα 4.68). Αυτό γίνεται για να υπάρχει μια εικόνα σχετικά με τη χρήση του λέβητα και να διευθετείται ανάλογα, η όποια συντήρηση του απαιτείται. Η μέθοδος και το HTML αρχείο είναι τα ίδια με αυτό της Ενεργοποίησης λέβητα, μιας και από τον ίδιο πίνακα της Β.Δ., εκμαιεύονται τα νούμερα που απαιτούνται (πίνακας `kron`).



Εικόνα 4.67: Διάγραμμα ενεργοποίησης λέβητα και διαγράμματα θερμοκρασίας-υγρασίας



Εικόνα 4.68: Συνολικές ώρες λειτουργίας του λέβητα για το διάστημα 1/3/2023 έως 11/3/2023

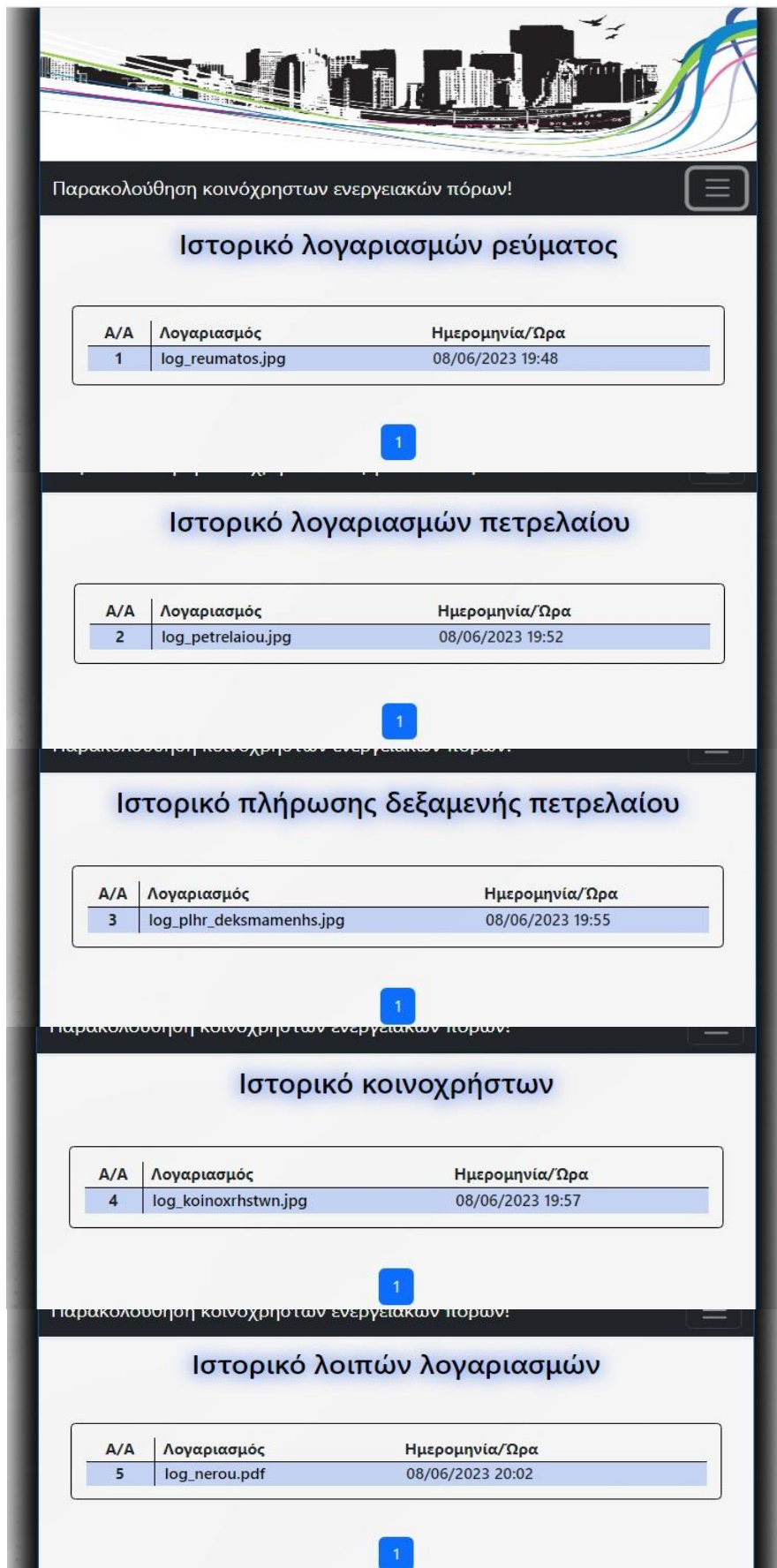
#### 4.15 Ιστορικό λογαριασμών (αποθηκευμένα αρχεία)

Μέσα από το μενού, Ιστορικό κι επιλέγοντας τον αντίστοιχο φάκελο εμφανίζονται τα αρχεία (λογαριασμοί) που αντιστοιχούν σε αυτόν (Εικόνα 4.69). Πατώντας πάνω σε κάθε αρχείο, γίνεται η αποθήκευση του τοπικά. Η μέθοδος υλοποίησης είναι η `def history(value_id)` και παίρνει σαν όρισμα την αντίστοιχη τιμή από το μενού. Το HTML αρχείο είναι το `people.html` και μέσω των Jinja templates και της τιμής `value_id` γίνεται ο διαχωρισμός, για το τι θα εμφανίσει στις αντίστοιχες στήλες, ως τίτλος. Το ερώτημα προς τη Β.Δ γίνεται με κριτήριο το «category». Επισημαίνεται ότι στον αντίστοιχο πίνακα (`upload_files`) υπάρχει αυτή η στήλη, ώστε τα αρχεία που αποθηκεύονται εκεί, να μπορούν έπειτα να γίνονται διαλογή και να επιστρέφουν πίσω όχι όλα, αλλά μόνο αυτά για τα οποία εκδήλωσε ενδιαφέρον, ο χρήστης.

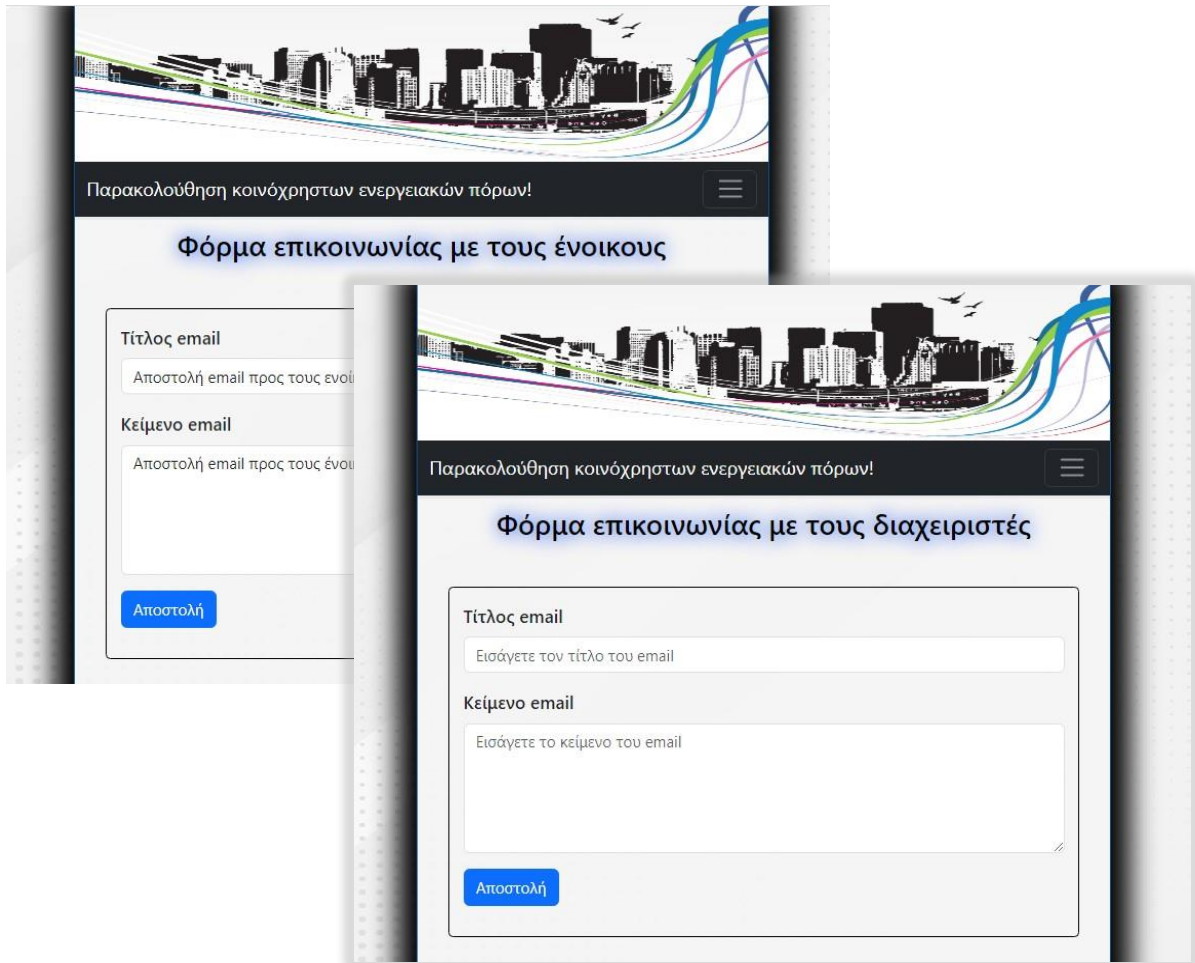
#### 4.16 Φόρμα επικοινωνίας

Μέσω της εφαρμογής υποστηρίζεται και η αποστολή email. Αυτή μπορεί να γίνει από τους διαχειριστές προς όλους τους ενοίκους της πολυκατοικίας ή από ένα συγκεκριμένο χρήστη, προς τους διαχειριστές. Η επιλογή γίνεται από το μενού, Φόρμα επικοινωνίας και η μέθοδος υποστήριξης είναι `def contact_form()`. Το HTML αρχείο είναι το `contact_form.html`, ενώ ο τίτλος της φόρμας προστίθεται δυναμικά, ανάλογα το ποιος χρήστης είναι συνδεδεμένος (Εικόνα 4.70). Με την επιτυχή αποστολή του email, η εφαρμογή ανακατευθύνεται στην Αρχική σελίδα, ενώ εμφανίζεται και το αντίστοιχο μήνυμα (Εικόνα 4.71). Στην Εικόνα 4.72 παρουσιάζονται τα απεσταλμένα email μεταξύ των ενοίκων και των διαχειριστών.

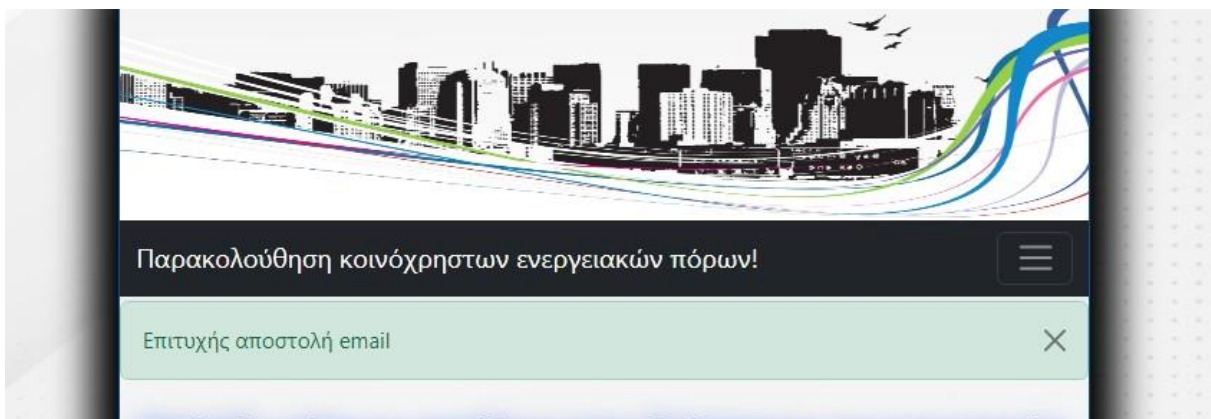




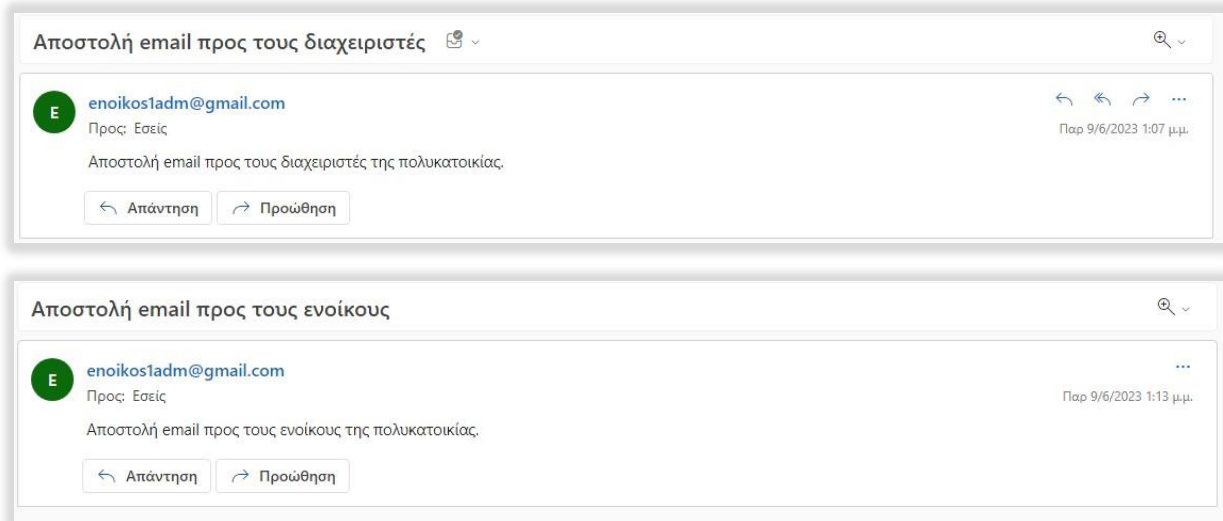
Εικόνα 4.69: Προβολή όλων των φακέλων με τα αρχεία και αποθήκευση αυτών



Εικόνα 4.70: Φόρμα επικοινωνίας μεταξύ ενοίκων και διαχειριστών



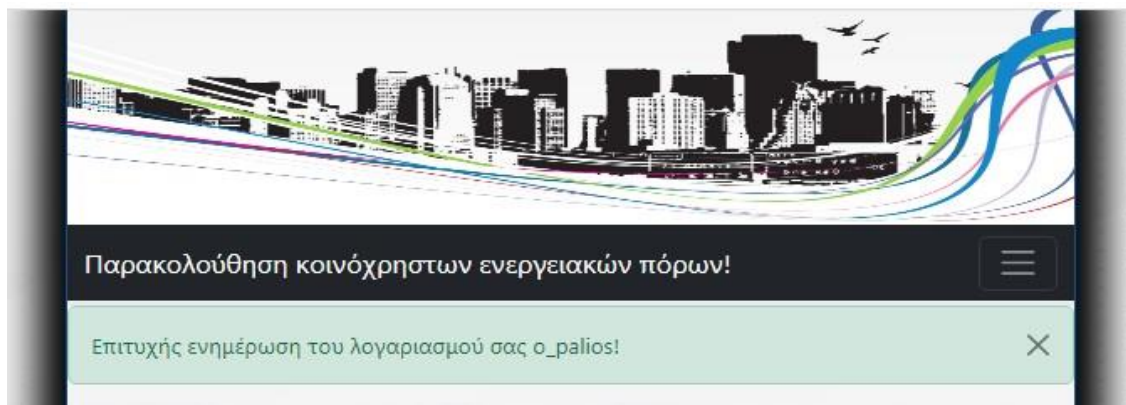
Εικόνα 4.71: Μήνυμα επιτυχούς αποστολής του email



Εικόνα 4.72: Εισερχόμενο email στους ενοίκους και τους διαχειριστές, αντίστοιχα

#### 4.17 Ενημέρωση στοιχείων λογαριασμού

Μέσα από το μενού και πατώντας πάνω στο username, επιλέγοντας Λογαριασμός, είναι δυνατή η ενημέρωση όλων των στοιχείων του λογαριασμού χρήστη (Εικόνα 4.74). Η μέθοδος γι' αυτό είναι η *def update\_account()* και το HTML αρχείο είναι το *update\_account.html*. Μέσα από τη μέθοδο, και την κλάση UpdateAccountForm, φορτώνονται τα δεδομένα ώστε να είναι ορατά και διαθέσιμα για επεξεργασία. Αυτό επιτυγχάνεται μέσω του «current\_user», από τη στιγμή που έγινε η σύνδεση. Παραδείγματος χάρη, το «email=current\_user.email» φορτώνει στη μεταβλητή email, το email του χρήστη. Αν κάποιος χρήστης θέλει μετά την εγγραφή του, να σταματήσει να έχει δικαιώματα διαχειριστή, αρκεί να κάνει ενημέρωση και στο πεδίο Admin\_Key, να προσθέσει τον αριθμό “0”. Αντίθετα, αν κάποιος θέλει να αποκτήσει δικαιώματα διαχειριστή, θα πρέπει να προσθέσει το “12123”. Μετά την επεξεργασία και επικύρωση των δεδομένων, γίνεται η ενημέρωση της Β.Δ. και η ανακατεύθυνση στην Αρχική σελίδα, εμφανίζοντας παράλληλα το αντίστοιχο flash message (Εικόνα 4.73).



Εικόνα 4.73: Επιτυχής ενημέρωση στοιχείων λογαριασμού

The image shows a web application interface. At the top, there is a navigation bar with a cityscape background and the text "Παρακολούθηση κοινόχρηστων ενεργειακών πόρων!". Below this is a main heading "Ενημέρωση στοιχείων λογαριασμού". The form contains several input fields: "Email" (cmmf211@outlook.com), "Όνομα χρήστη" (o\_paliος), "Νέος Κωδικός" (new password), "Όνομα" (Γεώργιος), "Επώνυμο" (Βλάχος), "Όροφος" (2), "Αριθμός διαμερίσματος" (3), and "Admin\_Key" (admin\_key). A blue "Αποστολή" button is at the bottom of the form. A dark "Menu" dropdown is open, showing "Welcome o\_paliος", "Λογαριασμός", and "Αρχική σελίδα".

Εικόνα 4.74: Ενημέρωση στοιχείων λογαριασμού

#### 4.18 Επίλογος

Στο κεφάλαιο αυτό, παρουσιάστηκε με κάθε λεπτομέρεια η αρχή λειτουργίας, του λογισμικού (software) του συστήματος. Παρουσιάστηκαν όλα τα εργαλεία που μετείχαν με σκοπό την υλοποίηση του front-end και back-end κομματιού, καθώς επίσης και η βάση δεδομένων. Παράλληλα, αναφέρθηκαν μια προς μια όλες οι εμπλεκόμενες μέθοδοι, με τις αντίστοιχες σελίδες για την προβολή των δεδομένων. Μέσα από τα παραδείγματα και τις πολλές φωτογραφίες, έγινε η προσπάθεια ώστε να χυθεί άπλετο φως και να αναδειχθεί κάθε πτυχή της προσπάθειας που καταβλήθηκε. Συνοψίζοντας, όσον αφορά το σύνολο του κώδικα που χρησιμοποιήθηκε, αυτό παρατίθεται αυτούσιο στα Παραρτήματα Γ έως Θ. Αναλυτικά, στο Παράρτημα Γ υπάρχουν τα Scripts της Β.Δ., στο Παράρτημα Δ οι κλάσεις για την επικοινωνία με τη Β.Δ., ενώ στο Παράρτημα Ε, οι κλάσεις και οι επικυρώσεις που χρησιμοποιήθηκαν για τα πεδία (φόρμες) της εγγραφής, επαναφοράς κωδικού, εισόδου και ενημέρωσης του λογαριασμού χρήστη. Επιπρόσθετα, αυτές για τη δημιουργία νέας ανακοίνωσης και της επικοινωνίας ενοίκων-διαχειριστών. Στο Παράρτημα ΣΤ παρατίθενται οι αρχικοποιήσεις που έπρεπε να γίνουν και στο Παράρτημα Ζ, εμφανίζονται όλες οι μέθοδοι (route ή rest endpoint). Τέλος, τα Παραρτήματα Η και Θ περιέχουν τα HTML αρχεία και το CSS αρχείο, αντίστοιχα.

## Κεφάλαιο 5ο: Συμπεράσματα και προτάσεις βελτίωσης

### 5.1 Συμπεράσματα

Κατά τη διάρκεια των δοκιμών του συστήματος, παρατηρήθηκε η σωστή και γρήγορη απόκριση στη λήψη και αποστολή των δεδομένων στο server, από την πλευρά της συσκευής. Ο server λειτουργούσε σταθερά και οι ταχύτητες επεξεργασίας και αποστολής των δεδομένων στη βάση, ήταν ικανοποιητικές. Η ίδια ικανοποιητική ταχύτητα, διαπιστώθηκε και όταν τα δεδομένα αντλούνταν από τη Β.Δ., με σκοπό την προβολή τους. Όσον αφορά την ιστοσελίδα, μετά από πιλοτική εφαρμογή και χρήση αυτής από άτομα κυρίως μεγαλύτερης ηλικίας, επισημάνθηκε η απλότητα και η ευχρηστία της. Εν κατακλείδι, ο στόχος που είχε τεθεί, για το πως η τεχνολογία μπορεί να συμβάλει στη ασφάλεια του χώρου, την εξοικονόμηση χρημάτων καθώς και τη διαφάνεια και άμεση ενημέρωση των ενοίκων μιας πολυκατοικίας, επετεύχθη. Δημιουργήθηκε μια διαδικτυακή εφαρμογή, πλήρως λειτουργική και σύναμμα λυτή, που μπορεί να χρησιμοποιηθεί εύκολα από όλες τις ηλιακές ομάδες.

### 5.2 Προτάσεις βελτίωσης

Όπως λέει και ο θυμόσοφος λαός, ο εχθρός του καλού είναι το καλύτερο. Το ίδιο ισχύει και στην συνολική εργασία που παρουσιάστηκε στα προηγούμενα κεφάλαια. Υπάρχουν παρεμβάσεις που μπορούν να γίνουν σε πολλαπλά επίπεδα, ώστε να βελτιωθεί ακόμα περισσότερο το όλο εγχείρημα.

Όσον αφορά την συσκευή, πρέπει να γίνει μια μελέτη ώστε να μπορεί να υποστηρίξει και το Φυσικό Αέριο, ως μέσο θέρμανσης. Αυτό, μιας και πολλές πολυκατοικίες χρησιμοποιούν το συγκεκριμένο καύσιμο, αντί του πετρελαίου. Επίσης πρέπει να γίνει μελέτη για το πώς η συσκευή ή το σύστημα, θα μπορεί να επικοινωνεί με τα συστήματα της αγοράς (π.χ. τους ωρομετρητές), ώστε να απομαστεύει τα συγκεκριμένα δεδομένα. Άλλη μια βελτίωση, σε επίπεδο σύνδεσης στο δίκτυο, είναι να υποστηρίζεται και η ενσύρματη επιλογή, πέρα της ασύρματης. Όσον αφορά την οθόνη, αυτή μπορεί να αντικατασταθεί με κάποια μεγαλύτερων ιντσών και να αλλάξει το γραφικό της περιβάλλον, ώστε να γίνει πιο ελκυστικό και οπτικά πιο σύγχρονο. Επίσης, μπορεί να γίνει αναβάθμιση κάποιων κυκλωμάτων, όπως για παράδειγμα τη θέση του buzzer να πάρει ένα κύκλωμα, που θα ενεργοποιεί μια σειρήνα. Επιπρόσθετα, να αποφασιστεί ποια αντλία θα χρησιμοποιηθεί, αν σε περίπτωση υπάρξει διαρροή της δεξαμενής και να υλοποιηθεί το αντίστοιχο κύκλωμα υποστήριξης. Όσον αφορά τα αισθητήρια, θα πρέπει σίγουρα να τοποθετηθούν πιο αξιόπιστα (με το ανάλογο κόστος) και αν η μέτρηση της δεξαμενής προτιμηθεί να γίνει με αισθητήρα υπερήχων, αυτός πρέπει να είναι IP65 και όχι ο κλασικός που χρησιμοποιήθηκε (λόγο χαμηλού κόστους) [50]. Ακόμα, πριν χρησιμοποιηθεί σε μια πραγματική εγκατάσταση, θα πρέπει η πλακέτα της συσκευής να τοποθετηθεί σε ένα κουτί, με προδιαγραφές IP65. Ολοκληρώνοντας, αν κριθεί ότι όσα αναφέρθηκαν, δεν μπορούν να υποστηριχθούν από το συγκεκριμένο επεξεργαστή, πρέπει να υπάρξει η αντικατάστασή του, από κάποιον με ακόμη καλύτερα χαρακτηριστικά, από του υπάρχοντος. Επίσης, θα χρειαστεί να γίνει και ο απαραίτητος επανασχεδιασμός, της πλακέτας.

Σε επίπεδο λογισμικού, βελτιώσεις μπορούν να υπάρξουν στον κώδικα τόσο της συσκευής όσο και του συστήματος. Μπορεί να γίνει αναθεώρηση κάποιων πρακτικών που ακολουθήθηκαν, ώστε ο κώδικας να έχει λιγότερες γραμμές και να καταστεί ευκολότερα συντηρήσιμος. Όσον αφορά την διαδικτυακή εφαρμογή, μια βελτίωση αποτελεί η προσθήκη ενός εικονιδίου δίπλα στο μενού, όπου με την είσοδο του ο χρήστης, θα ενημερώνεται αν υπάρχουν νέες ανακοινώσεις, που δεν έχει διαβάσει. Επίσης, να δημιουργηθούν οι κατάλληλες υποδομές, ώστε να εμφανίζονται και δεδομένα που αφορούν το Φυσικό

Αέριο. Ακόμα μια βελτίωση, αφορά τα διαγράμματα της θερμοκρασίας και της υγρασίας. Μπορεί να γίνει αλλαγή της τρέχουσας προβολής, με μια που τα δεδομένα θα εμφανίζονται κυλιόμενα με τη χρήση του ποντικού. Αυτό κρίνεται αναγκαίο, όταν τα δεδομένα είναι πολλά, πράγμα που συνεπάγεται ότι οι χρονικές στιγμές θα συνωστίζονται, σε ένα περιορισμένο πλαίσιο. Όλα αυτά βεβαίως, αφού πρώτα στηθεί το σύστημα σε ένα παραγωγικό server, που θα μπορεί να υποστηρίξει την ταυτόχρονη προσβασιμότητα πολλών χρηστών. Αυτό, μιας και για τις ανάγκες της προσομοίωσης της εργασίας, χρησιμοποιήθηκε ο server ανάπτυξης λογισμικού και στήθηκε σε οικιακό υπολογιστή (με σχεδόν μηδενική προσβασιμότητα). Τέλος, πέρα από την ιστοσελίδα, δύναται να δημιουργηθεί και η αντίστοιχη εφαρμογή για κινητά τηλέφωνα, με λειτουργικό iOS ή Android.

## ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] Μ. Βλαχογιάννης & Ν. Καμπούρας, Σχεδιασμός Εγκατάστασης Κεντρικής Θέρμανσης Κτιρίων (Εκπαίδευση Εξεταστών). ΤΕΙ Θεσσαλίας Τμήμα Μηχανολόγων Μηχανικών Τ.Ε., 2014. [Online]. Available: <https://www.teilar.gr/dbData/Dimosieyseis/ΣΧΕΔΙΑΣΜΟΣ%20ΕΓΚΑΤΑΣΤΑΣΗΣ%20ΚΕΝΤΡΙΚΗΣ%20ΘΕΡΜΑΝΣΗΣ.pdf>
- [2][Online]. Available: [https://www.kmavroidis.gr/wireless\\_timemetric.pdf](https://www.kmavroidis.gr/wireless_timemetric.pdf)
- [3][Online]. Available: <http://www.pdaenergy.it/>
- [4] Κ. Καλοβρέκτης & Ν. Κατέβας, Αισθητήρες Μέτρησης και Ελέγχου (2η Έκδοση-Νέα βελτιωμένη έκδοση). Θεσσαλονίκη, GR: ΕΚΔΟΣΕΙΣ ΤΖΙΟΛΑ, 2015.
- [5] Albert Malvino & David J. Bates, ΗΛΕΚΤΡΟΝΙΚΗ Αρχές και Εφαρμογές (7η Έκδοση). Θεσσαλονίκη, Ελλάδα: ΕΚΔΟΣΕΙΣ ΤΖΙΟΛΑ, 2014.
- [6][Online]. Available: <https://www.st.com/resource/en/datasheet/stm32f103c8.pdf>
- [7][Online]. Available: [https://www.st.com/resource/en/user\\_manual/um1850-description-of-stm32f1-hal-and-lowlayer-drivers-stmicroelectronics.pdf](https://www.st.com/resource/en/user_manual/um1850-description-of-stm32f1-hal-and-lowlayer-drivers-stmicroelectronics.pdf)
- [8] Α. Αλεξανδρής, ΟΠΤΟΗΛΕΚΤΡΟΝΙΚΗ Θεωρία-Εφαρμογές-Πειράματα (2η Έκδοση). Θεσσαλονίκη, GR: ΕΚΔΟΣΕΙΣ ΤΖΙΟΛΑ, 2010.
- [9][Online]. Available: [http://www.geo.auth.gr/courses/gmc/gmc318y/xf/pdf/Geo\\_Humidity.pdf](http://www.geo.auth.gr/courses/gmc/gmc318y/xf/pdf/Geo_Humidity.pdf)
- [10][Online]. Available: <https://www.sparkfun.com/datasheets/Sensors/Temperature/DHT22.pdf>
- [11][Online]. Available: <https://cdn.sparkfun.com/datasheets/Sensors/Proximity/HCSR04.pdf>
- [12][Online]. Available: <https://grobotronics.com/images/promo/MQ2.pdf>
- [13][Online]. Available: [http://data.leocom.kr/datasheets/258587\\_44978.pdf](http://data.leocom.kr/datasheets/258587_44978.pdf)
- [14][Online]. Available: <https://module-center.com/administrator/files/UploadFile/dc-mini-submersible-water-pump.pdf>
- [15][Online]. Available: <https://www.digikey.com/htmldatasheets/production/1640716/0/0/1/ILI9341-Datasheet.pdf>
- [16][Online]. Available: <https://arduinojson.org/v6/api/jsondocument/>
- [17] Χ. Ακριβιάδης & Κ. Βραχνής, Σχεδίαση και υλοποίηση συσκευής περιέλιξης πηνίων και μετασχηματιστών (Πτυχιακή Εργασία). Σχολή Μηχανικών, Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων Δι.Πα.Ε., 2021.
- [18][Online]. Available: <https://www.mysql.com/products/workbench/>
- [19][Online]. Available: <https://dev.mysql.com/doc/refman/8.0/en/integer-types.html>
- [20] V. Sahaan & R. H. Sianipar, A Guide to Python GUI Programming with MySQL. SPARTA PUBLISHING, January 2016.
- [21] C. Bell, Introducing MySQL Shell. Apress, September 2019.
- [22][Online]. Available: <https://flask.palletsprojects.com/en/2.2.x/>

- [23][Online]. Available: <https://docs.djangoproject.com/en/4.2/>
- [24][Online]. Available: <https://flask-sqlalchemy.palletsprojects.com/en/3.0.x/>
- [25] I. Maia, Building Web Applications with Flask. UK: Packt Publishing Ltd, June 2015.
- [26] J. Stouffer, Mastering Flask. Packt Publishing, September 2015.
- [27] M. Lathkar, Building Web Apps with Python and Flask. BPB, 2021.
- [28] K. Renal, Building REST APIs with Flask Create Python Web Services with MySQL. CA: Apress Berkeley, September 2019.
- [29][Online]. Available: <https://docs.sqlalchemy.org/en/20/dialects/mysql.html>
- [30] M. Copperwaite & C. Leifer, Learning Flask Framework. Packt Publishing, November 2015.
- [31] G. Hillar, Building RESTful Python Web Services. Packt Publishing, October 2016.
- [32] M. Ramm & M. Bayer & B. Rhodes, SQLAlchemy Database Access Using Python. Addison Wesley Professional, October 2011.
- [33][Online]. Available: <https://docs.python.org/3/library/datetime.html>
- [34][Online]. Available: <https://flask-wtf.readthedocs.io/en/1.0.x/>
- [35][Online]. Available: <https://wtforms.readthedocs.io/en/2.3.x/>
- [36][Online]. Available: <https://flask-bcrypt.readthedocs.io/en/1.0.1/>
- [37][Online]. Available: [https://itsdangerous.palletsprojects.com/en/2.1.x/url\\_safe/](https://itsdangerous.palletsprojects.com/en/2.1.x/url_safe/)
- [38][Online]. Available: <https://medium.com/@stevenrmonaghan/password-reset-with-flask-mail-protocol-ddcdcf190968>
- [39][Online]. Available: <https://mailtrap.io/blog/flask-email-sending/>
- [40][Online]. Available: <https://pythonhosted.org/Flask-Mail/>
- [41][Online]. Available: <https://flask-login.readthedocs.io/en/latest/>
- [42][Online]. Available: <https://www.postman.com/product/tools/>
- [43][Online]. Available: <https://docs.celeryq.dev/en/stable/userguide/periodic-tasks.html>
- [44][Online]. Available: <https://devcenter.heroku.com/articles/clock-processes-python>
- [45][Online]. Available: <https://flask.palletsprojects.com/en/2.2.x/patterns/celery/>
- [46][Online]. Available: <https://www.chartjs.org/docs/latest/>
- [47][Online]. Available: <https://getbootstrap.com/docs/4.3/components/navbar/>
- [48][Online]. Available: <https://flask-uploads.readthedocs.io/en/latest/>
- [49][Online]. Available: <https://flask.palletsprojects.com/en/2.2.x/patterns/fileuploads/>
- [50][Online]. Available: <https://web.archive.org/web/20130522115323/http://www.cemag.com/archive/06/ARG/bisenius.htm>



## Πηγές εικόνων

[51][Online]. Available: [42325241/ΩΡΟΜΕΤΡΗΤΗΣ-ΠΙΟΡΤΑΣ-ΠΑΓΑΣ-ΑΝΑΛΟΓΙΚΟΣ-OEM-22-00201.html](https://www.skrouz.gr/s/42325241/ΩΡΟΜΕΤΡΗΤΗΣ-ΠΙΟΡΤΑΣ-ΠΑΓΑΣ-ΑΝΑΛΟΓΙΚΟΣ-OEM-22-00201.html)

[52][Online]. Available: <https://www.skrouz.gr/s/28201220/Ωρομετρητής-ράγας-ψηφιακός-DHC15L-EL-TECH.html>

[53][Online]. Available: <https://hit.sbt.siemens.com/RWD/app.aspx?RC=GR&lang=el&MODULE=Catalog&ACTION=ShowProduct&KEY=S55561-F179>

[54][Online]. Available: <https://www.skrouz.gr/s/41452005/Θερμιδομετρητής-Hydrocal-M4-3-4-B-METERS.html>

[55][Online]. Available: [https://os.mbed.com/media/uploads/hudakz/stm32f103c8t6\\_pinout\\_voltage01.png](https://os.mbed.com/media/uploads/hudakz/stm32f103c8t6_pinout_voltage01.png)

[56][Online]. Available: <https://howtomechatronics.com/wp-content/uploads/2022/02/HC-SR04-Ultrasonic-Sensor-Pinout.png?ezimgfmt=ng:webp/ngcb2>

[57][Online]. Available: [https://components101.com/sites/default/files/component\\_pin/MQ2-Gas-sensor-Pinout.jpg](https://components101.com/sites/default/files/component_pin/MQ2-Gas-sensor-Pinout.jpg)

[58][Online]. Available: <https://robodoc.sgp1.cdn.digitaloceanspaces.com/production/uploads/products/4ea72b2f3213f2b807771384b8b254c2.png>

[59][Online]. Available: <https://hackster.imgix.net/uploads/attachments/987713/ESP8266-PINOUT.png?auto=compress%2Cformat&w=740&h=555&fit=max>

[60][Online]. Available: [https://wolles-elektronikkiste.de/wp-content/uploads/2019/05/ESP01\\_mit\\_Programmer-1.jpg](https://wolles-elektronikkiste.de/wp-content/uploads/2019/05/ESP01_mit_Programmer-1.jpg)

[61][Online]. Available: "Designed by starline / Freepik"

[62][Online]. Available: <https://ucarecdn.com/0295176b-b17a-4106-b3b0-5bf2b14365e6/>

[63][Online]. Available: [https://www.juicymedia.co.uk/application/files/4615/1838/4923/html\\_css\\_js.png](https://www.juicymedia.co.uk/application/files/4615/1838/4923/html_css_js.png)

[64][Online]. Available: [https://www.linuxjournal.com/sites/default/files/styles/360\\_250/public/nodeimage/story/sqlalchemy.jpg?itok=BZAG1c3k](https://www.linuxjournal.com/sites/default/files/styles/360_250/public/nodeimage/story/sqlalchemy.jpg?itok=BZAG1c3k)

[65][Online]. Available: [https://www.pngitem.com/pimgs/m/360-3600842\\_internet-of-things-iot-device-icon-transparent-hd.png](https://www.pngitem.com/pimgs/m/360-3600842_internet-of-things-iot-device-icon-transparent-hd.png)

[66][Online]. Available: [https://www.google.com/imgres?imgurl=https%3A%2F%2Fwww.pnggg.com%2Fpngimages%2F391%2F430%2Fpng-clipart-bootstrap-full-logo-tech-companies.png&tbnid=UehEiKWYDWzXM&vet=12ahUKEwixtJmA5Lv\\_AhWakqQKHdKKD00QMygDegUIARC2AQ..i&imgrefurl=https%3A%2F%2Fwww.pnggg.com%2Fen%2Fsearch%3Fq%3Dbootstrap%2BLogo&docid=36FTwhqlyAE5M&w=900&h=356&q=bootstrap%20logo%20png&ved=2ahUKEwixtJmA5Lv\\_AhWakqQKHdKKD00QMygDegUIARC2AQ#imgsrc=zYjzUREtZ7ZJQM&imgdii=EiSRCmy6e2eV8M](https://www.google.com/imgres?imgurl=https%3A%2F%2Fwww.pnggg.com%2Fpngimages%2F391%2F430%2Fpng-clipart-bootstrap-full-logo-tech-companies.png&tbnid=UehEiKWYDWzXM&vet=12ahUKEwixtJmA5Lv_AhWakqQKHdKKD00QMygDegUIARC2AQ..i&imgrefurl=https%3A%2F%2Fwww.pnggg.com%2Fen%2Fsearch%3Fq%3Dbootstrap%2BLogo&docid=36FTwhqlyAE5M&w=900&h=356&q=bootstrap%20logo%20png&ved=2ahUKEwixtJmA5Lv_AhWakqQKHdKKD00QMygDegUIARC2AQ#imgsrc=zYjzUREtZ7ZJQM&imgdii=EiSRCmy6e2eV8M)

[67][Online]. Available: [https://www.flaticon.com/free-icon/computer-screen-with-cogwheel-variant\\_30912](https://www.flaticon.com/free-icon/computer-screen-with-cogwheel-variant_30912)

[68][Online]. Available: [https://www.google.com/imgres?imgurl=https%3A%2F%2Flogos-world.net%2Fwp-content%2Fuploads%2F2021%2F10%2Fpython-Symbol.png&tbnid=wJ\\_8EiJgJRaS-M&vet=12ahUKEwioiPvZ57v\\_AhVH2aQKHabNAMQQMygDegUIARDDAQ..i&imgrefurl=https%3A%2F%2Flogos-world.net%2Fpython-logo%2F&docid=jGXJl4w5MRn50M&w=3840&h=2160&q=python%20logo&ved=2ahUKEwioiPvZ57v\\_AhVH2aQKHabNAMQQMygDegUIARDDAQ](https://www.google.com/imgres?imgurl=https%3A%2F%2Flogos-world.net%2Fwp-content%2Fuploads%2F2021%2F10%2Fpython-Symbol.png&tbnid=wJ_8EiJgJRaS-M&vet=12ahUKEwioiPvZ57v_AhVH2aQKHabNAMQQMygDegUIARDDAQ..i&imgrefurl=https%3A%2F%2Flogos-world.net%2Fpython-logo%2F&docid=jGXJl4w5MRn50M&w=3840&h=2160&q=python%20logo&ved=2ahUKEwioiPvZ57v_AhVH2aQKHabNAMQQMygDegUIARDDAQ)

[69][Online]. Available: <https://ih1.redbubble.net/image.2488655049.9084/st,small,507x507-pad,600x600,f8f8f8.jpg>

[70][Online]. Available: [https://img.freepik.com/premium-vector/110921\\_535345-3821.jpg?w=740](https://img.freepik.com/premium-vector/110921_535345-3821.jpg?w=740)

[71][Online]. Available: <https://cdn-icons-png.flaticon.com/512/2232/2232137.png>

[72][Online]. Available: [https://www.google.com/imgres?imgurl=https%3A%2F%2Fthumbs.dreamstime.com%2Fb%2Fhumidity-icon-editable-bold-outline-color-fill-design-vector-illustration-251848453.jpg&tbnid=dmSulkVsOnum5M&vet=12ahUKEwjoyPKbkrv\\_AhVZNUwKHefqAo8QMygIegUIARDPAQ..i&imgrefurl=https%3A%2F%2Fwww.dreamstime.com%2Fhumidity-icon-weather-symbol-simple-flat-design-forecast-sign-web-site-button-mobile-app-ui-image174252295&docid=MsZgipa8e-i-HM&w=800&h=800&q=measure%20temp%20hum%20icon&ved=2ahUKEwjoyPKbkrv\\_AhVZNUwKHefqAo8QMygIegUIARDPAQ](https://www.google.com/imgres?imgurl=https%3A%2F%2Fthumbs.dreamstime.com%2Fb%2Fhumidity-icon-editable-bold-outline-color-fill-design-vector-illustration-251848453.jpg&tbnid=dmSulkVsOnum5M&vet=12ahUKEwjoyPKbkrv_AhVZNUwKHefqAo8QMygIegUIARDPAQ..i&imgrefurl=https%3A%2F%2Fwww.dreamstime.com%2Fhumidity-icon-weather-symbol-simple-flat-design-forecast-sign-web-site-button-mobile-app-ui-image174252295&docid=MsZgipa8e-i-HM&w=800&h=800&q=measure%20temp%20hum%20icon&ved=2ahUKEwjoyPKbkrv_AhVZNUwKHefqAo8QMygIegUIARDPAQ)

[73][Online]. Available: [https://images.ctfassets.net/ity165ek7v1z/1Gwy0Hwdz3poQOzjycPJtu/4440d5d7be7c288c746bd96392840ec3/rsz\\_12mcu.png?w=600&h=400&q=80&fm=png](https://images.ctfassets.net/ity165ek7v1z/1Gwy0Hwdz3poQOzjycPJtu/4440d5d7be7c288c746bd96392840ec3/rsz_12mcu.png?w=600&h=400&q=80&fm=png)

[74][Online]. Available: [https://www.google.com/imgres?imgurl=https%3A%2F%2Fupload.wikimedia.org%2Fwikipedia%2Fcommons%2Fthumb%2F8%2F87%2FJinja\\_software\\_logo.svg%2F1200px-Jinja\\_software\\_logo.svg.png&tbnid=EyK6g5VfZHtZhM&vet=12ahUKEwi6v6jqkLv\\_AhXQyqQKHa1kCK0QMygBegUIARCzAQ..i&imgrefurl=https%3A%2F%2Fen.wikipedia.org%2Fwiki%2FJinja\\_\(template\\_engine\)&docid=Amxst16Myjz7aM&w=1200&h=480&q=jinja%20templates&ved=2ahUKEwi6v6jqkLv\\_AhXQyqQKHa1kCK0QMygBegUIARCzAQ](https://www.google.com/imgres?imgurl=https%3A%2F%2Fupload.wikimedia.org%2Fwikipedia%2Fcommons%2Fthumb%2F8%2F87%2FJinja_software_logo.svg%2F1200px-Jinja_software_logo.svg.png&tbnid=EyK6g5VfZHtZhM&vet=12ahUKEwi6v6jqkLv_AhXQyqQKHa1kCK0QMygBegUIARCzAQ..i&imgrefurl=https%3A%2F%2Fen.wikipedia.org%2Fwiki%2FJinja_(template_engine)&docid=Amxst16Myjz7aM&w=1200&h=480&q=jinja%20templates&ved=2ahUKEwi6v6jqkLv_AhXQyqQKHa1kCK0QMygBegUIARCzAQ)

[75][Online]. Available: <https://iconprocon.com/wp-content/uploads/2018/11/ultrapro-1000-ultrasonic-level-sensor-details1-600x600.jpg>

[76][Online]. Available: [https://www.matelecaustralia.com.au/content/product/regular/thumb/FSW\\_12570-1584-692.jpg](https://www.matelecaustralia.com.au/content/product/regular/thumb/FSW_12570-1584-692.jpg)

[77][Online]. Available: <https://cdn-icons-png.flaticon.com/128/2084/2084390.png>

[78][Online]. Available: [https://www.matelecaustralia.com.au/content/product/regular/thumb/FSW\\_12570-1584-692.jpg](https://www.matelecaustralia.com.au/content/product/regular/thumb/FSW_12570-1584-692.jpg)

[79][Online]. Available:  
[https://as1.ftcdn.net/v2/jpg/01/87/66/90/1000\\_F\\_187669000\\_XBq3CrYi8doqCwtmwKXJfjUT1hMiKcwh.jpg](https://as1.ftcdn.net/v2/jpg/01/87/66/90/1000_F_187669000_XBq3CrYi8doqCwtmwKXJfjUT1hMiKcwh.jpg)

[80][Online]. Available: [https://www.flaticon.com/free-icon/rss\\_408783](https://www.flaticon.com/free-icon/rss_408783)

[81][Online]. Available: "Designed by D3Images / Freepik"

## ΠΑΡΑΡΤΗΜΑ Α : Κώδικας για το STM32F103C8T6

```
/* USER CODE BEGIN Header */
/**
 * *****
 * @file      : main.c
 * @brief     : Main program body
 * *****
 * @attention
 *
 * Copyright (c) 2023 STMicroelectronics.
 * All rights reserved.
 *
 * This software is licensed under terms that can be found in the LICENSE file
 * in the root directory of this software component.
 * If no LICENSE file comes with this software, it is provided AS-IS.
 *
 * *****
 */
/* USER CODE END Header */
/* Includes -----*/
#include "main.h"

/* Private includes -----*/
/* USER CODE BEGIN Includes */
#include <stdio.h>
#include <stdlib.h>

/* BSP LCD driver */
#include "stm32_adafruit_lcd.h"
```

```

/* BSP TS driver */
#include "stm32_adafruit_ts.h"

char str1[20];

int set_pass[4], temp_pass[4], check_pass[4], set_apod_ar[3], temp_apod_ar[3], temp_apod_ar1[3];
int m=0, g=0, sp=0, spok=0, test=0, testout=0, ok=2, set_cursor_x = 85, set_cursor_y = 80, wifi=0;
int ethernet=0, wifi_on=0, ethernet_on=0;
int start_w_e=0, start_m=0, gas=0, petrol=0, gas_on=0, petrol_on=0, testa=0, y=0, z=0;
int set_apod=1, change_pass=1, start=0, temp_st=0;
TS_StateTypeDef ts;

/* USER CODE END Includes */

/* Private typedef -----*/
/* USER CODE BEGIN PTD */

/* USER CODE END PTD */

/* Private define -----*/
/* USER CODE BEGIN PD */
/* USER CODE END PD */

/* Private macro -----*/
/* USER CODE BEGIN PM */

/* USER CODE END PM */

/* Private variables -----*/
SPI_HandleTypeDef hspi1;
SPI_HandleTypeDef hspi2;

```

```
DMA_HandleTypeDef hdma_spi1_tx;
```

```
TIM_HandleTypeDef htim2;
```

```
TIM_HandleTypeDef htim3;
```

```
TIM_HandleTypeDef htim4;
```

```
UART_HandleTypeDef huart3;
```

```
/* USER CODE BEGIN PV */
```

```
int s1, s2, s3, s4, s5, s6;
```

```
int s1_f=0, s2_f=0, s3_f=0, s4_f=0, s5_f=0, s6_f=0, smoke=0, overflow=0, leak=0;
```

```
int f_l_l_l=0, f_l_h_l=0, f_h_l_l=0, f_h_h_l=0, f_l_l_h=0, f_l_h_h=0, f_h_l_h=0, f_h_h_h=0;
```

```
int tim4_sec=0, tim4_min=0, tim5_min=0, count_sec=0, count_min=0, count_min_uic=0,  
time_to_read=0, send_distance=0, uic_send=0;
```

```
uint32_t IC_Val1 = 0;
```

```
uint32_t IC_Val2 = 0;
```

```
uint32_t Difference = 0;
```

```
uint8_t Is_First_Captured = 0; // is the first value captured ?
```

```
float Distance = 0.0;
```

```
char buffer[12];
```

```
#define DHT22_PORT GPIOA
```

```
#define DHT22_PIN GPIO_PIN_0
```

```
uint8_t RH1, RH2, TC1, TC2, SUM, CHECK;
```

```
uint32_t pMillis, cMillis;
```

```
float tCelsius = 0;
```

```
float tFahrenheit = 0;
```

```
float RH = 0;
```

```
#define DHT22_IN_PORT GPIOA
```

```
#define DHT22_IN_PIN GPIO_PIN_4
```

```

uint8_t RH1_IN, RH2_IN, TC1_IN, TC2_IN, SUM_IN, CHECK_IN;
uint32_t pMillis_IN, cMillis_IN;
float tCelsius_IN = 0;
float tFahrenheit_IN = 0;
float RH_IN = 0;

```

```

void HAL_GPIO_EXTI_Callback(uint16_t GPIO_Pin)

```

```

{

```

```

    GPIO_InitTypeDef GPIO_InitStructure = {0};

```

```

/*                      Sensor1 Elegxos plhmmuras xwrou                      */

```

```

    if (GPIO_Pin == sens1_Pin)

```

```

    {

```

```

        HAL_NVIC_DisableIRQ(EXTI9_5_IRQn);

```

```

        for (int i=0; i<10000; i++) {}

```

```

        s1 = HAL_GPIO_ReadPin(GPIOA, GPIO_PIN_9);

```

```

        if (s1 == 1)

```

```

        {

```

```

            s1 = HAL_GPIO_ReadPin(GPIOA, GPIO_PIN_9);

```

```

            if (s1 == 1)

```

```

            {

```

```

                s1_f=1;

```

```

                GPIO_InitStructure.Pin = sens1_Pin;

```

```

                GPIO_InitStructure.Mode = GPIO_MODE_IT_FALLING;

```

```

                HAL_NVIC_EnableIRQ(EXTI9_5_IRQn);

```

```

            }

```

```

    }

    if (s1 == 0)
    {
        s1 = HAL_GPIO_ReadPin(GPIOA, GPIO_PIN_9);

        if (s1 == 0)
        {
            s1_f=0;

            GPIO_InitStruct.Pin = sens1_Pin;
            GPIO_InitStruct.Mode = GPIO_MODE_IT_RISING;
            HAL_NVIC_EnableIRQ(EXTI9_5_IRQn);
        }
    }
}

```

```

/*          Sensor2 Elegxos plhmmuras xwrou          */

```

```

if (GPIO_Pin == sens2_Pin)
{
    HAL_NVIC_DisableIRQ(EXTI15_10_IRQn);
    for (int i=0; i<10000; i++) {}
    s2 = HAL_GPIO_ReadPin(GPIOA, GPIO_PIN_10);

    if (s2 == 1)
    {
        s2 = HAL_GPIO_ReadPin(GPIOA, GPIO_PIN_10);
        if (s2 == 1)

```



```

        {
            s2_f=1;

            GPIO_InitStruct.Pin = sens2_Pin;
            GPIO_InitStruct.Mode = GPIO_MODE_IT_FALLING;
            HAL_NVIC_EnableIRQ(EXTI15_10_IRQn);
        }

    }

    if (s2 == 0)
    {
        s2 = HAL_GPIO_ReadPin(GPIOA, GPIO_PIN_10);

        if (s2 == 0)
        {
            s2_f=0;

            GPIO_InitStruct.Pin = sens2_Pin;
            GPIO_InitStruct.Mode = GPIO_MODE_IT_RISING;
            HAL_NVIC_EnableIRQ(EXTI15_10_IRQn);
        }
    }
}

```

```

/*          Sensor3 Elegxos diarrohs deksamenhs          */

```

```

if (GPIO_Pin == sens3_Pin)
{
    HAL_NVIC_DisableIRQ(EXTI15_10_IRQn);
}

```

```

for (int i=0; i<250000; i++) {}

s3 = HAL_GPIO_ReadPin(GPIOC, GPIO_PIN_14);

if (s3 == 1)
{
    s3 = HAL_GPIO_ReadPin(GPIOC, GPIO_PIN_14);
    if (s3 == 1)
    {
        s3_f=1;

        GPIO_InitStruct.Pin = sens3_Pin;
        GPIO_InitStruct.Mode = GPIO_MODE_IT_FALLING;
        HAL_NVIC_EnableIRQ(EXTI15_10_IRQn);
    }
}

if (s3 == 0)
{
    s3 = HAL_GPIO_ReadPin(GPIOC, GPIO_PIN_14);

    if (s3 == 0)
    {
        s3_f=0;

        GPIO_InitStruct.Pin = sens3_Pin;
        GPIO_InitStruct.Mode = GPIO_MODE_IT_RISING;
        HAL_NVIC_EnableIRQ(EXTI15_10_IRQn);
    }
}
}

```

```

/*          Sensor4 Elegxos diarrohs deksamenhs          */

if (GPIO_Pin == sens4_Pin)
{
    HAL_NVIC_DisableIRQ(EXTI15_10_IRQn);
    for (int p=0; p<600000; p++) {}
    s4 = HAL_GPIO_ReadPin(GPIOC, GPIO_PIN_15);

    if (s4 == 1)
    {
        s4 = HAL_GPIO_ReadPin(GPIOC, GPIO_PIN_15);

        if (s4 == 1)
        {
            s4_f=1;

            GPIO_InitStruct.Pin = sens4_Pin;
            GPIO_InitStruct.Mode = GPIO_MODE_IT_FALLING;
            HAL_NVIC_EnableIRQ(EXTI15_10_IRQn);
        }

/* Gia debug

        else
        {
            s4_f=44;
        }

*/
    }
}

```

```

if (s4 == 0)
{
    s4 = HAL_GPIO_ReadPin(GPIOC, GPIO_PIN_15);

    if (s4 == 0)
    {
        s4_f=0;

        GPIO_InitStruct.Pin = sens4_Pin;
        GPIO_InitStruct.Mode = GPIO_MODE_IT_RISING;
        HAL_NVIC_EnableIRQ(EXTI15_10_IRQn);
    }
}
}

```

```

/*          Sensor5 Elegxos uparkshs kapnou          */

```

```

if (GPIO_Pin == sens5_Pin) //energopoieite me 0
{
    HAL_NVIC_DisableIRQ(EXTI1_IRQn);
    for (int i=0; i<10000; i++) {}
    s5 = HAL_GPIO_ReadPin(GPIOA, GPIO_PIN_1);

    if (s5 == 0)
    {
        s5 = HAL_GPIO_ReadPin(GPIOA, GPIO_PIN_1);

        if (s5 == 0)
        {
            s5_f=1;

```

```

        GPIO_InitStruct.Pin = sens5_Pin;
        GPIO_InitStruct.Mode = GPIO_MODE_IT_RISING;
        HAL_NVIC_EnableIRQ(EXTI1_IRQn);
    }
}

if (s5 == 1)
{
    s5 = HAL_GPIO_ReadPin(GPIOA, GPIO_PIN_1);

    if (s5 == 1)
    {
        s5_f=0;

        GPIO_InitStruct.Pin = sens5_Pin;
        GPIO_InitStruct.Mode = GPIO_MODE_IT_FALLING;
        HAL_NVIC_EnableIRQ(EXTI1_IRQn);
    }
}
}

```

```

/*          Sensor6 Elegxos ON/OFF kausthra          */

```

```

if (GPIO_Pin == sens6_Pin) // energopoieite me 0
{
    HAL_NVIC_DisableIRQ(EXTI9_5_IRQn);
    for (int i=0; i<250000; i++) {}
    s6 = HAL_GPIO_ReadPin(GPIOB, GPIO_PIN_8);
}

```

```

if (s6 == 0)
{
    s6 = HAL_GPIO_ReadPin(GPIOB, GPIO_PIN_8);

    if (s6 == 0)
    {
        HAL_GPIO_WritePin(GPIOC, GPIO_PIN_13, GPIO_PIN_RESET);

//pcb_led off

        s6_f=1;

        GPIO_InitStruct.Pin = sens6_Pin;
        GPIO_InitStruct.Mode = GPIO_MODE_IT_RISING;
        HAL_NVIC_EnableIRQ(EXTI9_5_IRQn);
    }
}

if (s6 == 1)
{
    s6 = HAL_GPIO_ReadPin(GPIOB, GPIO_PIN_8);

    if (s6 == 1)
    {
        HAL_GPIO_WritePin(GPIOC, GPIO_PIN_13, GPIO_PIN_SET);

//pcb_led off

        s6_f=0;

        GPIO_InitStruct.Pin = sens6_Pin;
        GPIO_InitStruct.Mode = GPIO_MODE_IT_FALLING;
        HAL_NVIC_EnableIRQ(EXTI9_5_IRQn);
    }
}
}

```

```
}
```

```
void uic (void) // for last timestamp update internet connection
```

```
{
```

```
    char *s_email = "UIC";
```

```
    HAL_UART_Transmit(&huart3, (uint8_t *)&buffer, sprintf(buffer, s_email), 1000);
```

```
    HAL_Delay(10);
```

```
    for (int i=0; i<11; i++) // katharizei o buffer
```

```
    {
```

```
        buffer[i]=0;
```

```
    }
```

```
}
```

```
void event (void) // for send event's email
```

```
{
```

```
    //HAL_GPIO_WritePin(GPIOC, GPIO_PIN_13, GPIO_PIN_RESET); //pcb_led off
```

```
    if (f_h_h_h == 1) // L O S
```

```
    {
```

```
        overflow=1, leak=1, smoke=1;
```

```
        char s_email[6] = "EVLOS";
```

```
        HAL_UART_Transmit(&huart3, (uint8_t *)&buffer, sprintf(buffer, s_email), 1000);
```

```
        HAL_Delay(10);
```

```
    }
```

```
    else if (f_h_l_h == 1) // O S
```

```
    {
```

```
overflow=1, leak=0, smoke=1;
char s_email[5] = "EVOS";
HAL_UART_Transmit(&huart3, (uint8_t * )&buffer, sprintf(buffer, s_email), 1000);
HAL_Delay(10);
}
```

```
else if (f_1_h_h == 1) // L S
{
overflow=0, leak=1, smoke=1;
char s_email[5] = "EVLS";
HAL_UART_Transmit(&huart3, (uint8_t * )&buffer, sprintf(buffer, s_email), 1000);
HAL_Delay(10);
}
```

```
else if (f_1_l_h == 1) // S
{
overflow=0, leak=0, smoke=1;
char s_email[5] = "EVS";
HAL_UART_Transmit(&huart3, (uint8_t * )&buffer, sprintf(buffer, s_email), 1000);
HAL_Delay(10);
}
```

```
else if (f_h_h_l == 1) // O L
{
overflow=1, leak=1, smoke=0;
char s_email[5] = "EVOL";
HAL_UART_Transmit(&huart3, (uint8_t * )&buffer, sprintf(buffer, s_email), 1000);
HAL_Delay(10);
}
```

```
else if (f_h_l_l == 1) // O
{
```



```

overflow=1, leak=0, smoke=0;
char s_email[5] = "EVO";
HAL_UART_Transmit(&huart3, (uint8_t * )&buffer, sprintf(buffer, s_email), 1000);
HAL_Delay(10);
}

else if (f_1_h_1 == 1) // L
{
overflow=0, leak=1, smoke=0;
char s_email[5] = "EVL";
HAL_UART_Transmit(&huart3, (uint8_t * )&buffer, sprintf(buffer, s_email), 1000);
HAL_Delay(10);
}

for (int i=0; i<10; i++)
{
buffer[i]=0;
}

count_min_uic=0; // mhdenizei giati tha kanei post apo to event kai etsi sto backend tha parei
last timestamp
uic_send=0;
}

void buzzer (void)
{
HAL_GPIO_TogglePin(GPIOA, GPIO_PIN_12);
HAL_Delay(500);
}

void delay (uint16_t time) //microseconds

```

```

{
    __HAL_TIM_SET_COUNTER(&htim2, 0);
    while (__HAL_TIM_GET_COUNTER (&htim2) < time);
}

void HAL_TIM_IC_CaptureCallback(TIM_HandleTypeDef *htim)
{
    if (htim->Channel == HAL_TIM_ACTIVE_CHANNEL_2) // if the interrupt source is
channel2
    {
        if (Is_First_Captured==0) // if the first value is not captured
        {
            IC_Val1 = HAL_TIM_ReadCapturedValue(htim, TIM_CHANNEL_2); // read
the first value

            Is_First_Captured = 1; // set the first captured as true
            // Now change the polarity to falling edge
            __HAL_TIM_SET_CAPTUREPOLARITY(htim,          TIM_CHANNEL_2,
TIM_INPUTCHANNELPOLARITY_FALLING);
        }

        else if (Is_First_Captured==1) // if the first is already captured
        {
            IC_Val2 = HAL_TIM_ReadCapturedValue(htim, TIM_CHANNEL_2); //
read second value

            __HAL_TIM_SET_COUNTER(htim, 0); // reset the counter

            if (IC_Val2 > IC_Val1)
            {
                Difference = IC_Val2-IC_Val1;
            }

            else if (IC_Val1 > IC_Val2)
            {

```

```

        Difference = (0xffff - IC_Val1) + IC_Val2;
    }

    Distance = Difference * .034/2;
    Is_First_Captured = 0; // set it back to false

    // set polarity to rising edge
    __HAL_TIM_SET_CAPTUREPOLARITY(htim,      TIM_CHANNEL_2,
TIM_INPUTCHANNELPOLARITY_RISING);
    __HAL_TIM_DISABLE_IT(&htim2, TIM_IT_CC1);

    if ((start > 0 && start != 3) && Distance >= 2)
    {
        HAL_UART_Transmit(&huart3, (uint8_t * )&buffer, sprintf(buffer,
"D%.1f", Distance),1000);
    }

    if (start == 3 && Distance >= 2) // plhrwsh deksamenhs prinmai meta
    {
        HAL_UART_Transmit(&huart3, (uint8_t * )&buffer, sprintf(buffer, "f%.1f",
Distance),1000);
    }

    for (int i=0; i<11; i++) // katharismos buffer
    {
        buffer[i]=0;
    }

}

}

}

```

```

void HCSR04_Read (void)
{
    HAL_GPIO_WritePin(GPIOA, GPIO_PIN_15, GPIO_PIN_SET); // pull the TRIG pin HIGH
    delay(10); // wait for 10 us
    HAL_GPIO_WritePin(GPIOA, GPIO_PIN_15, GPIO_PIN_RESET); // pull the TRIG pin low

    __HAL_TIM_ENABLE_IT(&htim2, TIM_IT_CC1);
}

```

```

uint8_t DHT22_Start (void)
{
    uint8_t Response = 0;
    GPIO_InitTypeDef GPIO_InitStructPrivate = {0};
    GPIO_InitStructPrivate.Pin = DHT22_PIN;
    GPIO_InitStructPrivate.Mode = GPIO_MODE_OUTPUT_PP;
    GPIO_InitStructPrivate.Speed = GPIO_SPEED_FREQ_LOW;
    GPIO_InitStructPrivate.Pull = GPIO_NOPULL;
    HAL_GPIO_Init(DHT22_PORT, &GPIO_InitStructPrivate); // set the pin as output
    HAL_GPIO_WritePin (DHT22_PORT, DHT22_PIN, 0); // pull the pin low
    delay (1300); // wait for 1300us
    HAL_GPIO_WritePin (DHT22_PORT, DHT22_PIN, 1); // pull the pin high
    delay (30); // wait for 30us
    GPIO_InitStructPrivate.Mode = GPIO_MODE_INPUT;
    GPIO_InitStructPrivate.Pull = GPIO_PULLUP;
    HAL_GPIO_Init(DHT22_PORT, &GPIO_InitStructPrivate); // set the pin as input
    delay (40);
    if (!(HAL_GPIO_ReadPin (DHT22_PORT, DHT22_PIN)))
    {
        delay (80);
        if ((HAL_GPIO_ReadPin (DHT22_PORT, DHT22_PIN))) Response = 1;
    }
}

```

```

}
pMillis = HAL_GetTick();
cMillis = HAL_GetTick();
while ((HAL_GPIO_ReadPin (DHT22_PORT, DHT22_PIN)) && pMillis + 2 > cMillis)
{
    cMillis = HAL_GetTick();
}
return Response;
}

```

```

uint8_t DHT22_Read (void)
{
    uint8_t a,b;
    for (a=0;a<8;a++)
    {
        pMillis = HAL_GetTick();
        cMillis = HAL_GetTick();
        while (!(HAL_GPIO_ReadPin (DHT22_PORT, DHT22_PIN)) && pMillis + 2 > cMillis)
        { // wait for the pin to go high
            cMillis = HAL_GetTick();
        }
        delay (40); // wait for 40 us
        if (!(HAL_GPIO_ReadPin (DHT22_PORT, DHT22_PIN))) // if the pin is low
            b&= ~(1<<(7-a));
        else
            b|= (1<<(7-a));
        pMillis = HAL_GetTick();
        cMillis = HAL_GetTick();
        while ((HAL_GPIO_ReadPin (DHT22_PORT, DHT22_PIN)) && pMillis + 2 > cMillis)
        { // wait for the pin to go low
            cMillis = HAL_GetTick();

```

```

    }
}
return b;
}

void Temp_Hum (void)
{
    if(DHT22_Start())
    {
        RH1 = DHT22_Read(); // First 8bits of humidity
        RH2 = DHT22_Read(); // Second 8bits of Relative humidity
        TC1 = DHT22_Read(); // First 8bits of Celsius
        TC2 = DHT22_Read(); // Second 8bits of Celsius
        SUM = DHT22_Read(); // Check sum
        CHECK = RH1 + RH2 + TC1 + TC2;
        if (CHECK == SUM)
        {
            if (TC1>127) // If TC1=10000000, negative temperature
            {
                tCelsius = (float)TC2/10*(-1);
            }
            else
            {
                tCelsius = (float)((TC1<<8)|TC2)/10;
            }
            tFahrenheit = tCelsius * 9/5 + 32;
            RH = (float) ((RH1<<8)|RH2)/10;
        }
    }

    if (start != 0)
    {

```

```

        HAL_UART_Transmit(&huart3, (uint8_t * )&buffer, sprintf(buffer, "T%.1f%.1f",
tCelsius,RH),1000);
        HAL_Delay(2500);
    }

    for (int i=0; i<11; i++) // katharismos buffer
    {
        buffer[i]=0;
    }

//    count_min_uic=0; // mhdenizei giati tha kanei post apo to temp-hum kai etsi sto backend tha
parei last timestamp
//    uic_send=0;
}

```

////////////////////////////////// IN TEMP\_HUM

```

uint8_t DHT22_Start_IN (void)
{
    uint8_t Response = 0;
    GPIO_InitTypeDef GPIO_InitStructure = {0};
    GPIO_InitStructure.Pin = DHT22_IN_PIN;
    GPIO_InitStructure.Mode = GPIO_MODE_OUTPUT_PP;
    GPIO_InitStructure.Speed = GPIO_SPEED_FREQ_LOW;
    GPIO_InitStructure.Pull = GPIO_NOPULL;
    HAL_GPIO_Init(DHT22_IN_PORT, &GPIO_InitStructure); // set the pin as output
    HAL_GPIO_WritePin (DHT22_IN_PORT, DHT22_IN_PIN, 0); // pull the pin low
    delay (1300); // wait for 1300us
    HAL_GPIO_WritePin (DHT22_IN_PORT, DHT22_IN_PIN, 1); // pull the pin high
    delay (30); // wait for 30us
    GPIO_InitStructure.Mode = GPIO_MODE_INPUT;
    GPIO_InitStructure.Pull = GPIO_PULLUP;
}

```

```

HAL_GPIO_Init(DHT22_IN_PORT, &GPIO_InitStructPrivate); // set the pin as input
delay (40);
if (!(HAL_GPIO_ReadPin (DHT22_IN_PORT, DHT22_IN_PIN)))
{
    delay (80);
    if ((HAL_GPIO_ReadPin (DHT22_IN_PORT, DHT22_IN_PIN))) Response = 1;
}
pMillis_IN = HAL_GetTick();
cMillis_IN = HAL_GetTick();
while ((HAL_GPIO_ReadPin (DHT22_IN_PORT, DHT22_IN_PIN)) && pMillis_IN + 2 >
cMillis_IN)
{
    cMillis_IN = HAL_GetTick();
}
return Response;
}

```

```

uint8_t DHT22_Read_IN (void)
{
    uint8_t a,b;
    for (a=0;a<8;a++)
    {
        pMillis_IN = HAL_GetTick();
        cMillis_IN = HAL_GetTick();
        while (!(HAL_GPIO_ReadPin (DHT22_IN_PORT, DHT22_IN_PIN)) && pMillis_IN + 2 >
cMillis_IN)
        { // wait for the pin to go high
            cMillis_IN = HAL_GetTick();
        }
        delay (40); // wait for 40 us
        if (!(HAL_GPIO_ReadPin (DHT22_IN_PORT, DHT22_IN_PIN))) // if the pin is low
            b&= ~(1<<(7-a));
    }
}

```



```

else
    b|= (1<<(7-a));
pMillis_IN = HAL_GetTick();
cMillis_IN = HAL_GetTick();
while ((HAL_GPIO_ReadPin (DHT22_IN_PORT, DHT22_IN_PIN)) && pMillis_IN + 2 >
cMillis_IN)
    { // wait for the pin to go low
        cMillis_IN = HAL_GetTick();
    }
}
return b;
}

```

```

void Temp_Hum_IN (void)
{
    if(DHT22_Start_IN())
    {
        RH1_IN = DHT22_Read_IN(); // First 8bits of humidity
        RH2_IN = DHT22_Read_IN(); // Second 8bits of Relative humidity
        TC1_IN = DHT22_Read_IN(); // First 8bits of Celsius
        TC2_IN = DHT22_Read_IN(); // Second 8bits of Celsius
        SUM_IN = DHT22_Read_IN(); // Check sum
        CHECK_IN = RH1_IN + RH2_IN + TC1_IN + TC2_IN;
        if (CHECK_IN == SUM_IN)
        {
            if (TC1>127) // If TC1=10000000, negative temperature
            {
                tCelsius_IN = (float)TC2_IN/10*(-1);
            }
            else
            {
                tCelsius_IN = (float)((TC1_IN<<8)|TC2_IN)/10;
            }
        }
    }
}

```

```

        tFahrenheit_IN = tCelsius_IN * 9/5 + 32;
        RH_IN = (float) ((RH1_IN<<8)|RH2_IN)/10;
    }
}

if (tim5_min == 2 && tCelsius_IN > 40.0) //(start != 0)
{
    char s_email[5] = "EVT";
    HAL_UART_Transmit(&huart3, (uint8_t *)&buffer, sprintf(buffer, s_email), 1000);
    HAL_Delay(10);
    //HAL_UART_Transmit(&huart3, (uint8_t *)&buffer, sprintf(buffer, "TI%.1f%.1f",
tCelsius,RH),1000);
}

else if(start!=0 && tim4_min ==2) // h ==5 gia kathe 5 lepta
{
    //HAL_Delay(300);
    HAL_UART_Transmit(&huart3, (uint8_t *)&buffer, sprintf(buffer, "I%.1f%.1f",
tCelsius_IN,RH_IN),1000);
    HAL_Delay(10);
}

for (int i=0; i<11; i++) // katharismos buffer
{
    buffer[i]=0;
}

count_min_uic=0; // mhdenizei giati tha kanei post apo to temp-hum kai etsi sto backend tha
parei last timestamp
uic_send=0;
}

```

```
////////////////////////////////////// IN TEMP_HUM
```

```
void HAL_TIM_PeriodElapsedCallback(TIM_HandleTypeDef *htim) // timers
{
    if(htim->Instance == TIM4) // gia metrshh me to DHT22 kathe 5 lepta otan leitourgei o kausthras
    {
        tim4_sec++; // every second
    }

    if (tim4_sec == 60)
    {
        tim4_sec=0;
        tim4_min++; // every minute
    }

    if(htim->Instance == TIM3) // main counter gia Distance kai UIC
    {
        count_sec++; // every second
    }

    if (count_sec == 60)
    {
        count_sec=0;
        count_min++; // every minute gia distance
        count_min_uic++; // every minute gia uic
        tim5_min++;
    }

    if(count_min == 360) // 6*60 -> 360 gia 6 wres
    {
```

```

        time_to_read = 1;
        count_min=0;
        count_min_uic=0; // mhdenizei giati tha kanei post apo to distance kai etsi sto backend tha
parei last timestamp
        uic_send = 0;
    }

    if(count_min_uic == 5) // gia dummy json gia uic kathe 5 lepta
    {
        uic_send = 1; // set f "1"
        count_min_uic=0;
    }
}

/* USER CODE END PV */

/* Private function prototypes -----*/
void SystemClock_Config(void);
static void MX_GPIO_Init(void);
static void MX_DMA_Init(void);
static void MX_SPI1_Init(void);
static void MX_SPI2_Init(void);
static void MX_TIM2_Init(void);
static void MX_TIM3_Init(void);
static void MX_TIM4_Init(void);
static void MX_USART3_UART_Init(void);
/* USER CODE BEGIN PFP */
//void mainApp(void);
/* USER CODE END PFP */

/* Private user code -----*/
/* USER CODE BEGIN 0 */

```

```

/* USER CODE END 0 */

/**
 * @brief The application entry point.
 * @retval int
 */
int main(void)
{
    /* USER CODE BEGIN 1 */

    /* USER CODE END 1 */

    /* MCU Configuration-----*/

    /* Reset of all peripherals, Initializes the Flash interface and the Systick. */
    HAL_Init();

    /* USER CODE BEGIN Init */

    /* USER CODE END Init */

    /* Configure the system clock */
    SystemClock_Config();

    /* USER CODE BEGIN SysInit */

    /* USER CODE END SysInit */

    /* Initialize all configured peripherals */
    MX_GPIO_Init();
    MX_DMA_Init();

```

```

MX_SPI1_Init();
MX_SPI2_Init();
MX_TIM2_Init();
MX_TIM3_Init();
MX_TIM4_Init();
MX_USART3_UART_Init();
/* USER CODE BEGIN 2 */

HAL_TIM_IC_Start_IT(&htim2, TIM_CHANNEL_2);
HAL_TIM_Base_Start_IT(&htim3);
//HAL_TIM_Base_Start_IT(&htim4);

/* USER CODE END 2 */

/* Infinite loop */
/* USER CODE BEGIN WHILE */

void numbers (void)
{
    BSP_LCD_SetTextColor(LCD_COLOR_BLUE);
    BSP_LCD_DrawRect(9, 147, 22, 28); // 0
    BSP_LCD_DrawRect(40, 147, 22, 28); // 1
    BSP_LCD_DrawRect(71, 147, 22, 28); // 2
    BSP_LCD_DrawRect(102, 147, 22, 28); // 3
    BSP_LCD_DrawRect(133, 147, 22, 28); // 4
    BSP_LCD_DrawRect(164, 147, 22, 28); // 5
    BSP_LCD_DrawRect(195, 147, 22, 28); // 6
    BSP_LCD_DrawRect(226, 147, 22, 28); // 7
    BSP_LCD_DrawRect(257, 147, 22, 28); // 8
    BSP_LCD_DrawRect(288, 147, 22, 28); // 9
    BSP_LCD_SetTextColor(LCD_COLOR_RED);

```

```

BSP_LCD_DrawRect(86, 191, 54, 28);// DEL
BSP_LCD_SetTextColor(LCD_COLOR_GREEN);
BSP_LCD_DrawRect(197, 191, 38, 28);// OK

BSP_LCD_SetBackColor(LCD_COLOR_WHITE);
BSP_LCD_SetTextColor(LCD_COLOR_BLUE);
BSP_LCD_DisplayStringAt(12,150, (uint8_t *)"0", LEFT_MODE);
BSP_LCD_DisplayStringAt(43,150, (uint8_t *)"1", LEFT_MODE);
BSP_LCD_DisplayStringAt(74,150, (uint8_t *)"2", LEFT_MODE);
BSP_LCD_DisplayStringAt(105,150, (uint8_t *)"3", LEFT_MODE);
BSP_LCD_DisplayStringAt(136,150, (uint8_t *)"4", LEFT_MODE);
BSP_LCD_DisplayStringAt(167,150, (uint8_t *)"5", LEFT_MODE);
BSP_LCD_DisplayStringAt(198,150, (uint8_t *)"6", LEFT_MODE);
BSP_LCD_DisplayStringAt(229,150, (uint8_t *)"7", LEFT_MODE);
BSP_LCD_DisplayStringAt(260,150, (uint8_t *)"8", LEFT_MODE);
BSP_LCD_DisplayStringAt(291,150, (uint8_t *)"9", LEFT_MODE);
BSP_LCD_SetTextColor(LCD_COLOR_RED);
BSP_LCD_DisplayStringAt(87,193, (uint8_t *)"DEL", LEFT_MODE);
BSP_LCD_SetTextColor(LCD_COLOR_GREEN);
BSP_LCD_DisplayStringAt(198,193, (uint8_t *)"OK", LEFT_MODE);
BSP_LCD_SetBackColor(LCD_COLOR_WHITE);
BSP_LCD_SetTextColor(LCD_COLOR_BLUE);
}

```

```

void display_setpass (void)

```

```

{
    if(start == 0)
    {
        BSP_LCD_SetTextColor(LCD_COLOR_BLACK);
        BSP_LCD_DisplayStringAt(10, 10, (uint8_t *)"SET PASSWORD", LEFT_MODE);
        change_pass = 1;
    }
}

```

```

else if (start == 1)
{
    //BSP_LCD_SetBackColor(LCD_COLOR_WHITE);
    BSP_LCD_SetTextColor(LCD_COLOR_BLACK);
    BSP_LCD_DisplayStringAt(10, 10, (uint8_t *)"PASSWORD", LEFT_MODE);
}
else if ((start == 3 && change_pass == 1) || temp_st==2)
{
    BSP_LCD_SetBackColor(LCD_COLOR_RED);
    BSP_LCD_SetTextColor(LCD_COLOR_WHITE);
    BSP_LCD_DisplayStringAt(301, 1, (uint8_t *)"X", LEFT_MODE);
    BSP_LCD_SetBackColor(LCD_COLOR_WHITE);
    BSP_LCD_SetTextColor(LCD_COLOR_BLACK);
    BSP_LCD_DisplayStringAt(10, 10, (uint8_t *)"SET PASSWORD", LEFT_MODE);
}

    BSP_LCD_SetFont(&Font16);
    BSP_LCD_DisplayStringAt(10, 33, (uint8_t *)"4-Digit", LEFT_MODE);
    BSP_LCD_SetFont(&Font24);
    BSP_LCD_DrawRect(100, 71, 111, 41); // plasio kwdikou
    BSP_LCD_SetTextColor(LCD_COLOR_BLUE);
}

void display_apodosh (void)
{
    if(start == 0)
    {
        BSP_LCD_SetTextColor(LCD_COLOR_BLACK);
        BSP_LCD_DisplayStringAt(10, 10, (uint8_t *)"Vathmos apodosis", LEFT_MODE);
        set_apod = 1;
    }
}

```



```

else if ((start == 2 && set_apod == 1) || temp_st==2)
{
    BSP_LCD_SetBackColor(LCD_COLOR_RED);
    BSP_LCD_SetTextColor(LCD_COLOR_WHITE);
    BSP_LCD_DisplayStringAt(301, 1, (uint8_t *)"X", LEFT_MODE);
    BSP_LCD_SetBackColor(LCD_COLOR_WHITE);
    BSP_LCD_SetTextColor(LCD_COLOR_BLACK);
    BSP_LCD_DisplayStringAt(10, 5, (uint8_t *)"Vathmos", LEFT_MODE);
    BSP_LCD_DisplayStringAt(10, 25, (uint8_t *)"Apodosis", LEFT_MODE);
}

BSP_LCD_DrawRect(100, 71, 111, 41);// plaisio kwdikou
BSP_LCD_SetTextColor(LCD_COLOR_BLUE);
BSP_LCD_SetFont(&Font24);
}

void set_password(void)
{
    for (int i=0; i<4; i++)
    {
        temp_pass[i] = set_pass[i];
    }

    char s[5];
    do
    {
        numbers();
        display_setpass();
    }
    BSP_TS_GetState(&ts);
}

```

```

if(ts.TouchDetected)
{
    if(ts.Y >= 1 && ts.Y < 20 && ts.X >= 290 && ts.X <= 320)
    {
        change_pass = 0;
        for (int i=0; i<4; i++)
        {
            set_pass[i]=temp_pass[i];
        }
    }

    else if(ts.Y >= 145 && ts.Y < 175 && sp < 4 )
    {

        if (ts.X >= 10 && ts.X < 35) // 0
        {
            BSP_LCD_SetBackColor(LCD_COLOR_BLUE);
            BSP_LCD_FillRect(9, 147, 22, 28);
            BSP_LCD_SetTextColor(LCD_COLOR_WHITE);
            BSP_LCD_DisplayStringAt(12,150, (uint8_t *)"0", LEFT_MODE);
            set_pass[sp]=0;
            sp++;
            set_cursor_x= set_cursor_x + 25;
            sprintf(s, "%d", 0);
            BSP_LCD_DisplayStringAt(set_cursor_x, set_cursor_y, (uint8_t *)s,
LEFT_MODE);
        }

        else if (ts.X >= 45 && ts.X < 75) // 1
        {
            BSP_LCD_SetBackColor(LCD_COLOR_BLUE);

```

```

        BSP_LCD_FillRect(40, 147, 22, 28);
        BSP_LCD_SetTextColor(LCD_COLOR_WHITE);
        BSP_LCD_DisplayStringAt(43,150, (uint8_t *)"1", LEFT_MODE);
        set_pass[sp]=1;
        sp++;
        set_cursor_x= set_cursor_x + 25;
        sprintf(s, "%d", 1);
        BSP_LCD_DisplayStringAt(set_cursor_x, set_cursor_y, (uint8_t *)s,
LEFT_MODE);
    }

else if (ts.X >= 80 && ts.X < 104) // 2
{
    BSP_LCD_SetBackColor(LCD_COLOR_BLUE);
    BSP_LCD_FillRect(71, 147, 22, 28);
    BSP_LCD_SetTextColor(LCD_COLOR_WHITE);
    BSP_LCD_DisplayStringAt(74,150, (uint8_t *)"2", LEFT_MODE);
    set_pass[sp]=2;
    sp++;
    set_cursor_x= set_cursor_x + 25;
    sprintf(s, "%d", 2);
    BSP_LCD_DisplayStringAt(set_cursor_x, set_cursor_y, (uint8_t *)s,
LEFT_MODE);
}

else if (ts.X >= 111 && ts.X < 135) // 3
{
    BSP_LCD_SetBackColor(LCD_COLOR_BLUE);
    BSP_LCD_FillRect(102, 147, 22, 28);
    BSP_LCD_SetTextColor(LCD_COLOR_WHITE);
    BSP_LCD_DisplayStringAt(105,150, (uint8_t *)"3", LEFT_MODE);
    set_pass[sp]=3;
    sp++;

```

```

        set_cursor_x= set_cursor_x + 25;
        sprintf(s, "%d", 3);
        BSP_LCD_DisplayStringAt(set_cursor_x, set_cursor_y, (uint8_t *)s,
LEFT_MODE);
    }

else if (ts.X >= 150 && ts.X < 170) // 4
{
    BSP_LCD_SetBackColor(LCD_COLOR_BLUE);
    BSP_LCD_FillRect(133, 147, 22, 28);
    BSP_LCD_SetTextColor(LCD_COLOR_WHITE);
    BSP_LCD_DisplayStringAt(136,150, (uint8_t *)"4", LEFT_MODE);
    set_pass[sp]=4;
    sp++;
    set_cursor_x= set_cursor_x + 25;
    sprintf(s, "%d", 4);
    BSP_LCD_DisplayStringAt(set_cursor_x, set_cursor_y, (uint8_t *)s,
LEFT_MODE);
}

else if (ts.X >= 177 && ts.X < 205) // 5
{
    BSP_LCD_SetBackColor(LCD_COLOR_BLUE);
    BSP_LCD_FillRect(164, 147, 22, 28);
    BSP_LCD_SetTextColor(LCD_COLOR_WHITE);
    BSP_LCD_DisplayStringAt(167,150, (uint8_t *)"5", LEFT_MODE);
    set_pass[sp]=5;
    sp++;
    set_cursor_x= set_cursor_x + 25;
    sprintf(s, "%d", 5);
    BSP_LCD_DisplayStringAt(set_cursor_x, set_cursor_y, (uint8_t *)s,
LEFT_MODE);
}

```

```

else if (ts.X >= 210 && ts.X < 240) // 6
{
    BSP_LCD_SetBackColor(LCD_COLOR_BLUE);
    BSP_LCD_FillRect(195, 147, 22, 28);
    BSP_LCD_SetTextColor(LCD_COLOR_WHITE);
    BSP_LCD_DisplayStringAt(198,150, (uint8_t *)"6", LEFT_MODE);
    set_pass[sp]=6;
    sp++;
    set_cursor_x= set_cursor_x + 25;
    sprintf(s, "%d", 6);
    BSP_LCD_DisplayStringAt(set_cursor_x, set_cursor_y, (uint8_t *)s,
LEFT_MODE);
}

```

```

else if (ts.X >= 245 && ts.X < 275) // 7
{
    BSP_LCD_SetBackColor(LCD_COLOR_BLUE);
    BSP_LCD_FillRect(226, 147, 22, 28);
    BSP_LCD_SetTextColor(LCD_COLOR_WHITE);
    BSP_LCD_DisplayStringAt(229,150, (uint8_t *)"7", LEFT_MODE);
    set_pass[sp]=7;
    sp++;
    set_cursor_x= set_cursor_x + 25;
    sprintf(s, "%d", 7);
    BSP_LCD_DisplayStringAt(set_cursor_x, set_cursor_y, (uint8_t *)s,
LEFT_MODE);
}

```

```

else if (ts.X >= 280 && ts.X < 305) // 8
{
    BSP_LCD_SetBackColor(LCD_COLOR_BLUE);
    BSP_LCD_FillRect(257, 147, 22, 28);

```

```

        BSP_LCD_SetTextColor(LCD_COLOR_WHITE);
        BSP_LCD_DisplayStringAt(260,150, (uint8_t *)"8", LEFT_MODE);
        set_pass[sp]=8;
        sp++;
        set_cursor_x= set_cursor_x + 25;
        sprintf(s, "%d", 8);
        BSP_LCD_DisplayStringAt(set_cursor_x, set_cursor_y, (uint8_t *)s,
LEFT_MODE);
    }

    else if (ts.X >= 310 && ts.X < 340) // 9
    {
        BSP_LCD_SetBackColor(LCD_COLOR_BLUE);
        BSP_LCD_FillRect(288, 147, 22, 28);
        BSP_LCD_SetTextColor(LCD_COLOR_WHITE);
        BSP_LCD_DisplayStringAt(291,150, (uint8_t *)"9", LEFT_MODE);
        set_pass[sp]=9;
        sp++;
        set_cursor_x= set_cursor_x + 25;
        sprintf(s, "%d", 9);
        BSP_LCD_DisplayStringAt(set_cursor_x, set_cursor_y, (uint8_t *)s,
LEFT_MODE);
    }

    else
    {
        numbers();
    }

    HAL_Delay(300);

    BSP_LCD_SetTextColor(LCD_COLOR_WHITE);
    BSP_LCD_FillRect(9, 147, 22, 28);

```

```

BSP_LCD_FillRect(40, 147, 22, 28); // 1
BSP_LCD_FillRect(71, 147, 22, 28); // 2
BSP_LCD_FillRect(102, 147, 22, 28); // 3
BSP_LCD_FillRect(133, 147, 22, 28); // 4
BSP_LCD_FillRect(164, 147, 22, 28); // 5
BSP_LCD_FillRect(195, 147, 22, 28); // 6
BSP_LCD_FillRect(226, 147, 22, 28); // 7
BSP_LCD_FillRect(257, 147, 22, 28); // 8
BSP_LCD_FillRect(288, 147, 22, 28); // 9
BSP_LCD_FillRect(86, 191, 54, 28); // DEL
BSP_LCD_FillRect(197, 193, 38, 28); // OK

}

else if(ts.Y >= 185 && ts.Y < 220 && sp <= 4 ) // <= gia na exw epilogh del h ok
{
    if (ts.X >= 100 && ts.X < 160 && sp > 0) // DEL
    {
        BSP_LCD_SetBackColor(LCD_COLOR_RED);
        BSP_LCD_SetTextColor(LCD_COLOR_RED);
        BSP_LCD_DrawRect(86, 191, 54, 28);
        BSP_LCD_SetTextColor(LCD_COLOR_WHITE);
        BSP_LCD_DisplayStringAt(88,193, (uint8_t *)"DEL",
LEFT_MODE);

        sp--;
        set_pass[sp]=0;

        sprintf(s, " ");
        BSP_LCD_SetBackColor(LCD_COLOR_WHITE);
        BSP_LCD_DisplayStringAt(set_cursor_x, set_cursor_y, (uint8_t *)s,
LEFT_MODE);

        set_cursor_x= set_cursor_x - 25;
    }
}

```

```

else if (ts.X >= 210 && ts.X < 260 && sp == 4) // OK
{
    BSP_LCD_SetBackColor(LCD_COLOR_GREEN);
    BSP_LCD_SetTextColor(LCD_COLOR_GREEN);
    BSP_LCD_FillRect(197, 191, 37, 28); // OK
    BSP_LCD_SetTextColor(LCD_COLOR_WHITE);
    BSP_LCD_DisplayStringAt(198,193, (uint8_t *)"OK",
LEFT_MODE);

    spok=1;
    sp++;
    ok=1;

}

HAL_Delay(300);

BSP_LCD_SetTextColor(LCD_COLOR_WHITE);
BSP_LCD_FillRect(9, 147, 22, 28); // 0
BSP_LCD_FillRect(40, 147, 22, 28); // 1
BSP_LCD_FillRect(71, 147, 22, 28); // 2
BSP_LCD_FillRect(102, 147, 22, 28); // 3
BSP_LCD_FillRect(133, 147, 22, 28); // 4
BSP_LCD_FillRect(164, 147, 22, 28); // 5
BSP_LCD_FillRect(195, 147, 22, 28); // 6
BSP_LCD_FillRect(226, 147, 22, 28); // 7
BSP_LCD_FillRect(257, 147, 22, 28); // 8
BSP_LCD_FillRect(288, 147, 22, 28); // 9
BSP_LCD_FillRect(86, 191, 54, 28); // DEL
BSP_LCD_FillRect(197, 193, 38, 28); // OK
}

else if (sp >=4 && spok == 0)

```



```

    {
        numbers();
    }

}

}

while((sp<5 && spok==0) && change_pass == 1);
spok=0;
sp=0;
set_cursor_x = 85;
BSP_LCD_Clear(LCD_COLOR_WHITE);
if (start == 2)
{
    start=2;

}

else {start=1;}

}

```

```

void password (void)
{
    TS_StateTypeDef ts;

    char s[5];

```

```

do
{

    numbers();
    display_setpass();
    BSP_TS_GetState(&ts);
    if(ts.TouchDetected)
    {
        if(ts.Y >= 145 && ts.Y < 175 && sp < 4 )
        {

            if (ts.X >= 10 && ts.X < 35) // 0
            {
                BSP_LCD_SetBackColor(LCD_COLOR_BLUE);
                BSP_LCD_FillRect(9, 147, 22, 28);
                BSP_LCD_SetTextColor(LCD_COLOR_WHITE);
                BSP_LCD_DisplayStringAt(12,150, (uint8_t *)"0", LEFT_MODE);
                check_pass[sp]=0;
                sp++;
                set_cursor_x= set_cursor_x + 25;
                BSP_LCD_DisplayStringAt(set_cursor_x, set_cursor_y, (uint8_t
*)"*", LEFT_MODE);
            }

            else if (ts.X >= 45 && ts.X < 75) // 1
            {
                BSP_LCD_SetBackColor(LCD_COLOR_BLUE);
                BSP_LCD_FillRect(40, 147, 22, 28);
                BSP_LCD_SetTextColor(LCD_COLOR_WHITE);
                BSP_LCD_DisplayStringAt(43,150, (uint8_t *)"1", LEFT_MODE);
                check_pass[sp]=1;
                sp++;
            }
        }
    }
}

```

```

        set_cursor_x= set_cursor_x + 25;
        BSP_LCD_DisplayStringAt(set_cursor_x, set_cursor_y, (uint8_t
*)"", LEFT_MODE);
    }

    else if (ts.X >= 80 && ts.X < 104) // 2
    {
        BSP_LCD_SetBackColor(LCD_COLOR_BLUE);
        BSP_LCD_FillRect(71, 147, 22, 28);
        BSP_LCD_SetTextColor(LCD_COLOR_WHITE);
        BSP_LCD_DisplayStringAt(74,150, (uint8_t *)"2", LEFT_MODE);
        check_pass[sp]=2;
        sp++;
        set_cursor_x= set_cursor_x + 25;
        BSP_LCD_DisplayStringAt(set_cursor_x, set_cursor_y, (uint8_t
*)"", LEFT_MODE);
    }

    else if (ts.X >= 111 && ts.X < 135) // 3
    {
        BSP_LCD_SetBackColor(LCD_COLOR_BLUE);
        BSP_LCD_FillRect(102, 147, 22, 28);
        BSP_LCD_SetTextColor(LCD_COLOR_WHITE);
        BSP_LCD_DisplayStringAt(105,150, (uint8_t *)"3", LEFT_MODE);
        check_pass[sp]=3;
        sp++;
        set_cursor_x= set_cursor_x + 25;
        BSP_LCD_DisplayStringAt(set_cursor_x, set_cursor_y, (uint8_t
*)"", LEFT_MODE);
    }

    else if (ts.X >= 150 && ts.X < 170) // 4
    {

```

```

        BSP_LCD_SetBackColor(LCD_COLOR_BLUE);
        BSP_LCD_FillRect(133, 147, 22, 28);
        BSP_LCD_SetTextColor(LCD_COLOR_WHITE);
        BSP_LCD_DisplayStringAt(136,150, (uint8_t *)"4", LEFT_MODE);
        check_pass[sp]=4;
        sp++;
        set_cursor_x= set_cursor_x + 25;
        BSP_LCD_DisplayStringAt(set_cursor_x, set_cursor_y, (uint8_t
*)"*", LEFT_MODE);
    }

else if (ts.X >= 177 && ts.X < 205) // 5
{
    BSP_LCD_SetBackColor(LCD_COLOR_BLUE);
    BSP_LCD_FillRect(164, 147, 22, 28);
    BSP_LCD_SetTextColor(LCD_COLOR_WHITE);
    BSP_LCD_DisplayStringAt(167,150, (uint8_t *)"5", LEFT_MODE);
    check_pass[sp]=5;
    sp++;
    set_cursor_x= set_cursor_x + 25;
    BSP_LCD_DisplayStringAt(set_cursor_x, set_cursor_y, (uint8_t
*)"*", LEFT_MODE);
}

else if (ts.X >= 210 && ts.X < 240) // 6
{
    BSP_LCD_SetBackColor(LCD_COLOR_BLUE);
    BSP_LCD_FillRect(195, 147, 22, 28);
    BSP_LCD_SetTextColor(LCD_COLOR_WHITE);
    BSP_LCD_DisplayStringAt(198,150, (uint8_t *)"6", LEFT_MODE);
    check_pass[sp]=6;
    sp++;
    set_cursor_x= set_cursor_x + 25;

```

```

        BSP_LCD_DisplayStringAt(set_cursor_x, set_cursor_y, (uint8_t
*)"", LEFT_MODE);
    }

    else if (ts.X >= 245 && ts.X < 275) // 7
    {
        BSP_LCD_SetBackColor(LCD_COLOR_BLUE);
        BSP_LCD_FillRect(226, 147, 22, 28);
        BSP_LCD_SetTextColor(LCD_COLOR_WHITE);
        BSP_LCD_DisplayStringAt(229,150, (uint8_t *)"7", LEFT_MODE);
        check_pass[sp]=7;
        sp++;
        set_cursor_x= set_cursor_x + 25;
        BSP_LCD_DisplayStringAt(set_cursor_x, set_cursor_y, (uint8_t
*)"", LEFT_MODE);
    }

    else if (ts.X >= 280 && ts.X < 305) // 8
    {
        BSP_LCD_SetBackColor(LCD_COLOR_BLUE);
        BSP_LCD_FillRect(257, 147, 22, 28);
        BSP_LCD_SetTextColor(LCD_COLOR_WHITE);
        BSP_LCD_DisplayStringAt(260,150, (uint8_t *)"8", LEFT_MODE);
        check_pass[sp]=8;
        sp++;
        set_cursor_x= set_cursor_x + 25;
        BSP_LCD_DisplayStringAt(set_cursor_x, set_cursor_y, (uint8_t
*)"", LEFT_MODE);
    }

    else if (ts.X >= 310 && ts.X < 340) // 9
    {
        BSP_LCD_SetBackColor(LCD_COLOR_BLUE);

```

```

        BSP_LCD_FillRect(288, 147, 22, 28);
        BSP_LCD_SetTextColor(LCD_COLOR_WHITE);
        BSP_LCD_DisplayStringAt(291,150, (uint8_t *)"9", LEFT_MODE);
        check_pass[sp]=9;
        sp++;
        set_cursor_x= set_cursor_x + 25;
        BSP_LCD_DisplayStringAt(set_cursor_x, set_cursor_y, (uint8_t
*)"*", LEFT_MODE);
    }

//     else
//     {
//         numbers();
//     }

    HAL_Delay(300);

    BSP_LCD_SetTextColor(LCD_COLOR_WHITE);
    BSP_LCD_FillRect(9, 147, 22, 28);
    BSP_LCD_FillRect(40, 147, 22, 28);// 1
    BSP_LCD_FillRect(71, 147, 22, 28);// 2
    BSP_LCD_FillRect(102, 147, 22, 28);// 3
    BSP_LCD_FillRect(133, 147, 22, 28);// 4
    BSP_LCD_FillRect(164, 147, 22, 28);// 5
    BSP_LCD_FillRect(195, 147, 22, 28);// 6
    BSP_LCD_FillRect(226, 147, 22, 28);// 7
    BSP_LCD_FillRect(257, 147, 22, 28);// 8
    BSP_LCD_FillRect(288, 147, 22, 28);// 9
    BSP_LCD_FillRect(86, 191, 54, 28);// DEL
    BSP_LCD_FillRect(197, 193, 38, 28);// OK

}

```

```

else if(ts.Y >= 185 && ts.Y < 220 && sp <= 4 ) // <= gia na exw epilogh del h ok
{
    if (ts.X >= 100 && ts.X < 160 && sp > 0) // DEL
    {
        BSP_LCD_SetBackColor(LCD_COLOR_RED);
        BSP_LCD_SetTextColor(LCD_COLOR_RED);
        BSP_LCD_DrawRect(86, 191, 54, 28);
        BSP_LCD_SetTextColor(LCD_COLOR_WHITE);
        BSP_LCD_DisplayStringAt(88,193,      (uint8_t      *)"DEL",
LEFT_MODE);

        sp--;
        check_pass[sp]=0;

        sprintf(s, " ");
        BSP_LCD_SetBackColor(LCD_COLOR_WHITE);
        BSP_LCD_DisplayStringAt(set_cursor_x, set_cursor_y, (uint8_t *)s,
LEFT_MODE);

        set_cursor_x= set_cursor_x - 25;
    }

    else if (ts.X >= 210 && ts.X < 260 && sp == 4) // OK
    {
        BSP_LCD_SetBackColor(LCD_COLOR_GREEN);
        BSP_LCD_SetTextColor(LCD_COLOR_GREEN);
        BSP_LCD_FillRect(197, 191, 37, 28);// OK
        BSP_LCD_SetTextColor(LCD_COLOR_WHITE);
        BSP_LCD_DisplayStringAt(198,193,      (uint8_t      *)"OK",
LEFT_MODE);

        spok=1;
        sp++;
    }

    //      else

```

```

//          {
//              numbers();
//          }

HAL_Delay(300);

BSP_LCD_SetTextColor(LCD_COLOR_WHITE);
BSP_LCD_FillRect(9, 147, 22, 28); // 0
BSP_LCD_FillRect(40, 147, 22, 28); // 1
BSP_LCD_FillRect(71, 147, 22, 28); // 2
BSP_LCD_FillRect(102, 147, 22, 28); // 3
BSP_LCD_FillRect(133, 147, 22, 28); // 4
BSP_LCD_FillRect(164, 147, 22, 28); // 5
BSP_LCD_FillRect(195, 147, 22, 28); // 6
BSP_LCD_FillRect(226, 147, 22, 28); // 7
BSP_LCD_FillRect(257, 147, 22, 28); // 8
BSP_LCD_FillRect(288, 147, 22, 28); // 9
BSP_LCD_FillRect(86, 191, 54, 28); // DEL
BSP_LCD_FillRect(197, 193, 38, 28); // OK
    }

//          else if (sp >=4 && spok == 0)
//          {
//              numbers();
//          }

    }
}

while(sp<5 && spok==0);

for (int i=0; i<4; i++)

```



```

{
    if (check_pass[i] != set_pass[i])
    {
        g=1;
    }
}

if (g == 0)
{
    //correct
    BSP_LCD_Clear(LCD_COLOR_WHITE);
        BSP_LCD_SetTextColor(LCD_COLOR_BLACK);
        BSP_LCD_DisplayStringAt(15, 90, (uint8_t *)"CORRECT PASSWORD!",
LEFT_MODE);
        set_cursor_x = 85;
        start=3;    // index
        ok=2;
        spok=0;
        sp=0;
        HAL_Delay(1500);
        BSP_LCD_Clear(LCD_COLOR_WHITE);
}

else
{
    //wrong pass
    spok=0;
    sp=0;
    g=0;
    start=1;
    set_cursor_x = 85;
    BSP_LCD_Clear(LCD_COLOR_WHITE);
}

```

```

    BSP_LCD_SetTextColor(LCD_COLOR_BLACK);
    BSP_LCD_SetBackColor(LCD_COLOR_M_RED);
    BSP_LCD_DisplayStringAt(30, 90, (uint8_t *)"WRONG PASSWORD!", LEFT_MODE);
    HAL_Delay(1500);
    //BSP_LCD_Clear(LCD_COLOR_WHITE);

}

}

```

```

void set_apodosh(void)

```

```

{
    for (int i=0; i<3; i++)
    {
        temp_apod_ar1[i]=set_apod_ar[i];
        set_apod_ar[i]=0;
    }

```

```

    char s[5];

```

```

do

```

```

{
    numbers();
    display_apodosh();
    BSP_TS_GetState(&ts);

```

```

    if(ts.TouchDetected)

```

```

    {
        if(ts.Y >= 1 && ts.Y < 20 && ts.X >= 290 && ts.X <= 320) // gia X fortwnei

```

```

oti eixe

```

```

    {
        set_apod = 0;
        for (int i=0; i<3; i++)
        {
            set_apod_ar[i]=temp_apod_ar1[i];
        }
    }

else if(ts.Y >= 145 && ts.Y < 175 && sp < 3 )
{

if (ts.X >= 10 && ts.X < 35) // 0
{
    BSP_LCD_SetBackColor(LCD_COLOR_BLUE);
    BSP_LCD_FillRect(9, 147, 22, 28);
    BSP_LCD_SetTextColor(LCD_COLOR_WHITE);
    BSP_LCD_DisplayStringAt(12,150, (uint8_t *)"0", LEFT_MODE);
    set_apod_ar[sp]=0;
    sp++;
    set_cursor_x= set_cursor_x + 25;
    sprintf(s, "%d", 0);
    BSP_LCD_DisplayStringAt(set_cursor_x, set_cursor_y, (uint8_t *)s,
LEFT_MODE);
}

else if (ts.X >= 45 && ts.X < 75) // 1
{
    BSP_LCD_SetBackColor(LCD_COLOR_BLUE);
    BSP_LCD_FillRect(40, 147, 22, 28);
    BSP_LCD_SetTextColor(LCD_COLOR_WHITE);
    BSP_LCD_DisplayStringAt(43,150, (uint8_t *)"1", LEFT_MODE);
    set_apod_ar[sp]=1;

```

```

        sp++;
        set_cursor_x= set_cursor_x + 25;
        sprintf(s, "%d", 1);
        BSP_LCD_DisplayStringAt(set_cursor_x, set_cursor_y, (uint8_t *)s,
LEFT_MODE);
    }

else if (ts.X >= 80 && ts.X < 104) // 2
{
    BSP_LCD_SetBackColor(LCD_COLOR_BLUE);
    BSP_LCD_FillRect(71, 147, 22, 28);
    BSP_LCD_SetTextColor(LCD_COLOR_WHITE);
    BSP_LCD_DisplayStringAt(74,150, (uint8_t *)"2", LEFT_MODE);
    set_apod_ar[sp]=2;
    sp++;
    set_cursor_x= set_cursor_x + 25;
    sprintf(s, "%d", 2);
    BSP_LCD_DisplayStringAt(set_cursor_x, set_cursor_y, (uint8_t *)s,
LEFT_MODE);
}

else if (ts.X >= 111 && ts.X < 135) // 3
{
    BSP_LCD_SetBackColor(LCD_COLOR_BLUE);
    BSP_LCD_FillRect(102, 147, 22, 28);
    BSP_LCD_SetTextColor(LCD_COLOR_WHITE);
    BSP_LCD_DisplayStringAt(105,150, (uint8_t *)"3", LEFT_MODE);
    set_apod_ar[sp]=3;
    sp++;
    set_cursor_x= set_cursor_x + 25;
    sprintf(s, "%d", 3);
    BSP_LCD_DisplayStringAt(set_cursor_x, set_cursor_y, (uint8_t *)s,
LEFT_MODE);
}

```

```

    }

else if (ts.X >= 150 && ts.X < 170) // 4
{
    BSP_LCD_SetBackColor(LCD_COLOR_BLUE);
    BSP_LCD_FillRect(133, 147, 22, 28);
    BSP_LCD_SetTextColor(LCD_COLOR_WHITE);
    BSP_LCD_DisplayStringAt(136,150, (uint8_t *)"4", LEFT_MODE);
    set_apod_ar[sp]=4;
    sp++;
    set_cursor_x= set_cursor_x + 25;
    sprintf(s, "%d", 4);
    BSP_LCD_DisplayStringAt(set_cursor_x, set_cursor_y, (uint8_t *)s,
LEFT_MODE);
}

else if (ts.X >= 177 && ts.X < 205) // 5
{
    BSP_LCD_SetBackColor(LCD_COLOR_BLUE);
    BSP_LCD_FillRect(164, 147, 22, 28);
    BSP_LCD_SetTextColor(LCD_COLOR_WHITE);
    BSP_LCD_DisplayStringAt(167,150, (uint8_t *)"5", LEFT_MODE);
    set_apod_ar[sp]=5;
    sp++;
    set_cursor_x= set_cursor_x + 25;
    sprintf(s, "%d", 5);
    BSP_LCD_DisplayStringAt(set_cursor_x, set_cursor_y, (uint8_t *)s,
LEFT_MODE);
}

else if (ts.X >= 210 && ts.X < 240) // 6
{
    BSP_LCD_SetBackColor(LCD_COLOR_BLUE);

```

```

        BSP_LCD_FillRect(195, 147, 22, 28);
        BSP_LCD_SetTextColor(LCD_COLOR_WHITE);
        BSP_LCD_DisplayStringAt(198,150, (uint8_t *)"6", LEFT_MODE);
        set_apod_ar[sp]=6;
        sp++;
        set_cursor_x= set_cursor_x + 25;
        sprintf(s, "%d", 6);
        BSP_LCD_DisplayStringAt(set_cursor_x, set_cursor_y, (uint8_t *)s,
LEFT_MODE);
    }

else if (ts.X >= 245 && ts.X < 275) // 7
{
    BSP_LCD_SetBackColor(LCD_COLOR_BLUE);
    BSP_LCD_FillRect(226, 147, 22, 28);
    BSP_LCD_SetTextColor(LCD_COLOR_WHITE);
    BSP_LCD_DisplayStringAt(229,150, (uint8_t *)"7", LEFT_MODE);
    set_apod_ar[sp]=7;
    sp++;
    set_cursor_x= set_cursor_x + 25;
    sprintf(s, "%d", 7);
    BSP_LCD_DisplayStringAt(set_cursor_x, set_cursor_y, (uint8_t *)s,
LEFT_MODE);
}

else if (ts.X >= 280 && ts.X < 305) // 8
{
    BSP_LCD_SetBackColor(LCD_COLOR_BLUE);
    BSP_LCD_FillRect(257, 147, 22, 28);
    BSP_LCD_SetTextColor(LCD_COLOR_WHITE);
    BSP_LCD_DisplayStringAt(260,150, (uint8_t *)"8", LEFT_MODE);
    set_apod_ar[sp]=8;
    sp++;

```

```

        set_cursor_x= set_cursor_x + 25;
        sprintf(s, "%d", 8);
        BSP_LCD_DisplayStringAt(set_cursor_x, set_cursor_y, (uint8_t *)s,
LEFT_MODE);
    }

else if (ts.X >= 310 && ts.X < 340) // 9
{
    BSP_LCD_SetBackColor(LCD_COLOR_BLUE);
    BSP_LCD_FillRect(288, 147, 22, 28);
    BSP_LCD_SetTextColor(LCD_COLOR_WHITE);
    BSP_LCD_DisplayStringAt(291,150, (uint8_t *)"9", LEFT_MODE);
    set_apod_ar[sp]=9;
    sp++;
    set_cursor_x= set_cursor_x + 25;
    sprintf(s, "%d", 9);
    BSP_LCD_DisplayStringAt(set_cursor_x, set_cursor_y, (uint8_t *)s,
LEFT_MODE);
}

else
{
    numbers();
}

HAL_Delay(300);

BSP_LCD_SetTextColor(LCD_COLOR_WHITE);
BSP_LCD_FillRect(9, 147, 22, 28);
BSP_LCD_FillRect(40, 147, 22, 28);// 1
BSP_LCD_FillRect(71, 147, 22, 28); // 2
BSP_LCD_FillRect(102, 147, 22, 28);// 3
BSP_LCD_FillRect(133, 147, 22, 28); // 4

```

```

        BSP_LCD_FillRect(164, 147, 22, 28);// 5
        BSP_LCD_FillRect(195, 147, 22, 28); // 6
        BSP_LCD_FillRect(226, 147, 22, 28);// 7
        BSP_LCD_FillRect(257, 147, 22, 28); // 8
        BSP_LCD_FillRect(288, 147, 22, 28);// 9
        BSP_LCD_FillRect(86, 191, 54, 28);// DEL
        BSP_LCD_FillRect(197, 193, 38, 28);// OK
    }

else if(ts.Y >= 185 && ts.Y < 220 && sp <= 3 ) // <= gia na exw epilogh del h ok
{
    if (ts.X >= 100 && ts.X < 160 && sp > 0) // DEL
    {
        BSP_LCD_SetBackColor(LCD_COLOR_RED);
        BSP_LCD_SetTextColor(LCD_COLOR_RED);
        BSP_LCD_DrawRect(86, 191, 54, 28);
        BSP_LCD_SetTextColor(LCD_COLOR_WHITE);
        BSP_LCD_DisplayStringAt(88,193,      (uint8_t      *)"DEL",
LEFT_MODE);

        sp--;
        set_apod_ar[sp]=0;
        temp_apod_ar[sp]=0;

        sprintf(s, " ");
        BSP_LCD_SetBackColor(LCD_COLOR_WHITE);
        BSP_LCD_DisplayStringAt(set_cursor_x, set_cursor_y, (uint8_t *)s,
LEFT_MODE);

        set_cursor_x= set_cursor_x - 25;
    }

else if ((ts.X >= 210 && ts.X < 260) && (sp == 2 || sp == 3)) // OK
{

```



```

        BSP_LCD_SetBackColor(LCD_COLOR_GREEN);
        BSP_LCD_SetTextColor(LCD_COLOR_GREEN);
        BSP_LCD_FillRect(197, 191, 37, 28);// OK
        BSP_LCD_SetTextColor(LCD_COLOR_WHITE);
        BSP_LCD_DisplayStringAt(198,193, (uint8_t *)"OK",
LEFT_MODE);

        spok=1;
        if(sp==2)
        {
            temp_apod_ar[0]=set_apod_ar[0];
            temp_apod_ar[1]=set_apod_ar[1];
            temp_apod_ar[2]=set_apod_ar[2];

            set_apod_ar[0]=temp_apod_ar[2];
            set_apod_ar[1]=temp_apod_ar[0];
            set_apod_ar[2]=temp_apod_ar[1];
        }
        //hal_uart
        HAL_UART_Transmit(&huart3, (uint8_t *)&buffer, sprintf(buffer,
"a%d%d%d", set_apod_ar[0],set_apod_ar[1],set_apod_ar[2]),1000);

        sp++;
        ok=1;

    }

    HAL_Delay(300);

    BSP_LCD_SetTextColor(LCD_COLOR_WHITE);
    BSP_LCD_FillRect(9, 147, 22, 28); // 0
    BSP_LCD_FillRect(40, 147, 22, 28); // 1
    BSP_LCD_FillRect(71, 147, 22, 28); // 2
    BSP_LCD_FillRect(102, 147, 22, 28); // 3

```

```

        BSP_LCD_FillRect(133, 147, 22, 28); // 4
        BSP_LCD_FillRect(164, 147, 22, 28); // 5
        BSP_LCD_FillRect(195, 147, 22, 28); // 6
        BSP_LCD_FillRect(226, 147, 22, 28); // 7
        BSP_LCD_FillRect(257, 147, 22, 28); // 8
        BSP_LCD_FillRect(288, 147, 22, 28); // 9
        BSP_LCD_FillRect(86, 191, 54, 28); // DEL
        BSP_LCD_FillRect(197, 193, 38, 28); // OK
    }

    else if (sp >=3 && spok == 0)
    {
        numbers();
    }

}

}

while((sp<4 && spok==0) && set_apod == 1);
spok=0;
sp=0;
set_cursor_x = 85;
BSP_LCD_Clear(LCD_COLOR_WHITE);
if (start == 2)
{
    start=2;
}

else {start=1;}

```

```

    }

void settings (void)
{
    BSP_LCD_Clear(LCD_COLOR_WHITE);
    temp_st=start;
    y=0;
    do
    {
        BSP_TS_GetState(&ts);

        BSP_LCD_SetBackColor(LCD_COLOR_RED);
        BSP_LCD_SetTextColor(LCD_COLOR_WHITE);
        BSP_LCD_DisplayStringAt(301, 1, (uint8_t *)"X", LEFT_MODE);

        BSP_LCD_SetTextColor(LCD_COLOR_BLACK);
        BSP_LCD_SetBackColor(LCD_COLOR_WHITE);
        BSP_LCD_DisplayStringAt(10, 5, (uint8_t *)"Settings", LEFT_MODE);
        BSP_LCD_DrawHLine(10,30,135);

        BSP_LCD_SetTextColor(LCD_COLOR_BLUE);
        BSP_LCD_DisplayStringAt(75, 70, (uint8_t *)"Connection", LEFT_MODE);
        BSP_LCD_DisplayStringAt(128, 110, (uint8_t *)"Mode", LEFT_MODE);
        BSP_LCD_DisplayStringAt(30, 150, (uint8_t *)"Change Password", LEFT_MODE);
        BSP_LCD_DisplayStringAt(27, 195, (uint8_t *)"Vathmos apodosis", LEFT_MODE);

        //BSP_TS_GetState(&ts);
        if(ts.TouchDetected)
        {
            if(ts.Y >= 1 && ts.Y < 20 && ts.X >= 295 && ts.X <= 340)
                //if(ts.Y >= 1500 && ts.Y < 2500 && ts.X >= 2750 && ts.X <= 3200)

```

```

    {
        y=1;
        start=3;
        BSP_LCD_Clear(LCD_COLOR_WHITE);
    }

else if(ts.Y >= 60 && ts.Y < 90 && ts.X >= 70 && ts.X <= 270)
{
    BSP_LCD_SetBackColor(LCD_COLOR_BLUE);
    BSP_LCD_SetTextColor(LCD_COLOR_WHITE);
    BSP_LCD_DisplayStringAt(75, 70, (uint8_t *)"Connection",
LEFT_MODE);

    HAL_Delay(250);
    BSP_LCD_SetTextColor(LCD_COLOR_WHITE);////////
    BSP_LCD_FillRect(0,50,320,200);////////
    BSP_LCD_SetBackColor(LCD_COLOR_WHITE);
    BSP_LCD_SetTextColor(LCD_COLOR_BLUE);
    BSP_LCD_DisplayStringAt(75, 70, (uint8_t *)"Connection",
LEFT_MODE);

    BSP_LCD_DisplayStringAt(128, 125, (uint8_t *)"          ",
LEFT_MODE);

    BSP_LCD_DisplayStringAt(30, 180, (uint8_t *)"          ",
LEFT_MODE);

    BSP_LCD_SetTextColor(LCD_COLOR_BLACK);
    BSP_LCD_DrawCircle(20,150,10);
    BSP_LCD_DrawCircle(215,150,10);

    BSP_LCD_SetTextColor(LCD_COLOR_BLUE);
    BSP_LCD_DisplayStringAt(40, 140, (uint8_t *)"Ethernet",
LEFT_MODE);

```

```

LEFT_MODE);
BSP_LCD_DisplayStringAt(235, 140, (uint8_t *)"WiFi",

if (start_w_e == 0 || (ethernet == 0 && wifi == 0))
{

    BSP_LCD_SetTextColor(LCD_COLOR_GRAY);
    BSP_LCD_FillCircle(20,150,7);
    BSP_LCD_FillCircle(215,150,7);

    start_w_e = 1;

}

if (ethernet == 1)
{
    BSP_LCD_SetTextColor(LCD_COLOR_GREEN);
    BSP_LCD_FillCircle(20,150,7); //ethernet
    BSP_LCD_SetTextColor(LCD_COLOR_GRAY);
    BSP_LCD_FillCircle(215,150,7); // WiFi
    BSP_LCD_SetTextColor(LCD_COLOR_BLUE);
    //ethernet=0;

}

else if (wifi == 1)
{
    BSP_LCD_SetTextColor(LCD_COLOR_GREEN);
    BSP_LCD_FillCircle(215,150,7); // WiFi
    BSP_LCD_SetTextColor(LCD_COLOR_GRAY);
    BSP_LCD_FillCircle(20,150,7); //ethernet
    BSP_LCD_SetTextColor(LCD_COLOR_BLUE);
    //wifi=0;
}

```

```

    }

    int x=0;
    do
    {
        BSP_TS_GetState(&ts);
        if(ts.TouchDetected)
        {
            if(ts.Y >= 1 && ts.Y < 20 && ts.X >= 290 && ts.X
<= 320)
            {
                x=1;
            }

            else if(ts.Y >= 135 && ts.Y < 160 && ts.X >= 25 &&
ts.X <= 180)
            {

                BSP_LCD_SetTextColor(LCD_COLOR_GREEN);
                BSP_LCD_FillCircle(20,150,7); //ethernet

                BSP_LCD_SetTextColor(LCD_COLOR_GRAY);
                BSP_LCD_FillCircle(215,150,7); // WiFi

                BSP_LCD_SetTextColor(LCD_COLOR_BLUE);
                ethernet=1;
                wifi=0;
                //x=1;
            }

            else if(ts.Y >= 135 && ts.Y < 160 && ts.X >= 230
&& ts.X <= 315)
            {

```

```

BSP_LCD_SetTextColor(LCD_COLOR_GREEN);
                                BSP_LCD_FillCircle(215,150,7); // WiFi

BSP_LCD_SetTextColor(LCD_COLOR_GRAY);
                                BSP_LCD_FillCircle(20,150,7); //ethernet

BSP_LCD_SetTextColor(LCD_COLOR_BLUE);
                                wifi=1;
                                ethernet=0;
                                //x=1;
                                }

                                else
                                {
                                    ts.TouchDetected;
                                }

                                }

                                }

                                while(x==0);

                                HAL_Delay(250);
                                BSP_LCD_SetTextColor(LCD_COLOR_WHITE);
                                BSP_LCD_FillRect(0,50,320,200);
                                HAL_Delay(100);
                                }

                                else if(ts.Y >= 100 && ts.Y < 135 && ts.X >= 135 && ts.X <= 215)
                                {

                                    BSP_LCD_SetBackColor(LCD_COLOR_BLUE);

```

```

        BSP_LCD_SetTextColor(LCD_COLOR_WHITE);
        BSP_LCD_DisplayStringAt(128, 110, (uint8_t *)"Mode",
LEFT_MODE);

        HAL_Delay(200);
        BSP_LCD_SetTextColor(LCD_COLOR_WHITE);////////
        BSP_LCD_FillRect(0,50,320,200);////////
        BSP_LCD_SetBackColor(LCD_COLOR_WHITE);
        BSP_LCD_SetTextColor(LCD_COLOR_BLUE);
        BSP_LCD_DisplayStringAt(76, 70, (uint8_t *)" Mode ",
LEFT_MODE);

        BSP_LCD_DisplayStringAt(128, 110, (uint8_t *)" ",
LEFT_MODE);

        BSP_LCD_DisplayStringAt(30, 180, (uint8_t *)" ",
LEFT_MODE);

        BSP_LCD_SetTextColor(LCD_COLOR_BLACK);
        BSP_LCD_DrawCircle(20,150,10);
        BSP_LCD_DrawCircle(225,150,10);

        BSP_LCD_SetTextColor(LCD_COLOR_BLUE);
        BSP_LCD_DisplayStringAt(40, 140, (uint8_t *)"Petrol",
LEFT_MODE);

        BSP_LCD_DisplayStringAt(245, 140, (uint8_t *)"Gas",
LEFT_MODE);

        if (start_m == 0 || (petrol == 0 && gas == 0))
        {
            BSP_LCD_SetTextColor(LCD_COLOR_GRAY);
            BSP_LCD_FillCircle(20,150,7);
            BSP_LCD_FillCircle(225,150,7);

            start_m = 1;
        }

```



```

if (petrol == 1)
{
    BSP_LCD_SetTextColor(LCD_COLOR_GREEN);
    BSP_LCD_FillCircle(20,150,7); //petrol
    BSP_LCD_SetTextColor(LCD_COLOR_GRAY);
    BSP_LCD_FillCircle(225,150,7); // gas
    BSP_LCD_SetTextColor(LCD_COLOR_BLUE);
}

else if (gas == 1)
{
    BSP_LCD_SetTextColor(LCD_COLOR_GREEN);
    BSP_LCD_FillCircle(225,150,7); // gas
    BSP_LCD_SetTextColor(LCD_COLOR_GRAY);
    BSP_LCD_FillCircle(20,150,7); //petrol
    BSP_LCD_SetTextColor(LCD_COLOR_BLUE);
}

int x=0;
do
{

    BSP_TS_GetState(&ts);
    if(ts.TouchDetected)
    {

        if(ts.Y >= 135 && ts.Y < 160 && ts.X >= 25 && ts.X
<= 175)
        {

```

```

BSP_LCD_SetTextColor(LCD_COLOR_GREEN);
                                BSP_LCD_FillCircle(20,150,7); //gas

BSP_LCD_SetTextColor(LCD_COLOR_GRAY);
                                BSP_LCD_FillCircle(225,150,7); // petrol

BSP_LCD_SetTextColor(LCD_COLOR_BLUE);
                                petrol=1;
                                gas=0;
                                }

                                else if(ts.Y >= 135 && ts.Y < 160 && ts.X >= 240
&& ts.X <= 315)
                                {

BSP_LCD_SetTextColor(LCD_COLOR_GREEN);
                                BSP_LCD_FillCircle(225,150,7); // gas

BSP_LCD_SetTextColor(LCD_COLOR_GRAY);
                                BSP_LCD_FillCircle(20,150,7); //petrol

BSP_LCD_SetTextColor(LCD_COLOR_BLUE);
                                gas=1;
                                petrol=0;
                                }

                                else if(ts.Y >= 1 && ts.Y < 20 && ts.X >= 290 &&
ts.X <= 320)
                                {
                                x=1;
                                }

```

```

        else
        {
            ts.TouchDetected;
        }
    }

}

while(x==0);
HAL_Delay(250);
BSP_LCD_SetTextColor(LCD_COLOR_WHITE);
BSP_LCD_FillRect(0,50,320,200);
HAL_Delay(100);
}

else if(ts.Y >= 140 && ts.Y < 175 && ts.X >= 110 && ts.X <= 310)
{
    change_pass=1;
    BSP_LCD_SetBackColor(LCD_COLOR_BLUE);
    BSP_LCD_SetTextColor(LCD_COLOR_WHITE);
    BSP_LCD_DisplayStringAt(30, 150, (uint8_t *)"Change Password",
LEFT_MODE);

    HAL_Delay(250);
    BSP_LCD_Clear(LCD_COLOR_WHITE);
    set_password();
}

else if(ts.Y >= 195 && ts.Y < 220 && ts.X >= 110 && ts.X <= 310)
{
    set_apod=1;
    //change_pass=0;
    BSP_LCD_SetBackColor(LCD_COLOR_BLUE);
    BSP_LCD_SetTextColor(LCD_COLOR_WHITE);

```

```

        BSP_LCD_DisplayStringAt(27, 195, (uint8_t *)"Vathmos apodosis",
LEFT_MODE);

        HAL_Delay(250);

        BSP_LCD_Clear(LCD_COLOR_WHITE);

        set_apodosh();

    }

    HAL_Delay(300);

}

}

while(y==0);

start=3;

}

void index (void)
{
    z=0;

    BSP_LCD_SetBackColor(LCD_COLOR_RED);
    BSP_LCD_SetTextColor(LCD_COLOR_WHITE);
    BSP_LCD_DisplayStringAt(301, 1, (uint8_t *)"X", LEFT_MODE);

    BSP_LCD_SetBackColor(LCD_COLOR_WHITE);
    BSP_LCD_SetTextColor(LCD_COLOR_BLUE);
    BSP_LCD_DisplayStringAt(95, 70, (uint8_t *)"Settings", LEFT_MODE);
    BSP_LCD_DisplayStringAt(7, 150, (uint8_t *)"Metrhsh Deksamensh", LEFT_MODE);

    do
    {
        BSP_TS_GetState(&ts);

        if(ts.TouchDetected)
        {

```

```

if(ts.Y >= 1 && ts.Y < 20 && ts.X >= 295 && ts.X <= 340)
{
    z=1;
    BSP_LCD_Clear(LCD_COLOR_WHITE);
    BSP_LCD_SetTextColor(LCD_COLOR_BLACK);
    BSP_LCD_DrawRect(5,60, 310,100);
    BSP_LCD_SetFont(&Font16);
    BSP_LCD_DisplayStringAt(270, 5, (uint8_t *)"LOCK",
LEFT_MODE);

    BSP_LCD_SetFont(&Font20);
    BSP_LCD_DisplayStringAt(8, 80, (uint8_t *)"Connection:",
LEFT_MODE);

    BSP_LCD_DisplayStringAt(10, 125, (uint8_t *)"Mode:",
LEFT_MODE);

    BSP_LCD_SetFont(&Font24);
    BSP_LCD_SetTextColor(LCD_COLOR_ORANGE);
    if (wifi == 1){BSP_LCD_DisplayStringAt(170, 77, (uint8_t
*)"WiFi", LEFT_MODE);}

    else{BSP_LCD_DisplayStringAt(170, 77, (uint8_t
*)"Ethernet", LEFT_MODE);}

    if (gas == 1){BSP_LCD_DisplayStringAt(90, 122, (uint8_t
*)"Gas", LEFT_MODE);}

    else {BSP_LCD_DisplayStringAt(90, 122, (uint8_t
*)"Petrol", LEFT_MODE);}

    if (wifi == 0 && ethernet == 0 && gas == 0 && petrol == 0)
    {
        BSP_LCD_DisplayStringAt(170, 77, (uint8_t *)"No
Setup", LEFT_MODE);

        BSP_LCD_DisplayStringAt(90, 122, (uint8_t *)"No
Setup", LEFT_MODE);

    }
    BSP_LCD_SetTextColor(LCD_COLOR_BLACK);
    start=4;
}

```

```

else if(ts.Y >=60 && ts.Y < 90 && ts.X >= 100 && ts.X <= 250)
{
    BSP_LCD_SetBackColor(LCD_COLOR_BLUE);
    BSP_LCD_SetTextColor(LCD_COLOR_WHITE);
    BSP_LCD_DisplayStringAt(95, 70, (uint8_t *)"Settings",
LEFT_MODE);

    HAL_Delay(250);
    //BSP_LCD_Clear(LCD_COLOR_WHITE);
    start=2;
    z=1;
}

else if(ts.Y >=140 && ts.Y < 170 && ts.X >= 5 && ts.X <= 350)
{
    BSP_LCD_SetBackColor(LCD_COLOR_BLUE);
    BSP_LCD_SetTextColor(LCD_COLOR_WHITE);
    BSP_LCD_DisplayStringAt(7, 150, (uint8_t *)"Metrhsh
Deksamenhs", LEFT_MODE);

    HAL_Delay(250);
    BSP_LCD_Clear(LCD_COLOR_WHITE);
    HAL_Delay(300);
    int m=0;

    BSP_LCD_SetTextColor(LCD_COLOR_WHITE);////////
    BSP_LCD_FillRect(0,50,320,200);////////
    BSP_LCD_SetBackColor(LCD_COLOR_WHITE);
    BSP_LCD_SetTextColor(LCD_COLOR_BLUE);
    BSP_LCD_SetFont(&Font20);
    BSP_LCD_DisplayStringAt(10, 10, (uint8_t *)"Na ginei
metrhsh ths", LEFT_MODE);

    BSP_LCD_DisplayStringAt(10, 30, (uint8_t *)"apostashs
deksamenhs?", LEFT_MODE);

```

```

LEFT_MODE);
BSP_LCD_DisplayStringAt(128, 125, (uint8_t *)"    ",
LEFT_MODE);
BSP_LCD_DisplayStringAt(30, 180, (uint8_t *)"    ",
LEFT_MODE);

BSP_LCD_SetFont(&Font24);

BSP_LCD_SetTextColor(LCD_COLOR_GREEN);
BSP_LCD_DisplayStringAt(69, 112, (uint8_t *)"Nai",
LEFT_MODE);

BSP_LCD_SetTextColor(LCD_COLOR_RED);
BSP_LCD_DisplayStringAt(206, 112, (uint8_t *)"Oxi",
LEFT_MODE);

BSP_LCD_SetTextColor(LCD_COLOR_GREEN);
BSP_LCD_DrawRect(68, 110, 54, 28);// Nai
BSP_LCD_SetTextColor(LCD_COLOR_RED);
BSP_LCD_DrawRect(205, 110, 54, 28);// Oxi

do
{
    BSP_TS_GetState(&ts);

    if(ts.TouchDetected)
    {
        if(ts.Y >= 95 && ts.Y < 145 && ts.X >= 70
&& ts.X <= 140) //nai
        {
            BSP_LCD_SetBackColor(LCD_COLOR_GREEN);

            BSP_LCD_SetTextColor(LCD_COLOR_GREEN);
            BSP_LCD_FillRect(70, 110, 54,
28);// OK

            BSP_LCD_SetTextColor(LCD_COLOR_WHITE);

```

```

(uint8_t*)"Nai", LEFT_MODE);

BSP_LCD_DisplayStringAt(69,112,

HAL_Delay(250);

m=1;

z=1;

start=5;

BSP_LCD_SetTextColor(LCD_COLOR_WHITE);

BSP_LCD_FillRect(5, 5, 320, 240);

}

else if(ts.Y >= 95 && ts.Y < 145 && ts.X >=
210 && ts.X <= 290) //oxi

{

BSP_LCD_SetBackColor(LCD_COLOR_RED);

BSP_LCD_SetTextColor(LCD_COLOR_RED);

BSP_LCD_SetTextColor(LCD_COLOR_WHITE);

BSP_LCD_DisplayStringAt(206,112, (uint8_t*)"Oxi", LEFT_MODE);

m=1;

z=1;

start=3;

HAL_Delay(250);

BSP_LCD_SetTextColor(LCD_COLOR_WHITE);

BSP_LCD_FillRect(5, 5, 320, 240);

}

}

}

while(m==0);

```



```

        }
        HAL_Delay(300);
    }
}
while(z==0);
}

```

```

//char s[5], d[5];
//
//sprintf(s, "Y%d", (int)ts.Y);
//sprintf(d, "X%d", (int)ts.X);
//BSP_LCD_DisplayStringAt(10, 50, s, LEFT_MODE);
//BSP_LCD_DisplayStringAt(0, 100, d, LEFT_MODE);

```

```

BSP_LCD_Init();
BSP_LCD_Clear(LCD_COLOR_WHITE);
HCSR04_Read();
Temp_Hum();
Temp_Hum_IN();

```

```

while (1)

```

```

{

```

```

    /* USER CODE END WHILE */

```

```

    /* USER CODE BEGIN 3 */

```

```

if (start == 0) // mono thn 1 fora pou tha treksei gia na fortwthoun times stis metavlhtes
{
    HCSR04_Read();
    Temp_Hum();
    Temp_Hum_IN();
    set_password();

}

if (start == 1)
{
    BSP_LCD_Clear(LCD_COLOR_WHITE);
    password();
}

if (start == 2)
{
    settings();
}

if(start==3)
{
    index();
}

if(start==5) // metrhsh deksamens prin kai otan ginei h plhrwsh
{
    HCSR04_Read();
    start=3;
}

```

```

BSP_TS_GetState(&ts);
if(ts.TouchDetected)
{
    start=1;
}

if (s1_f == 0 && s2_f == 0 && s3_f == 0 && s4_f == 0 && s5_f == 0) // 0 0 0
{
    //HAL_GPIO_WritePin(GPIOC, GPIO_PIN_13, GPIO_PIN_SET); //pcb_led off
    HAL_GPIO_WritePin(GPIOA, GPIO_PIN_12, GPIO_PIN_RESET); // buzzer
    HAL_GPIO_WritePin(GPIOA, GPIO_PIN_11, GPIO_PIN_RESET); // pump
    overflow=0, leak=0, smoke=0;
    f_h_h_h=0, f_h_l_h=0, f_l_h_h=0, f_l_l_h=0, f_h_h_l=0, f_h_l_l=0, f_l_h_l=0;
}

/* O L S */

else if (s1_f == 0 && s2_f == 0 && s3_f == 1 && s4_f == 1 && s5_f == 0) // 0 1 0
{
    HAL_GPIO_WritePin(GPIOA, GPIO_PIN_11, GPIO_PIN_SET); // pump
    overflow=0, leak=1, smoke=0;
    f_h_h_h=0, f_h_l_h=0, f_l_h_h=0, f_l_l_h=0, f_h_h_l=0, f_h_l_l=0;
    buzzer();

    if (f_l_h_l == 0)
    {
        f_l_h_l = 1;
        event();
    }
}

```

```

else if (s1_f == 1 && s2_f == 1 && s3_f == 0 && s4_f == 0 && s5_f == 0) // 1 0 0
{
    HAL_GPIO_WritePin(GPIOA, GPIO_PIN_11, GPIO_PIN_RESET); // pump
    overflow=1, leak=0, smoke=0;
    f_h_h_h=0, f_h_l_h=0, f_l_h_h=0, f_l_l_h=0, f_h_h_l=0, f_l_h_l=0;
    buzzer();

    if (f_h_l_l == 0)
    {
        f_h_l_l = 1;
        event();
    }
}

```

```

else if (s1_f == 1 && s2_f == 1 && s3_f == 1 && s4_f == 1 && s5_f == 0) // 1 1 0
{
    HAL_GPIO_WritePin(GPIOA, GPIO_PIN_11, GPIO_PIN_SET); // pump
    overflow=1, leak=1, smoke=0;
    f_h_h_h=0, f_h_l_h=0, f_l_h_h=0, f_l_l_h=0, f_h_l_l=0, f_l_h_l=0;
    buzzer();

    if (f_h_h_l == 0)
    {
        f_h_h_l = 1;
        event();
    }
}

```

```

else if (s1_f == 0 && s2_f == 0 && s3_f == 0 && s4_f == 0 && s5_f == 1) // 0 0 1
{
    HAL_GPIO_WritePin(GPIOA, GPIO_PIN_11, GPIO_PIN_RESET); // pump

```

```

overflow=0, leak=0, smoke=1;
f_h_h_h=0, f_h_l_h=0, f_l_h_h=0, f_h_h_l=0, f_h_l_l=0, f_l_h_l=0,
buzzer();

if (f_l_l_h == 0)
{
    f_l_l_h = 1;
    event();
}
}

else if (s1_f == 0 && s2_f == 0 && s3_f == 1 && s4_f == 1 && s5_f == 1) // 0 1 1
{
    HAL_GPIO_WritePin(GPIOA, GPIO_PIN_11, GPIO_PIN_SET); // pump
    overflow=0, leak=1, smoke=1;
    f_h_h_h=0, f_h_l_h=0, f_l_l_h=0, f_h_h_l=0, f_h_l_l=0, f_l_h_l=0,
    buzzer();

    if (f_l_h_h == 0)
    {
        f_l_h_h = 1;
        event();
    }
}

else if (s1_f == 1 && s2_f == 1 && s3_f == 0 && s4_f == 0 && s5_f == 1) // 1 0 1
{
    HAL_GPIO_WritePin(GPIOA, GPIO_PIN_11, GPIO_PIN_RESET); // pump
    overflow=1, leak=0, smoke=1;
    f_h_h_h=0, f_l_h_h=0, f_l_l_h=0, f_h_h_l=0, f_h_l_l=0, f_l_h_l=0,
    buzzer();
}

```

```

    if (f_h_l_h == 0)
    {
        f_h_l_h = 1;
        event();
    }
}

else if (s1_f == 1 && s2_f == 1 && s3_f == 1 && s4_f == 1 && s5_f == 1) // 1 1 1
{
    HAL_GPIO_WritePin(GPIOA, GPIO_PIN_11, GPIO_PIN_SET); // pump
    overflow=1, leak=1, smoke=1;
    f_h_l_h=0, f_l_h_h=0, f_l_l_h=0, f_h_h_l=0, f_h_l_l=0, f_l_h_l=0,
    buzzer();

    if (f_h_h_h == 0)
    {
        f_h_h_h = 1;
        event();
    }
}

if (time_to_read == 1) // metrshsh apostashs sth deksamenh
{
    HCSR04_Read();
    time_to_read = 0;
    uic_send = 0;
    count_min_uic=0;
}

if (s6_f == 1) // on kausthras
{

```

```

HAL_TIM_Base_Start_IT(&htim4);
tim5_min=0;

if (tim4_min == 5) // kathe 5 lepta
{
    Temp_Hum(); //ekswterikh
    //HAL_Delay(250);
    Temp_Hum_IN(); // eswterikh
    tim4_min=0; // mhdenizei gia na ksenisoun ta epomena 5 lepta
}

count_min_uic=0; // mhdenizei giati tha kanei post apo to temp-hum kai etsi sto
backend tha parei last timestamp
uic_send = 0;
}

else if (s6_f == 0) // off kausthras
{
    HAL_TIM_Base_Stop_IT(&htim4);
    // HAL_GPIO_WritePin(GPIOC, GPIO_PIN_13, GPIO_PIN_RESET);
    tim4_min=0;
    tim4_sec=0;

    if(tim5_min == 5) // metrshsh thermokrasias sto levhtostasio kathe 5 lepta
    {
        Temp_Hum_IN(); // eswterikh
        tim5_min=0;
    }
}
}

```

```
/* dummy json kathe 5 lepta efonon den exei ginei post apo kati allo (distance, events, temp-hum) */
```

```
    if (s6_f == 0 && time_to_read == 0
        && s1_f == 0 && s2_f == 0 && s3_f == 0 && s4_f == 0 && s5_f == 0
        && uic_send == 1)
    {
        uic();
        uic_send=0;
    }
```

```
}
```

```
/* USER CODE END 3 */
```

```
}
```

```
/**
```

```
 * @brief System Clock Configuration
```

```
 * @retval None
```

```
 */
```

```
void SystemClock_Config(void)
```

```
{
```

```
    RCC_OscInitTypeDef RCC_OscInitStruct = {0};
```

```
    RCC_ClkInitTypeDef RCC_ClkInitStruct = {0};
```

```
/** Initializes the RCC Oscillators according to the specified parameters
```

```
 * in the RCC_OscInitTypeDef structure.
```

```
 */
```

```
RCC_OscInitStruct.OscillatorType = RCC_OSCILLATORTYPE_HSE;
```

```
RCC_OscInitStruct.HSEState = RCC_HSE_ON;
```

```
RCC_OscInitStruct.HSEPredivValue = RCC_HSE_PREDIV_DIV1;
```

```
RCC_OscInitStruct.HSIState = RCC_HSI_ON;
```



```

RCC_OscInitStruct.PLL.PLLState = RCC_PLL_ON;
RCC_OscInitStruct.PLL.PLLSource = RCC_PLLSOURCE_HSE;
RCC_OscInitStruct.PLL.PLLMUL = RCC_PLL_MUL9;
if (HAL_RCC_OscConfig(&RCC_OscInitStruct) != HAL_OK)
{
    Error_Handler();
}
/** Initializes the CPU, AHB and APB buses clocks
*/
RCC_ClkInitStruct.ClockType = RCC_CLOCKTYPE_HCLK|RCC_CLOCKTYPE_SYSCLK
                               |RCC_CLOCKTYPE_PCLK1|RCC_CLOCKTYPE_PCLK2;
RCC_ClkInitStruct.SYSCLKSource = RCC_SYSCLKSOURCE_PLLCLK;
RCC_ClkInitStruct.AHBCLKDivider = RCC_SYSCLK_DIV1;
RCC_ClkInitStruct.APB1CLKDivider = RCC_HCLK_DIV2;
RCC_ClkInitStruct.APB2CLKDivider = RCC_HCLK_DIV1;

if (HAL_RCC_ClockConfig(&RCC_ClkInitStruct, FLASH_LATENCY_2) != HAL_OK)
{
    Error_Handler();
}
}

/**
 * @brief SPI1 Initialization Function
 * @param None
 * @retval None
 */
static void MX_SPI1_Init(void)
{
    /** USER CODE BEGIN SPI1_Init 0 */

```

```

/* USER CODE END SPI1_Init 0 */

/* USER CODE BEGIN SPI1_Init 1 */

/* USER CODE END SPI1_Init 1 */
/* SPI1 parameter configuration*/
hspi1.Instance = SPI1;
hspi1.Init.Mode = SPI_MODE_MASTER;
hspi1.Init.Direction = SPI_DIRECTION_2LINES;
hspi1.Init.DataSize = SPI_DATASIZE_8BIT;
hspi1.Init.CLKPolarity = SPI_POLARITY_LOW;
hspi1.Init.CLKPhase = SPI_PHASE_1EDGE;
hspi1.Init.NSS = SPI_NSS_SOFT;
hspi1.Init.BaudRatePrescaler = SPI_BAUDRATEPRESCALER_4;
hspi1.Init.FirstBit = SPI_FIRSTBIT_MSB;
hspi1.Init.TIMode = SPI_TIMODE_DISABLE;
hspi1.Init.CRCCalculation = SPI_CRCCALCULATION_DISABLE;
hspi1.Init.CRCPolynomial = 10;
if (HAL_SPI_Init(&hspi1) != HAL_OK)
{
    Error_Handler();
}
/* USER CODE BEGIN SPI1_Init 2 */

/* USER CODE END SPI1_Init 2 */

}

/**
 * @brief SPI2 Initialization Function
 * @param None
 * @retval None

```

```

*/
static void MX_SPI2_Init(void)
{

/* USER CODE BEGIN SPI2_Init 0 */

/* USER CODE END SPI2_Init 0 */

/* USER CODE BEGIN SPI2_Init 1 */

/* USER CODE END SPI2_Init 1 */
/* SPI2 parameter configuration*/
hspi2.Instance = SPI2;
hspi2.Init.Mode = SPI_MODE_MASTER;
hspi2.Init.Direction = SPI_DIRECTION_2LINES;
hspi2.Init.DataSize = SPI_DATASIZE_8BIT;
hspi2.Init.CLKPolarity = SPI_POLARITY_LOW;
hspi2.Init.CLKPhase = SPI_PHASE_1EDGE;
hspi2.Init.NSS = SPI_NSS_SOFT;
hspi2.Init.BaudRatePrescaler = SPI_BAUDRATEPRESCALER_32;
hspi2.Init.FirstBit = SPI_FIRSTBIT_MSB;
hspi2.Init.TIMode = SPI_TIMODE_DISABLE;
hspi2.Init.CRCCalculation = SPI_CRCCALCULATION_DISABLE;
hspi2.Init.CRCPolynomial = 10;
if (HAL_SPI_Init(&hspi2) != HAL_OK)
{
Error_Handler();
}
/* USER CODE BEGIN SPI2_Init 2 */

/* USER CODE END SPI2_Init 2 */

```

```

}

/**
 * @brief TIM2 Initialization Function
 * @param None
 * @retval None
 */
static void MX_TIM2_Init(void)
{

/* USER CODE BEGIN TIM2_Init 0 */

/* USER CODE END TIM2_Init 0 */

TIM_MasterConfigTypeDef sMasterConfig = {0};
TIM_IC_InitTypeDef sConfigIC = {0};

/* USER CODE BEGIN TIM2_Init 1 */

/* USER CODE END TIM2_Init 1 */
htim2.Instance = TIM2;
htim2.Init.Prescaler = 72-1;
htim2.Init.CounterMode = TIM_COUNTERMODE_UP;
htim2.Init.Period = 65535-1;
htim2.Init.ClockDivision = TIM_CLOCKDIVISION_DIV1;
htim2.Init.AutoReloadPreload = TIM_AUTORELOAD_PRELOAD_DISABLE;
if (HAL_TIM_IC_Init(&htim2) != HAL_OK)
{
    Error_Handler();
}
sMasterConfig.MasterOutputTrigger = TIM_TRGO_RESET;
sMasterConfig.MasterSlaveMode = TIM_MASTERSLAVEMODE_DISABLE;

```

```

if (HAL_TIMEx_MasterConfigSynchronization(&htim2, &sMasterConfig) != HAL_OK)
{
    Error_Handler();
}
sConfigIC.ICPolarity = TIM_INPUTCHANNELPOLARITY_RISING;
sConfigIC.ICSelection = TIM_ICSELECTION_DIRECTTI;
sConfigIC.ICPrescaler = TIM_ICPSC_DIV1;
sConfigIC.ICFilter = 0;
if (HAL_TIM_IC_ConfigChannel(&htim2, &sConfigIC, TIM_CHANNEL_2) != HAL_OK)
{
    Error_Handler();
}
/* USER CODE BEGIN TIM2_Init 2 */

/* USER CODE END TIM2_Init 2 */

}

/**
 * @brief TIM3 Initialization Function
 * @param None
 * @retval None
 */
static void MX_TIM3_Init(void)
{
    /* USER CODE BEGIN TIM3_Init 0 */

    /* USER CODE END TIM3_Init 0 */

    TIM_ClockConfigTypeDef sClockSourceConfig = {0};
    TIM_MasterConfigTypeDef sMasterConfig = {0};

```

```

/* USER CODE BEGIN TIM3_Init 1 */

/* USER CODE END TIM3_Init 1 */
htim3.Instance = TIM3;
htim3.Init.Prescaler = 1200-1;
htim3.Init.CounterMode = TIM_COUNTERMODE_UP;
htim3.Init.Period = 60000-1;
htim3.Init.ClockDivision = TIM_CLOCKDIVISION_DIV1;
htim3.Init.AutoReloadPreload = TIM_AUTORELOAD_PRELOAD_DISABLE;
if (HAL_TIM_Base_Init(&htim3) != HAL_OK)
{
    Error_Handler();
}
sClockSourceConfig.ClockSource = TIM_CLOCKSOURCE_INTERNAL;
if (HAL_TIM_ConfigClockSource(&htim3, &sClockSourceConfig) != HAL_OK)
{
    Error_Handler();
}
sMasterConfig.MasterOutputTrigger = TIM_TRGO_RESET;
sMasterConfig.MasterSlaveMode = TIM_MASTERSLAVEMODE_DISABLE;
if (HAL_TIMEx_MasterConfigSynchronization(&htim3, &sMasterConfig) != HAL_OK)
{
    Error_Handler();
}
/* USER CODE BEGIN TIM3_Init 2 */

/* USER CODE END TIM3_Init 2 */

}

/**

```

```

* @brief TIM4 Initialization Function
* @param None
* @retval None
*/
static void MX_TIM4_Init(void)
{

/* USER CODE BEGIN TIM4_Init 0 */

/* USER CODE END TIM4_Init 0 */

TIM_ClockConfigTypeDef sClockSourceConfig = {0};
TIM_MasterConfigTypeDef sMasterConfig = {0};

/* USER CODE BEGIN TIM4_Init 1 */

/* USER CODE END TIM4_Init 1 */
htim4.Instance = TIM4;
htim4.Init.Prescaler = 1200-1;
htim4.Init.CounterMode = TIM_COUNTERMODE_UP;
htim4.Init.Period = 60000-1;
htim4.Init.ClockDivision = TIM_CLOCKDIVISION_DIV1;
htim4.Init.AutoReloadPreload = TIM_AUTORELOAD_PRELOAD_DISABLE;
if (HAL_TIM_Base_Init(&htim4) != HAL_OK)
{
Error_Handler();
}
sClockSourceConfig.ClockSource = TIM_CLOCKSOURCE_INTERNAL;
if (HAL_TIM_ConfigClockSource(&htim4, &sClockSourceConfig) != HAL_OK)
{
Error_Handler();
}
}

```

```

sMasterConfig.MasterOutputTrigger = TIM_TRGO_RESET;
sMasterConfig.MasterSlaveMode = TIM_MASTERSLAVEMODE_DISABLE;
if (HAL_TIMEx_MasterConfigSynchronization(&htim4, &sMasterConfig) != HAL_OK)
{
    Error_Handler();
}
/* USER CODE BEGIN TIM4_Init 2 */

/* USER CODE END TIM4_Init 2 */

}

/**
 * @brief USART3 Initialization Function
 * @param None
 * @retval None
 */
static void MX_USART3_UART_Init(void)
{
    /* USER CODE BEGIN USART3_Init 0 */

    /* USER CODE END USART3_Init 0 */

    /* USER CODE BEGIN USART3_Init 1 */

    /* USER CODE END USART3_Init 1 */
    huart3.Instance = USART3;
    huart3.Init.BaudRate = 115200;
    huart3.Init.WordLength = UART_WORDLENGTH_8B;
    huart3.Init.StopBits = UART_STOPBITS_1;
    huart3.Init.Parity = UART_PARITY_NONE;

```



```

huart3.Init.Mode = UART_MODE_TX_RX;
huart3.Init.HwFlowCtl = UART_HWCONTROL_NONE;
huart3.Init.OverSampling = UART_OVERSAMPLING_16;
if (HAL_UART_Init(&huart3) != HAL_OK)
{
    Error_Handler();
}
/* USER CODE BEGIN USART3_Init 2 */

/* USER CODE END USART3_Init 2 */

}

/**
 * Enable DMA controller clock
 */
static void MX_DMA_Init(void)
{

    /* DMA controller clock enable */
    __HAL_RCC_DMA1_CLK_ENABLE();

    /* DMA interrupt init */
    /* DMA1_Channel3_IRQn interrupt configuration */
    HAL_NVIC_SetPriority(DMA1_Channel3_IRQn, 0, 0);
    HAL_NVIC_EnableIRQ(DMA1_Channel3_IRQn);

}

/**
 * @brief GPIO Initialization Function
 * @param None

```

```

* @retval None
*/
static void MX_GPIO_Init(void)
{
    GPIO_InitTypeDef GPIO_InitStructure = {0};

    /* GPIO Ports Clock Enable */
    __HAL_RCC_GPIOC_CLK_ENABLE();
    __HAL_RCC_GPIOD_CLK_ENABLE();
    __HAL_RCC_GPIOA_CLK_ENABLE();
    __HAL_RCC_GPIOB_CLK_ENABLE();

    /*Configure GPIO pin Output Level */
    HAL_GPIO_WritePin(LED_GPIO_Port, LED_Pin, GPIO_PIN_SET);

    /*Configure GPIO pin Output Level */
    HAL_GPIO_WritePin(GPIOA, T_H_Pin|LCD_RST_Pin|T_H_IN_Pin|Pump_Pin
        |Buzzer_Pin|Trig_Pin, GPIO_PIN_RESET);

    /*Configure GPIO pin Output Level */
    HAL_GPIO_WritePin(GPIOB, TS_CS_Pin|LCD_CS_Pin, GPIO_PIN_SET);

    /*Configure GPIO pin Output Level */
    HAL_GPIO_WritePin(LCD_RS_GPIO_Port, LCD_RS_Pin, GPIO_PIN_RESET);

    /*Configure GPIO pin : LED_Pin */
    GPIO_InitStructure.Pin = LED_Pin;
    GPIO_InitStructure.Mode = GPIO_MODE_OUTPUT_PP;
    GPIO_InitStructure.Pull = GPIO_NOPULL;
    GPIO_InitStructure.Speed = GPIO_SPEED_FREQ_LOW;
    HAL_GPIO_Init(LED_GPIO_Port, &GPIO_InitStructure);

```

```

/*Configure GPIO pins : sens3_Pin sens4_Pin */
GPIO_InitStruct.Pin = sens3_Pin|sens4_Pin;
GPIO_InitStruct.Mode = GPIO_MODE_IT_RISING;
GPIO_InitStruct.Pull = GPIO_NOPULL;
HAL_GPIO_Init(GPIOC, &GPIO_InitStruct);

/*Configure GPIO pins : T_H_Pin LCD_RST_Pin T_H_IN_Pin Pump_Pin
                        Buzzer_Pin Trig_Pin */
GPIO_InitStruct.Pin = T_H_Pin|LCD_RST_Pin|T_H_IN_Pin|Pump_Pin
                    |Buzzer_Pin|Trig_Pin;
GPIO_InitStruct.Mode = GPIO_MODE_OUTPUT_PP;
GPIO_InitStruct.Pull = GPIO_NOPULL;
GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_LOW;
HAL_GPIO_Init(GPIOA, &GPIO_InitStruct);

/*Configure GPIO pin : sens5_Pin */
GPIO_InitStruct.Pin = sens5_Pin;
GPIO_InitStruct.Mode = GPIO_MODE_IT_RISING;
GPIO_InitStruct.Pull = GPIO_PULLUP;
HAL_GPIO_Init(sens5_GPIO_Port, &GPIO_InitStruct);

/*Configure GPIO pins : TS_CS_Pin LCD_RS_Pin LCD_CS_Pin */
GPIO_InitStruct.Pin = TS_CS_Pin|LCD_RS_Pin|LCD_CS_Pin;
GPIO_InitStruct.Mode = GPIO_MODE_OUTPUT_PP;
GPIO_InitStruct.Pull = GPIO_NOPULL;
GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_HIGH;
HAL_GPIO_Init(GPIOB, &GPIO_InitStruct);

/*Configure GPIO pin : TS_IRQ_Pin */
GPIO_InitStruct.Pin = TS_IRQ_Pin;
GPIO_InitStruct.Mode = GPIO_MODE_INPUT;
GPIO_InitStruct.Pull = GPIO_PULLUP;

```

```
HAL_GPIO_Init(TS_IRQ_GPIO_Port, &GPIO_InitStruct);
```

```
/*Configure GPIO pins : sens1_Pin sens2_Pin */
```

```
GPIO_InitStruct.Pin = sens1_Pin|sens2_Pin;
```

```
GPIO_InitStruct.Mode = GPIO_MODE_IT_RISING;
```

```
GPIO_InitStruct.Pull = GPIO_NOPULL;
```

```
HAL_GPIO_Init(GPIOA, &GPIO_InitStruct);
```

```
/*Configure GPIO pin : sens6_Pin */
```

```
GPIO_InitStruct.Pin = sens6_Pin;
```

```
GPIO_InitStruct.Mode = GPIO_MODE_IT_FALLING;
```

```
GPIO_InitStruct.Pull = GPIO_NOPULL;
```

```
HAL_GPIO_Init(sens6_GPIO_Port, &GPIO_InitStruct);
```

```
/* EXTI interrupt init*/
```

```
HAL_NVIC_SetPriority(EXTI1_IRQn, 0, 0);
```

```
HAL_NVIC_EnableIRQ(EXTI1_IRQn);
```

```
HAL_NVIC_SetPriority(EXTI9_5_IRQn, 0, 0);
```

```
HAL_NVIC_EnableIRQ(EXTI9_5_IRQn);
```

```
HAL_NVIC_SetPriority(EXTI15_10_IRQn, 0, 0);
```

```
HAL_NVIC_EnableIRQ(EXTI15_10_IRQn);
```

```
}
```

```
/* USER CODE BEGIN 4 */
```

```
/* USER CODE END 4 */
```

```

/**
 * @brief This function is executed in case of error occurrence.
 * @retval None
 */
void Error_Handler(void)
{
    /* USER CODE BEGIN Error_Handler_Debug */
    /* User can add his own implementation to report the HAL error return state */
    __disable_irq();
    while (1)
    {
    }
    /* USER CODE END Error_Handler_Debug */
}

#ifdef USE_FULL_ASSERT
/**
 * @brief Reports the name of the source file and the source line number
 *        where the assert_param error has occurred.
 * @param file: pointer to the source file name
 * @param line: assert_param error line source number
 * @retval None
 */
void assert_failed(uint8_t *file, uint32_t line)
{
    /* USER CODE BEGIN 6 */
    /* User can add his own implementation to report the file name and line number,
       ex: printf("Wrong parameters value: file %s on line %d\r\n", file, line) */
    /* USER CODE END 6 */
}
#endif /* USE_FULL_ASSERT */

```

## ΠΑΡΑΡΤΗΜΑ Β : Κώδικας για το ESP-01

```
#include <ESP8266WiFi.h>
#include <ESP8266HTTPClient.h>
#include <ArduinoJson.h>

const int LED = 2;
const char* ssid = "xxxxxxxxxx";
const char* password = "xxxxxxxxxx";
const char* serverName = "http://192.168.2.11:5000/device_post/";

int x=0, smoke=0, leak=0, overflow=0, rlen=0, apod=0, ev_temp=0;
float temp=0.0, hum=0.0, dist=0.0;
char str[11]="\0";
bool ledState;

WiFiClient client;
HTTPClient http;

/* // Timer variables - gia delay metaksu twv post an einai anagkaio
unsigned long lastTime = 0;
unsigned long timerDelay = 10000; */

void setup()
{
  Serial.begin(115200); // Define serial monitor
  pinMode(LED, OUTPUT);
  digitalWrite(LED, LOW);

  Serial.println("Attempting to connect");

  while(WiFi.status() != WL_CONNECTED)
```

```

{
  WiFi.begin(ssid, password);
  for (int i=0; i<40; i++)
  {
    digitalWrite(LED, ledState);
    ledState = !ledState;
    delay(250);
  }
}
}

```

```
void loop()
```

```

{
  // if ((millis() - lastTime) > timerDelay) // gia delay metaksu tw n post an einai anagkaio
  //{
  // Connect or reconnect to WiFi
  if(WiFi.status() != WL_CONNECTED)
  {
    Serial.println("Attempting to connect");
    while(WiFi.status() != WL_CONNECTED)
    {
      WiFi.begin(ssid, password);
      for (int i=0; i<40; i++)
      {
        digitalWrite(LED, ledState);
        ledState = !ledState;
        delay(250);
      }
    }

    if (WiFi.status() == WL_CONNECTED)

```

```

{
  Serial.println("Connected.");
  digitalWrite(LED, HIGH);
}
}

if (Serial.available() > 0)
{
  // Read the incoming bytes:
  rlen = Serial.readBytes(str, 11);
  delay(100);
  // Print the received data
  Serial.print("STR-> ");

  for(int i=0; i<rlen; i++)
  {
    str[i];
    Serial.print(String(str[i])+" ");
  }

  x = 1;
  Serial.println("\nLength " +String(rlen));
}
delay(1000);

if(str[0] == 'U' && str[1] == 'T' && str[2] == 'C' && x == 1)
{
  Serial.println("UIC. Dummy json for last timestamp update internet connection");

  StaticJsonDocument<200> doc;
  String json;

```



```

doc["token"] = "bH5GxzFoj4";

serializeJson(doc, json);

http.begin(client, serverName);
http.addHeader("Content-Type", "application/json");
http.POST(json);
Serial.print(http.getString());
http.end(); //Close connection

//lastTime = millis(); // gia delay metaksu twn post an einai anagkaio
x=0;
str[11]='\0';
Serial.println();

}

else if(str[0] == 'E' && str[1] == 'V' && x == 1)
{
Serial.print("I received: ");

for(int i=2; i<rlen; i++)
{
if (str[i] == 'L')
{
leak = 1;
}

if (str[i] == 'O')
{
overflow = 1;
}
}
}

```

```

if (str[i] == 'S')
{
    smoke = 1;
}

if (str[i] == 'T')
{
    ev_temp = 1;
}

Serial.print(String(str[i])+" ");
}

Serial.println();

if (smoke == 1 && overflow == 0 && leak == 0) // energop aisthitira kapnou
{
    Serial.println("SMOKE");

    StaticJsonDocument<200> doc;
    String json;

    doc["token"] = "bH5GxzFoj4";
    doc["event"] = "Ενεργοποίηση αισθητήρα καπνού";

    serializeJson(doc, json);

    http.begin(client, serverName);
    http.addHeader("Content-Type", "application/json");
    http.POST(json);
}

```

```

Serial.print(http.getString());
http.end();
}

else if (overflow == 1 && smoke == 0 && leak == 0) // plhmmura lebhtostasiou
{
Serial.println("OVERFLOW");

StaticJsonDocument<200> doc;
String json;

doc["token"] = "bH5GxzFoj4";
doc["event"] = "Ενεργοποίηση αισθητήρα πλημμύρας χώρου";

serializeJson(doc, json);

http.begin(client, serverName);
http.addHeader("Content-Type", "application/json");
http.POST(json);
Serial.print(http.getString());
http.end();
}

else if (leak == 1 && overflow == 0 && smoke == 0) // diarroh deksamenhs
{
Serial.println("LEAK");

StaticJsonDocument<200> doc;
String json;

doc["token"] = "bH5GxzFoj4";
doc["event"] = "Ενεργοποίηση αισθητήρα διαρροής δεξαμενής";

```

```

serializeJson(doc, json);

http.begin(client, serverName);
http.addHeader("Content-Type", "application/json");
http.POST(json);
Serial.print(http.getString());
http.end();
}

else if (smoke == 1 && overflow == 1 && leak == 0)
{
  Serial.println("SMOKE and OVERFLOW");

  StaticJsonDocument<200> doc;
  String json;

  doc["token"] = "bH5GxzFoj4";
  doc["event"] = "Ενεργοποίηση αισθητήρα καπνού και αισθητήρα πλημμύρας χώρου";

  serializeJson(doc, json);

  http.begin(client, serverName);
  http.addHeader("Content-Type", "application/json");
  http.POST(json);
  Serial.print(http.getString());
  http.end();
}

else if (smoke == 1 && leak == 1 && overflow == 0)
{
  Serial.println("SMOKE and LEAK");

```

```

StaticJsonDocument<200> doc;
String json;

doc["token"] = "bH5GxzFoj4";
doc["event"] = "Ενεργοποίηση αισθητήρα καπνού και αισθητήρα διαρροής δεξαμενής";

serializeJson(doc, json);

http.begin(client, serverName);
http.addHeader("Content-Type", "application/json");
http.POST(json);
Serial.print(http.getString());
http.end();
}

else if (overflow == 1 && leak == 1 && smoke == 0)
{
Serial.println("OVERFLOW and LEAK");

StaticJsonDocument<200> doc;
String json;

doc["token"] = "bH5GxzFoj4";
doc["event"] = "Ενεργοποίηση αισθητήρα διαρροής δεξαμενής και αισθητήρα πλημμύρας
χώρου";

serializeJson(doc, json);

http.begin(client, serverName);
http.addHeader("Content-Type", "application/json");
http.POST(json);
Serial.print(http.getString());

```

```

    http.end();
}

else if (smoke == 1 && overflow == 1 && leak == 1)
{
    Serial.println("SMOKE and OVERFLOW and LEAK");

    StaticJsonDocument<200> doc;
    String json;

    doc["token"] = "bH5GxzFoj4";
    doc["event"] = "Ενεργοποίηση αισθητήρα καπνού, αισθητήρα πλημμύρας χώρου και αισθητήρα
διαρροής δεξαμενής";

    serializeJson(doc, json);

    http.begin(client, serverName);
    http.addHeader("Content-Type", "application/json");
    http.POST(json);
    Serial.print(http.getString());
    http.end();
}

if (ev_temp == 1)
{
    Serial.println("High Temperature");

    StaticJsonDocument<200> doc;
    String json;

    doc["token"] = "bH5GxzFoj4";
    doc["event"] = "Υπαρξη θερμοκρασίας στο λεβητοστάσιο μεγαλύτερη από 40οC";

```

```

serializeJson(doc, json);

http.begin(client, serverName);
http.addHeader("Content-Type", "application/json");
http.POST(json);
Serial.print(http.getString());
http.end();
}

//lastTime = millis(); // gia delay metaksu tw n post an einai anagkaio
smoke=0, leak=0, overflow=0, ev_temp=0, x=0;
Serial.println();
}

if (str[0] == 'T' && x==1) // thermokrasia ugrasia ekswterikou xwrou
{
Serial.print("I received: ");

for(int i=1; i<rlen; i++)
{
Serial.print(String(str[i])+" ");
}
Serial.println("\nlen "+String(rlen));

if (rlen == 7) //0.1 rlen 0-6+1
{
float aT = ((str[1])-48);
float zT = ((str[3])-48);
float bT = zT/10;
temp = aT+bT;

float aH = ((str[4])-48);

```

```
float zH = ((str[6])-48);
float bH = zH/10;
hum = aH+bH;
}
```

```
else if (rlen == 9) //20.4
```

```
{
float aT = ((str[1])-48)*10;
float bT = ((str[2])-48);
float zT = ((str[4])-48);
float cT = zT/10;
temp = aT+bT+cT;
```

```
float aH = ((str[5])-48)*10;
float bH = ((str[6])-48);
float zH = ((str[8])-48);
float cH = zH/10;
hum = aH+bH+cH;
}
```

```
else if (rlen == 8)
```

```
{
if (str[2] == '.')
{
float aT = ((str[1])-48);
float zT = ((str[3])-48);
float bT = zT/10;
temp = aT+bT;

float aH = ((str[4])-48)*10;
float bH = ((str[5])-48);
float zH = ((str[7])-48);
```



```

float cH = zH/10;
hum = aH+bH+cH;
}
else
{
float aT = ((str[1])-48)*10;
float bT = ((str[2])-48);
float zT = ((str[4])-48);
float cT = zT/10;
temp = aT+bT+cT;

float aH = ((str[5])-48);
float zH = ((str[7])-48);
float bH = zH/10;
hum = aH+bH;
}
}

StaticJsonDocument<200> doc;
String json;

doc["token"] = "bH5GxzFoj4";
doc["temp"] = temp;
doc["hum"] = hum;

serializeJson(doc, json);

HttpClient http;
http.begin(client, serverName);
http.addHeader("Content-Type", "application/json");
http.POST(json);
Serial.print(http.getString());

```

```

http.end();

// lastTime = millis(); // gia delay metaksu tw n post an einai anagkaio
x=0;
str[11]='\0';
Serial.println();
}

if (str[0] == 'T' && x==1) // thermokrasia ugrasia sto lebhtostasio
{
    Serial.print("T received: ");

    for(int i=1; i<rlen; i++)
    {
        Serial.print(String(str[i])+" ");
    }
    Serial.println("\nlen "+String(rlen));

    if (rlen == 7) //0.1 rlen 0-6+1
    {
        float aT = ((str[1])-48);
        float zT = ((str[3])-48);
        float bT = zT/10;
        temp = aT+bT;

        float aH = ((str[4])-48);
        float zH = ((str[6])-48);
        float bH = zH/10;
        hum = aH+bH;
    }

    else if (rlen == 9) //20.4

```

```

{
    float aT = ((str[1])-48)*10;
    float bT = ((str[2])-48);
    float zT = ((str[4])-48);
    float cT = zT/10;
    temp = aT+bT+cT;

    float aH = ((str[5])-48)*10;
    float bH = ((str[6])-48);
    float zH = ((str[8])-48);
    float cH = zH/10;
    hum = aH+bH+cH;
}

```

```

else if (rlen == 8)

```

```

{
    if (str[3] == '.') // TII.210.9
    {
        float aT = ((str[1])-48);
        float zT = ((str[3])-48);
        float bT = zT/10;
        temp = aT+bT;

        float aH = ((str[4])-48)*10;
        float bH = ((str[5])-48);
        float zH = ((str[7])-48);
        float cH = zH/10;
        hum = aH+bH+cH;
    }
}

```

```

else

```

```

{
    float aT = ((str[1])-48)*10;

```

```

float bT = ((str[2])-48);
float zT = ((str[4])-48);
float cT = zT/10;
temp = aT+bT+cT;

float aH = ((str[5])-48);
float zH = ((str[7])-48);
float bH = zH/10;
hum = aH+bH;
}
}

StaticJsonDocument<200> doc;
String json;

doc["token"] = "bH5GxzFoj4";
doc["temp_in"] = temp;
doc["hum_in"] = hum;

serializeJson(doc, json);

HttpClient http;
http.begin(client, serverName);
http.addHeader("Content-Type", "application/json");
http.POST(json);
Serial.print(http.getString());
http.end();

// lastTime = millis(); // gia delay metaksu tw n post an einai anagkaio
x=0;
str[11]='\0';
Serial.println();

```

```

}

else if (str[0] == 'D' && x==1) // metrshh deksamensh
{
    Serial.print("I received: ");

    for(int i=1; i<rlen; i++)
    {
        Serial.print(String(str[i])+" ");
    }
    Serial.println("\nlen "+String(rlen));

    if (rlen == 4) //0.1
    {
        float da = ((str[1])-48);
        float dz = ((str[3])-48);
        float db = dz/10;
        dist = da+db;
    }

    else if (rlen == 5) //10.1
    {
        float da = ((str[1])-48)*10;
        float db = ((str[2])-48);
        float dz = ((str[4])-48);
        float dc = dz/10;
        dist = da+db+dc;
    }

    else if (rlen == 6) //100.1
    {
        float da = ((str[1])-48)*100;

```

```

float db = ((str[2])-48)*10;
float dc = ((str[3])-48);
float dz = ((str[5])-48);
float dd = dz/10;
dist = da+db+dc+dd;
}

StaticJsonDocument<200> doc;
String json;

doc["token"] = "bH5GxzFoj4";
doc["dist"] = dist;

serializeJson(doc, json);

http.begin(client, serverName);
http.addHeader("Content-Type", "application/json");
http.POST(json);
Serial.print(http.getString());
http.end();

//lastTime = millis(); // gia delay metaksu twn post an einai anagkaio
x=0;
Serial.println();
}

else if (str[0] == 'f' && x==1) // metrshsh prin kai meta apo thn plhrwsh ths deksamenshs
{
Serial.print("I received: ");

for(int i=1; i<rln; i++)
{

```

```

    Serial.print(String(str[i])+" ");
}
Serial.println("\nlen "+String(rlen));

if (rlen == 4) //0.1
{
    float da = ((str[1])-48);
    float dz = ((str[3])-48);
    float db = dz/10;
    dist = da+db;
}

else if (rlen == 5) //10.1
{
    float da = ((str[1])-48)*10;
    float db = ((str[2])-48);
    float dz = ((str[4])-48);
    float dc = dz/10;
    dist = da+db+dc;
}

else if (rlen == 6) //100.1
{
    float da = ((str[1])-48)*100;
    float db = ((str[2])-48)*10;
    float dc = ((str[3])-48);
    float dz = ((str[5])-48);
    float dd = dz/10;
    dist = da+db+dc+dd;
}

JsonObject doc;

```

```

String json;

doc["token"] = "bH5GxzFoj4";
doc["f_dist_tank"] = dist;

serializeJson(doc, json);

http.begin(client, serverName);
http.addHeader("Content-Type", "application/json");
http.POST(json);
Serial.print(http.getString());
http.end();

//lastTime = millis(); // gia delay metaksu tw n post an einai anagkaio
x=0;
Serial.println();
}

else if (str[0] == 'a' && x==1) // vathmos apodoshs
{
    Serial.print("I received: ");

    for(int i=1; i<rlen; i++)
    {
        Serial.print(String(str[i])+" ");
    }
    Serial.println("\nlen "+String(rlen));

    // if (rstr[0] == 0) //99
    // {
    //     float da = ((str[1])-48);
    //     float dz = ((str[3])-48);

```



```

// float db = dz/10;
// dist = da+db;
// }

if (str[1] == '0') // 90
{
    int da = ((str[2])-48)*10;
    int db = ((str[3])-48);

    apod = da+db;
}

else //100.1
{
    int da = ((str[1])-48)*100;
    int db = ((str[2])-48)*10;
    int dc = ((str[3])-48);

    apod = da+db+dc;
}

StaticJsonDocument<200> doc;
String json;

doc["token"] = "bH5GxzFoj4";
doc["apod"] = apod;

serializeJson(doc, json);

http.begin(client, serverName);
http.addHeader("Content-Type", "application/json");
http.POST(json);

```

```
Serial.print(http.getString());  
http.end();  
  
//lastTime = millis(); // gia delay metaksu twn post an einai anagkaio  
x=0;  
Serial.println();  
}  
// } // gia delay metaksu twn post an einai anagkaio  
}
```

## ΠΑΡΑΡΤΗΜΑ Γ : Scripts για τη Βάση Δεδομένων

### *Δημιουργία βάσης δεδομένων*

```
show databases;
```

```
CREATE DATABASE users;
```

### *Δημιουργία πίνακα diamerismata*

```
CREATE TABLE diamerismata
```

```
(  
    id INT UNSIGNED PRIMARY KEY NOT NULL UNIQUE AUTO_INCREMENT,  
    h_id VARCHAR(50) NOT NULL,  
    wres VARCHAR(50) NOT NULL,  
    tank VARCHAR(50) NOT NULL,  
    part VARCHAR(50) NOT NULL,  
    date_created DATETIME NOT NULL  
);
```

### *Δημιουργία πίνακα eventsb*

```
CREATE TABLE eventsb
```

```
(  
    id INT UNSIGNED PRIMARY KEY NOT NULL UNIQUE AUTO_INCREMENT,  
    eventb VARCHAR(200) NOT NULL,  
    date_created DATETIME NOT NULL  
);
```

### *Δημιουργία πίνακα kpon*

```
CREATE TABLE kpon
```

```
(  
    id INT UNSIGNED PRIMARY KEY NOT NULL UNIQUE AUTO_INCREMENT,  
    temp VARCHAR(50) NOT NULL,  
    hum VARCHAR(50) NOT NULL,  
    temp_in VARCHAR(50) NOT NULL,
```

```
hum_in VARCHAR(50) NOT NULL,  
date_created DATETIME NOT NULL  
);
```

### ***Δημιουργία πίνακα upload\_files***

```
CREATE TABLE upload_files  
(  
    id INT UNSIGNED PRIMARY KEY NOT NULL UNIQUE AUTO_INCREMENT,  
    filename VARCHAR(50) NOT NULL,  
    category VARCHAR(50) NOT NULL,  
    filedata BLOB,  
    date_created DATETIME NOT NULL  
);
```

### ***Δημιουργία πίνακα users***

```
CREATE TABLE users  
(  
    id INT UNSIGNED PRIMARY KEY NOT NULL UNIQUE AUTO_INCREMENT,  
    username VARCHAR(30) NOT NULL,  
    email VARCHAR(30) NOT NULL,  
    password VARCHAR(100) NOT NULL,  
    first_name VARCHAR(50) NOT NULL,  
    last_name VARCHAR(50) NOT NULL,  
    floor TINYINT UNSIGNED NOT NULL,  
    house_number TINYINT UNSIGNED NOT NULL,  
    date_created DATETIME NOT NULL,  
    admin_key SMALLINT UNSIGNED  
);
```

## ΠΑΡΑΡΤΗΜΑ Δ : Κλάσεις για τη Βάση Δεδομένων

```
from Thesis_Build_Flask import my_db, login_manager
from datetime import datetime
from flask_login import UserMixin

from sqlalchemy.dialects.mysql import INTEGER, SMALLINT, TINYINT, MEDIUMBLOB,
LONGBLOB

@login_manager.user_loader
def load_user(users_id):
    return Users.query.get(int(users_id))

#UserMixin xreiazetai gia to login is authenticated.

class Users(my_db.Model, UserMixin):
    id = my_db.Column(INTEGER(unsigned=True), primary_key=True, unique=True, nullable=False)
    username = my_db.Column(my_db.String(30), unique=True, nullable=False)
    email = my_db.Column(my_db.String(30), nullable=False)
    password = my_db.Column(my_db.String(100), nullable=False)
    first_name = my_db.Column(my_db.String(50), nullable=False)
    last_name = my_db.Column(my_db.String(50), nullable=False)
    floor = my_db.Column(TINYINT(unsigned=True), nullable=False) #nunable not null
    house_number = my_db.Column(TINYINT(unsigned=True), nullable=False)
    date_created = my_db.Column(my_db.DateTime, default=datetime.now)
    admin_key = my_db.Column(SMALLINT(unsigned=True), nullable=False)
    announcements = my_db.relationship("New_announcement", backref="build_admin", lazy=True) #
sundesht sxeshs 1 pros polla

    #to lazy einai gia th stigmh pou fortwnetai o xrhsths na fortwnontai kai oles tis anakoinwseis autou
    tou xrhsth.

    # ara tha einai kapou sth mnhmh kai grhgora prosvasima

    #repr method. Gia na print px ena username. Den mporei praktina na tupwthei ena antikeimeno, den
    tha dwsei ta data
```

```

#h def repr antiproswepei to antikeimeno kai edw mpainei oti thelw na tupenetai sto print
def __repr__(self):
    return f"{self.username} : {self.email}" # {self.floor}

```

```

class New_announcement(my_db.Model, UserMixin):
    id = my_db.Column(INTEGER(unsigned=True), primary_key=True, unique=True, nullable=False)
    announc_title = my_db.Column(my_db.String(50), nullable=False)
    announc_main = my_db.Column(my_db.Text(), nullable=False)
    date_created = my_db.Column(my_db.DateTime, default=datetime.now)
    user_id = my_db.Column(INTEGER(unsigned=True), my_db.ForeignKey("users.id"),
nullable=False)

    # "user.id" o pinakas user kai to pedio id

    #thelei ton pinaka kai to onoma oxi thn klash kai to antikeimeno gi auto to user einai me mikro U
(oxi to modelo User dld)

```

```

class Upload_files(my_db.Model, UserMixin):
    id = my_db.Column(INTEGER(unsigned=True), primary_key=True, unique=True, nullable=False)
    filename = my_db.Column(my_db.String(50), nullable=False)
    category = my_db.Column(my_db.String(50), nullable=False)
    filedata = my_db.Column(LONGBLOB(), nullable=False)
    date_created = my_db.Column(my_db.DateTime, default=datetime.now)

```

```

class Eventsb(my_db.Model, UserMixin): #stathmh
    id = my_db.Column(INTEGER(unsigned=True), primary_key=True, unique=True, nullable=False)
    date_created = my_db.Column(my_db.DateTime, default=datetime.now)
    eventb = my_db.Column(my_db.String(200), nullable=False)
    def __repr__(self):
        return f"{self.date_created} : {self.eventb}"

```

```
class Kpon(my_db.Model, UserMixin):
    id = my_db.Column(INTEGER(unsigned=True), primary_key=True, unique=True, nullable=False)
    date_created = my_db.Column(my_db.DateTime, default=datetime.now)
    temp = my_db.Column(my_db.String(50), nullable=False)
    hum = my_db.Column(my_db.String(50), nullable=False)
    temp_in = my_db.Column(my_db.String(50), nullable=False)
    hum_in = my_db.Column(my_db.String(50), nullable=False)
```

```
class Diamerismata(my_db.Model, UserMixin):
    id = my_db.Column(INTEGER(unsigned=True), primary_key=True, unique=True, nullable=False)
    date_created = my_db.Column(my_db.DateTime, default=datetime.now)
    wres = my_db.Column(my_db.String(50), nullable=False)
    h_id = my_db.Column(my_db.String(50), nullable=False)
    tank = my_db.Column(my_db.String(50), nullable=False)
    part = my_db.Column(my_db.String(50), nullable=False)
```

## ΠΑΡΑΡΤΗΜΑ Ε : Κλάσεις για τις φόρμες

```
from flask_wtf import FlaskForm

from wtforms import StringField, IntegerField, BooleanField, SubmitField, TextAreaField, SelectField,
PasswordField

from wtforms.validators import DataRequired, NumberRange, Email, Length, EqualTo,
ValidationError

from Thesis_Build_Flask.my_db_models import Users

from flask_login import current_user

from Thesis_Build_Flask import my_encrypt

from flask_wtf.file import FileField, FileRequired

class SignupForm(FlaskForm):

    email = StringField(label="Email", render_kw={"placeholder": "Email"},
validators=[DataRequired(message="Υποχρεωτικό πεδίο"),
            Email(message="Λάθος mail")])

    #stringfield einai to input type="text" apo html

    username = StringField(label="Όνομα χρήστη", render_kw={"placeholder": "Username"},
validators=[DataRequired(message="Υποχρεωτικό πεδίο"),
            Length(min=5, max=20, message="Από 5 έως 15 χαρακτήρες")])

    password = PasswordField(label="Κωδικός", render_kw={"placeholder": "Password"},
validators=[DataRequired(message="Υποχρεωτικό πεδίο"),
            Length(min=5, max=20, message="Από 5 έως 15 χαρακτήρες")])

    confirm_password = PasswordField(label="Επιβεβαίωση κωδικού", render_kw={"placeholder":
"Confirm password"}, validators=[DataRequired(message="Υποχρεωτικό πεδίο"),
            Length(min=5, max=20, message="Από 5 έως 15 χαρακτήρες"),
            EqualTo("password", message='Να είναι ίδια'),])

    first_name = StringField(label="Όνομα", render_kw={"placeholder": "First name"},
validators=[DataRequired(message="Υποχρεωτικό πεδίο"),
```



```
Length(min=2, max=20, message="Από 2 έως 20 χαρακτήρες"))]
```

```
last_name = StringField(label="Επώνυμο", render_kw={"placeholder": "Last name"},  
validators=[DataRequired(message="Υποχρεωτικό πεδίο"),  
Length(min=2, max=50, message="Από 2 έως 50 χαρακτήρες")])
```

```
floor = IntegerField(label="Όροφος", render_kw={"placeholder": "Floor"},  
validators=[DataRequired(message="Υποχρεωτικό πεδίο"),  
NumberRange(min=0, max=12, message="2 χαρακτήρες  
όροφος")])
```

```
house_number = IntegerField(label="Αριθμός διαμερίσματος", render_kw={"placeholder":  
"House_id"}, validators=[DataRequired(message="Υποχρεωτικό πεδίο"),  
NumberRange(min=0, max=50, message="2 χαρακτήρες  
αριθμός διαμερ.")])
```

```
admin_key = StringField(label="Admin_Key", render_kw={"placeholder": "Admin_key"})
```

```
submit_btn = SubmitField("Αποστολή")
```

```
def validate_username(self, username):
```

```
    check_user = Users.query.filter_by(username=username.data).first()
```

```
    if check_user:
```

```
        raise ValidationError(f"To username <b>{check_user.username}</b> χρησιμοποιείται!")
```

```
def validate_email(self, email):
```

```
    check_email = Users.query.filter_by(email=email.data).first()
```

```
    if check_email:
```

```
        raise ValidationError(
```

```
            f"To email <b>{check_email.email}</b> χρησιμοποιείται!")
```

```

class UpdateAccountForm(FlaskForm):
    email = StringField(label="Email", validators=[DataRequired(message="Υποχρεωτικό πεδίο"),
        Email(message="Λάθος mail")])

    username = StringField(label="Όνομα χρήστη", validators=[DataRequired(message="Υποχρεωτικό
πεδίο"),
        Length(min=5, max=20, message="Από 5 έως 15 χαρακτήρες")])

    password = PasswordField(label="Νέος Κωδικός", render_kw={"placeholder": "new password"},
default="")

    first_name = StringField(label="Όνομα", validators=[DataRequired(message="Υποχρεωτικό
πεδίο"),
        Length(min=2, max=20, message="Από 2 έως 20 χαρακτήρες")])

    last_name = StringField(label="Επώνυμο", validators=[DataRequired(message="Υποχρεωτικό
πεδίο"),
        Length(min=2, max=50, message="Από 2 έως 50 χαρακτήρες")])

    floor = IntegerField(label="Όροφος", validators=[DataRequired(message="Υποχρεωτικό πεδίο"),
        NumberRange(min=0, max=12, message="2 χαρακτήρες όροφος")])

    house_number = IntegerField(label="Αριθμός διαμερίσματος",
validators=[DataRequired(message="Υποχρεωτικό πεδίο"),
        NumberRange(min=0, max=50,
            message="2 χαρακτήρες αριθμος διαμερ.")])

    admin_key = StringField(label="Admin_Key", render_kw={"placeholder": "admin_key"},)

    submit_btn = SubmitField("Αποστολή")

```

```

def validate_username(self, username):
    if username.data != current_user.username:
        check_user = Users.query.filter_by(username=username.data).first()
        if check_user:
            raise ValidationError(
                f"To username <b>{check_user.username}</b> χρησιμοποιείται!")

def validate_email(self, email):
    if email.data != current_user.email:
        check_user = Users.query.filter_by(email=email.data).first()
        if check_user:
            raise ValidationError(
                f"To username <b>{check_user.email}</b> χρησιμοποιείται!")

def validate_password(self, password):
    if password.data == "":
        x = Users.query.filter_by(id=current_user.id).first()
        y = x.password
        password.data = y
    else:
        if len(password.data) < 5 or len(password.data) > 21:
            raise ValidationError(f"Το password θα πρέπει να είναι από 5 έως 20 χαρακτήρες!")
        else:
            q = password.data
            password.data = my_encrypt.generate_password_hash(password.data).decode("utf-8")

def validate_admin_key(self, admin_key):
    if admin_key.data == "":
        x = Users.query.filter_by(id=current_user.id).first()

```

```
y = x.admin_key
admin_key.data = y
```

```
class LoginForm(FlaskForm):
    email = StringField(label="Email", render_kw={"placeholder": "Email"},
        validators=[DataRequired(message="Υποχρεωτικό πεδίο"),
            Email(message="Λάθος email")])

    password = PasswordField(label="Password", render_kw={"placeholder": "Password"},
        validators=[DataRequired(message="Υποχρεωτικό πεδίο")])

    remember_me = BooleanField(label="Να με θυμάσαι")

    submit_btn = SubmitField("Είσοδος")
```

```
class NewAnnouncementForm(FlaskForm):
    announc_title = StringField(label="Τίτλος ανακοίνωσης", render_kw={"placeholder": "Εισάγετε τον
        τίτλο της ανακοίνωσης"}, validators=[DataRequired(message="Υποχρεωτικό πεδίο"),
            Length(min=5, max=80, message="Από 5 έως 80 χαρακτήρες")])

    announc_main = TextAreaField(label="Ανακοίνωση", render_kw={"placeholder": "Εισάγετε το
        κείμενο της ανακοίνωσης"}, validators=[DataRequired(message="Υποχρεωτικό πεδίο"),
            Length(min=10, message="Ελάχιστο 10 χαρακτήρες")])

    submit_btn = SubmitField("Αποστολή")
```

```
class ContactForm(FlaskForm):
    contact_title = StringField(label="Τίτλος email", render_kw={"placeholder": "Εισάγετε τον τίτλο του
        email"}, validators=[DataRequired(message="Υποχρεωτικό πεδίο"),
            Length(min=5, max=80, message="Από 5 έως 80 χαρακτήρες")])
```

```
contact_main = TextAreaField(label="Κείμενο email", render_kw={"placeholder": "Εισάγετε το  
κείμενο του email"}, validators=[DataRequired(message="Υποχρεωτικό πεδίο"),  
Length(min=10, message="Ελάχιστο 10 χαρακτήρες")])
```

```
submit_btn = SubmitField("Αποστολή")
```

```
class ResetPasswEmailForm(FlaskForm):
```

```
email = StringField(label="Συμπληρώστε τη διεύθυνση email σας.", render_kw={"placeholder":  
"Email"}, validators=[DataRequired(message="Υποχρεωτικό πεδίο"),  
Email(message="Λάθος email!")])
```

```
submit_btn = SubmitField("Αποστολή")
```

```
class ResetPasswordForm(FlaskForm):
```

```
password = PasswordField(label="Κωδικός", render_kw={"placeholder": "password"},  
validators=[DataRequired(message="Υποχρεωτικό πεδίο"),  
Length(min=5, max=20, message="Από 5 έως 15 χαρακτήρες")])
```

```
confirm_password = PasswordField(label="Επιβεβαίωση κωδικού", render_kw={"placeholder":  
"confirm password"},
```

```
validators=[DataRequired(message="Υποχρεωτικό πεδίο"),  
Length(min=5, max=20, message="Από 5 έως 15 χαρακτήρες"),  
EqualTo("password", message='Να είναι ίδια'), ])
```

```
submit_btn = SubmitField("Αποστολή")
```

## ΠΑΡΑΡΤΗΜΑ ΣΤ : Αρχικοποιήσεις

```
from flask import Flask
from flask_sqlalchemy import SQLAlchemy
from flask_bcrypt import Bcrypt
from flask_login import LoginManager
from datetime import timedelta
from flask_mail import Mail
from itsdangerous import URLSafeTimedSerializer
from flask_apscheduler import APScheduler

class Config:
    SCHEDULER_API_ENABLED = True

app = Flask(__name__) # me to app trexei olh h efarmogh

app.config["SECRET_KEY"] = "thesis_aes_ihu_2811fd0723a59de5" # antiproswpeuei olh thn
efarmogh
app.config["WTF_CSRF_SECRET_KEY"] = "ihu_23_aes_1e5d746c54c0f3c2"
app.config["SQLALCHEMY_DATABASE_URI"] = "mysql://xxx:xxx@localhost:3306/flask_db_thesis?charset=utf8mb4"
app.config["REMEMBER_COOKIE_DURATION"] = timedelta(hours=1)
app.config["PERMANENT_SESSION_LIFETIME"] = timedelta(hours=1)
app.config["MAIL_SERVER"] = "smtp.gmail.com"
app.config["MAIL_PORT"] = 465
app.config["MAIL_USE_TLS"] = False
app.config["MAIL_USE_SSL"] = True
app.config["MAIL_USERNAME"] = "enoikos1adm@gmail.com"
app.config["MAIL_PASSWORD"] = "lrrfeaczbutawvr"
password_reset_serializer = URLSafeTimedSerializer(app.config["SECRET_KEY"])
app.config.from_object(Config())
```

```
mail = Mail(app)
```

```
my_db = SQLAlchemy(app) #sundeei thn efarmogh me th vash dedomenwn
```

```
my_encrypt = Bcrypt(app)
```

```
login_manager = LoginManager(app)
```

```
login_manager.login_view = "login" # an paw na kane nea anakoinwsh kai den exw sundethei me  
gurnaei
```

```
# sth login gia na sundethw den petaei 404 kai kanei redirect apo mono tou
```

```
login_manager.login_message_category = "warning" #gia xrwma sto mhnuma
```

```
login_manager.login_message = "Παρακαλώ συνδεθείτε"
```

```
scheduler = APScheduler()
```

```
scheduler.init_app(app)
```

```
scheduler.start()
```

```
from Thesis_Build_Flask import routes, my_db_models
```

## ΠΑΡΑΡΤΗΜΑ Ζ : Μέθοδοι (routes)

```
from flask import render_template, redirect, url_for, request, abort, flash, send_file, jsonify, Response,
make_response

from Thesis_Build_Flask import app, my_db, my_encrypt, mail, password_reset_serializer, scheduler

from Thesis_Build_Flask.valid_forms import SignupForm, LoginForm, NewAnnouncementForm,
UpdateAccountForm, ContactForm, \

    ResetPasswEmailForm, ResetPasswordForm

from Thesis_Build_Flask.my_db_models import Users, New_announcement, Upload_files, Eventsb,
Kpon, Diamerismata

from flask_login import login_user, current_user, login_required, logout_user

from flask_mail import Message

from io import BytesIO

import os

from werkzeug.utils import secure_filename # den uposthrizei ellhnika

from PIL import Image

from datetime import datetime, date, time

from sqlalchemy import and_, func

import pandas as pd

# auta ta arxeia xrhsimopoiountai apo tis formes
ALLOWED_EXTENSIONS = {'pdf', 'png', 'jpg', 'jpeg'}

app.config['MAX_CONTENT_LENGTH'] = 4 * 1024 * 1024 #4MB

@app.route('/signup/', methods=["GET", "POST"]) # signup
def signup():
    signup_form = SignupForm() # fortwnei olh th class kai to arxikopoiw se form
    if request.method == 'POST' and signup_form.validate_on_submit():
        email = signup_form.email.data # to data pairnei dedomena kai klhronomeitai apo to FlaskForm
        username = signup_form.username.data
        password = signup_form.password.data
        confirm_password = signup_form.confirm_password.data
```



```

first_name = signup_form.first_name.data
last_name = signup_form.last_name.data
floor = signup_form.floor.data
house_number = signup_form.house_number.data
admin_key = signup_form.admin_key.data
if admin_key == "":
    admin_key = 0 # to default otan dinw apo edw

encrypt_password = my_encrypt.generate_password_hash(password).decode("utf-8")

new_user = Users(username=username, email=email, password=encrypt_password,
first_name=first_name,
                last_name=last_name, floor=floor, house_number=house_number,
admin_key=admin_key)
my_db.session.add(new_user)
my_db.session.commit()
# neos user gia na kataxwrithe to encrypt_password tou password pou edwse o xrhsths
flash(f"Επιτυχής εγγραφή!", "success") #dinw to onoma ths kathgorias analoga me to xrwma apo
to bootstrap
# to flash proferei mhnymata sto xrhsth. Einai apo to Flask (vivliothiki flash) kai ta jinja templates
return redirect(url_for('login')) # sto url_for thelei to onoma ths methodou

return render_template('signup.html', signupform=signup_form)

# gia na leitourgisei swsta prepei na pernaei to form sth selida,arxika h forma tha einai kenh kai meta
tha parei ta data

#to session apothikeuetai sto server tou flask (server side)
@app.route('/', methods=["GET", "POST"]) # arxikh
@app.route('/login/', methods=["GET", "POST"]) # login
def login():
    login_form = LoginForm()

    if current_user.is_authenticated:

```

```

return redirect(url_for("index"))

#to of .. POST mporei na paralhfthei apo h html forma ta stelnei me POST
if request.method == 'POST' and login_form.validate_on_submit():
    email = login_form.email.data
    password = login_form.password.data
    remember_me = login_form.remember_me.data

    # to user einai antikeimeno
    user = Users.query.filter_by(email=email).first() # epistrefei lista an den uparxei user gurnaei
None

    # user.password h encrypt morfh
    if user and my_encrypt.check_password_hash(user.password, password) and (user.admin_key ==
12123):
        flash(f"Επιτυχής σύνδεση στο λογαριασμό σας!", "success")
        login_user(user, remember_me) # ftiaxnetai ena deuterio token apo to flask login gia na thumatai
to xrhsth
        # thn epomenh fora sta cookiew tha exei ena session kai ena remember token

        return redirect(url_for("index")) # den kserw giati to exw edw

elif user and my_encrypt.check_password_hash(user.password, password):
    flash(f"Επιτυχής σύνδεση στο λογαριασμό σας", "success")
    login_user(user, remember_me)
    return redirect(url_for("index"))

else:
    flash(f"Η σύνδεση στο λογαριασμό σας απέτυχε. Παρακαλώ προσπαθήστε πάλι!", "danger")

return render_template('login.html', loginform=login_form)

```

```

@app.route('/update_account/', methods=["GET", "POST"]) # update account
@login_required
def update_account():
    update_account_form = UpdateAccountForm(email=current_user.email,
        username=current_user.username,
            first_name=current_user.first_name,
            last_name=current_user.last_name, floor=current_user.floor,
            house_number=current_user.house_number)

    # ola einai pedia ths klash kai mporoun ma mpoun sthn parenthesh

    # to sqlalchemy mporei na katalavei poios einai o current user. opote porw na allaksw ta stoixeia
    # tou current user pou einai fortwmenos sto session
    if update_account_form.validate_on_submit():
        current_user.username = update_account_form.username.data
        current_user.email = update_account_form.email.data
        current_user.first_name = update_account_form.first_name.data
        current_user.last_name = update_account_form.last_name.data
        current_user.floor = update_account_form.floor.data
        current_user.house_number = update_account_form.house_number.data
        current_user.admin_key = update_account_form.admin_key.data
        current_user.password = update_account_form.password.data
        my_db.session.commit()

        flash(f"Επιτυχής ενημέρωση του λογαριασμού σας {current_user.username}!", "success")
        return redirect(url_for('index')) # sto url_for thelei to onoma ths methodou

    return render_template('update_account.html', updateaccount=update_account_form)

@app.route('/logout/') # logout
@login_required
def logout():
    logout_user() # kanei logout den xreiazetai pop opws sto session

```

```

flash(f"Επιτυχής αποσύνδεση!", "success")
return redirect(url_for("login"))

z = datetime.now()
_from = z.date()
_to = z.date()
from_today = z.date()
to_today = z.date()
my_list = []
@app.route('/index/', methods=["GET", "POST"])# arxikh ##kalo einai to route kai h methodos na
exoun idio onoma
@login_required
def index():
    global my_list, today, _from, _to, z, q, from_today, to_today, wres, date, sunolo_wres,
temp_value_id, wres_all\
        ,tank_diff

z = datetime.now()
_from = z.date()
_to = z.date()
from_today = z.date()
to_today = z.date()
wres = 0
sunolo_wres = 0
wres_all = 0
q = 0

try:
    my_list.clear() # katharismos ths listas

```

```

x = Users.query.filter_by(floor=current_user.floor,
house_number=current_user.house_number).first_or_404()

x = "h_id" + str(x.floor) + "_" + str(x.house_number)

#print(f" h_id: {x}")

#gia diasthma hmerwn
if request.form.get("from") or request.form.get("to"):
    _from = request.form.get("from")
    _to = request.form.get("to")
    from_today = _from
    to_today = _to
    range_date = pd.date_range(start=_from, end=_to) # euros timwn mesw tou panda

#diasthma hmeromhniwn
for i in range_date:
    #an oi hmeromhnies einai idies
    if _from == _to:
        q = Diamerismata.query.filter(func.date(Diamerismata.date_created) == from_today) \
            .filter_by(h_id=x).all()

        q_all = Diamerismata.query.filter(func.date(Diamerismata.date_created) ==
from_today).all()

        for i in q:
            wres = float(i.wres)
            sunolo_wres += wres # oi dikes tou wres
            date = str(i.date_created.time().strftime("%H:%M"))
            c = (wres, date)
            my_list.append(c)

        for i in q_all:
            if i.wres == "": # otan ginei eggraphh gia thn plhrwsh deksmanehs prin kai meta, oi wres
exoun keno

```

```

        i.wres = 0
        wres_all += float(i.wres) # sunolikes wres olwn twv diamerismatwn

#query gia diasthma hmerwn
else:
    q = Diamerismata.query.filter(func.date(Diamerismata.date_created) == i).filter_by(
        h_id=x).all() # tou xrhsth
    q_all = Diamerismata.query.filter(func.date(Diamerismata.date_created) == i).all() #oles oi
wres

    for i in q:
        wres += float(i.wres)
        date = str(i.date_created.date().strftime("%d-%m-%Y"))
        c = (wres, date)
        my_list.append(c)
        sunolo_wres += wres
        wres=0

    for i in q_all:
        if i.wres == "":
            i.wres = 0
            wres_all += float(i.wres)

# dunamika pername ta page_title, sunolo, pososto sta Jinja templates
page_title = "Διάγραμμα ωρών λειτουργίας θερμοστάτη"
sunolo = "Σύνολικές ώρες λειτουργίας: " + str(round(sunolo_wres, 2)) + " ώρες"
pososto = "Ποσοστό ωρών λειτουργίας επί του συνόλου: " \
        + str(round(((sunolo_wres / wres_all) * 100), 2)) + " %"

# mpainontas sthn arxikh selida kathe fortwnontai arxika ta dedomena ths trexousas hmeras apo to
else
else:
    q = Diamerismata.query.filter(func.date(Diamerismata.date_created) == from_today)\

```

```

.filter_by(h_id=x).all()

q_all = Diamerismata.query.filter(func.date(Diamerismata.date_created) == from_today).all()

for i in q:
    wres = float(i.wres)
    sunolo_wres += wres
    date = str(i.date_created.time().strftime("%H:%M"))
    c = (wres,date)
    my_list.append(c)

for i in q_all:
    if i.wres == "":
        i.wres=0
    wres_all += int(i.wres)

page_title = "Διάγραμμα ωρών λειτουργίας θερμοστάτη"
pososto = "Ποσοστό ωρών λειτουργίας επί του συνόλου: " \
    + str(round(((sunolo_wres / wres_all) * 100), 2)) + " %"
sunolo = "Σύνολική ώρες λειτουργίας: " + str(sunolo_wres) + " ώρες"

wres = [row[0] for row in my_list]
date = [row[1] for row in my_list]

return render_template('index.html', field1=date, field2=wres, from_today=from_today,
to_today=to_today,
    page_title=page_title, sunolo=sunolo, pososto=pososto) # emfanizei to html arxeio

except:
    return render_template('index.html', page_title="Υπήρξε κάποιο σφάλμα στο διάστημα
ημερομηνιών",
        from_today=from_today, to_today=to_today)

```

```

@app.route('/index_petrol/', methods=["GET", "POST"]) # sthathmh petrelaiou
@login_required
def index_petrol():
    global my_list, today, _from, _to, z, q, from_today, to_today, wres, date, sunolo_wres,
temp_value_id, wres_all\
        ,tank_diff, bef_t

    z = datetime.now()
    _from = z.date()
    _to = z.date()
    from_today = z.date()
    to_today = z.date()
    wres = 0
    sunolo_wres = 0
    wres_all = 0
    q = 0
    q_sun=0
    q_plh=0
    aft_bef_s=0
    bef=0
    aft=0
    bef_t=""
    bef_t1=""
    bef_t2=""
    aft_bef_l = []
    posoto=""
    page_title=""
    sunolo=""

    try:
        my_list.clear()

```



```
x = Users.query.filter_by(floor=current_user.floor,  
house_number=current_user.house_number).first_or_404()
```

```
x = "h_id" + str(x.floor) + "_" + str(x.house_number)
```

```
q_plhr = Diamerismata.query.with_entities(Diamerismata.tank, Diamerismata.date_created). \  
    filter(func.date(Diamerismata.date_created) == from_today).filter_by(  
    h_id="", wres="").all()
```

```
if request.form.get("from1") or request.form.get("to1"):
```

```
    _from = request.form.get("from1")
```

```
    _to = request.form.get("to1")
```

```
    from_today = _from
```

```
    to_today = _to
```

```
    range_date = pd.date_range(start=_from, end=_to)
```

```
    sum_lit = 0
```

```
    tank_diff = 0
```

```
    tank_temp_list = []
```

```
    tank_temp_date_list = []
```

```
    tank_temp_list.clear()
```

```
    tank_temp_date_list.clear()
```

```
    aft_bef_1.clear()
```

```
for i in range_date:
```

```
    if _from == _to:
```

```
        q = Diamerismata.query.with_entities(Diamerismata.tank, Diamerismata.date_created). \
```

```
            filter(func.date(Diamerismata.date_created) == from_today).filter_by(  
            h_id=x).all()
```

```
q_sun = Diamerismata.query.with_entities(Diamerismata.tank,  
Diamerismata.date_created). \
```

```
    filter(func.date(Diamerismata.date_created) == i).distinct().all()
```

```

#an to query ginei gia sugkekrimenh mera ekto ths trexousas
q_plhr = Diamerismata.query.with_entities(Diamerismata.tank,
Diamerismata.date_created). \
    filter(func.date(Diamerismata.date_created) == from_today).filter_by(
    h_id="", wres="").all()

if q_plhr != []:
    for i in q_plhr:
        bef = float(i.tank)
        aft_bef_l.append(bef)

    bef = aft_bef_l[0]
    aft = aft_bef_l[1]
    aft_bef_s = aft - bef
    bef_t = "Λίτρα πετρελαίου πριν και μετά την πλήρωση της δεξαμενής"
    bef_t1 = "πριν: " + str(bef) + " μετά: " + str(aft)
    bef_t2 = "Συνολικά λίτρα πλήρωσης: " + str(aft_bef_s)

for i in q:
    wres = float(i.tank)
    sunolo_wres += wres
    date = str(i.date_created.time().strftime("%H:%M"))
    c = (wres, date)
    my_list.append(c)

for i in q_sun:
    tank_temp_list.append(float(i.tank))

for i in range(len(tank_temp_list)-1):

```

```

if tank_temp_list[i] > tank_temp_list[i+1]:
    tank_diff += tank_temp_list[i] - tank_temp_list[i+1]

sum_lit = tank_diff

else:
    q = Diamerismata.query.with_entities(Diamerismata.tank, Diamerismata.date_created).\
        filter(func.date(Diamerismata.date_created) == i).distinct().all() # mono mia fora tis
times
#gia polla spitia tha ginetai pio grhgora h diadikasia

for i in q:
    tank_temp_list.append(float(i.tank))
    tank_temp_date_list.append(str(i.date_created.date().strftime("%d-%m-%Y")))

for i in range(len(tank_temp_list)-1):
    if tank_temp_list[i] > tank_temp_list[i+1]:
        tank_diff += tank_temp_list[i] - tank_temp_list[i+1]

tank_temp_list.clear()
date = tank_temp_date_list[0]
tank_temp_date_list.clear()
sum_lit += tank_diff
c = (tank_diff, date)
my_list.append(c)
tank_diff = 0

page_title = "Διάγραμμα κατανάλωσης λίτρων πετρελαίου"
sunolo = "Συνολική κατανάλωση λίτρων πετρελαίου: " + str(round(sum_lit,2)) + " λίτρα"
pososto = ""

```

```
temp_value_id = 0
```

```
else:
```

```
q = Diamerismata.query.with_entities(Diamerismata.tank, Diamerismata.date_created). \
    filter(func.date(Diamerismata.date_created) == from_today).filter_by(
    h_id=x).all()
```

```
if q_plhr != []:
```

```
    for i in q_plhr:
```

```
        bef = float(i.tank)
```

```
        aft_bef_l.append(bef)
```

```
    bef = aft_bef_l[0]
```

```
    aft = aft_bef_l[1]
```

```
    aft_bef_s = aft - bef
```

```
    bef_t = "Λίτρα πετρελαίου πριν και μετά την πλήρωση της δεξαμενής"
```

```
    bef_t1 = "πριν: " + str(bef) + " μετά: " + str(aft)
```

```
    bef_t2 = "Συνολικά λίτρα πλήρωσης: " + str(aft_bef_s)
```

```
for i in q:
```

```
    wres = float(i.tank)
```

```
    sunolo_wres += wres
```

```
    date = str(i.date_created.time()).strftime("%H:%M")
```

```
    c = (wres,date)
```

```
    my_list.append(c)
```

```
page_title = "Διάγραμμα κατανάλωσης λίτρων πετρελαίου"
```

```
sunolo = "Συνολική κατανάλωση λίτρων πετρελαίου: " + str(sunolo_wres) + " λίτρα"
```

```

pososto = ""

wres = [row[0] for row in my_list]
date = [row[1] for row in my_list]

return render_template('index_petrol.html', field1=date, field2=wres, from_today=from_today,
to_today=to_today,
                        page_title=page_title, sunolo=sunolo, pososto=pososto, bef_tit=bef_t, bfr=bef_t1,
                        sun=bef_t2)

except:
    return render_template('index_petrol.html', page_title="Υπήρξε κάποιο σφάλμα στο διάστημα
ημερομηνιών",
                            from_today=from_today, to_today=to_today)

z = datetime.now()
_from = z.date()
_to = z.date()
from_today = z.date()
to_today = z.date()
from_today_r = z.date()
to_today_r = z.date()
list_field1=[]
list_field2=[]
list_field3=[]
list_temp_in=[]
list_hum_in=[]
list_kpon=[]
range_list_kpon=[]
list_kpon_count = 0
temp_list_kpon_count = 0
range_sunolo_h_m_kpon=""

```

```

b_li=[]

@app.route('/temp_hum/', methods=["GET", "POST"])

@login_required

def temp_hum(): # thermokrasies-ugrasies otan leitourgei o lebhtas

    global list_kpon_count, today, _from, z, from_today, to_today, kpon_temp, kpon_hum, date,
list_field1, list_field2, \

        list_field3, a, b, c, d, ee, list_kpon, range_list_kpon, from_today_r, to_today_r,
range_sunolo_h_m_kpon, b_li

    z = datetime.now()

    _from = z.date()

    _to = z.date()

    from_today = z.date()

    to_today = z.date()

    from_today_r = z.date()

    to_today_r = z.date()

    kpon_temp=0

    kpon_hum=0

    kpon_temp_in=0

    kpon_hum_in=0

    q_temp_hum = 0

    list_kpon_count=0

    sun_min_kpon=0

    kpon_h=0

    kpon_m=0

    a=0

    b=0

    c=0

    d=0

    ee=0

    range_list_kpon_count=0

    range_sun_min_kpon=0

```

```
range_kpon_h=0
range_kpon_m=0
sunolo_h_m_kpon=""
range_sunolo_h_m_kpon=""
page_title=""
sunolo=""
```

```
try:
```

```
list_field1.clear()
list_field2.clear()
list_field3.clear()
list_kpon.clear()
list_temp_in.clear()
list_hum_in.clear()
range_list_kpon.clear()
b_li.clear()
if temp_value_id == 3:
```

```
    if request.form.get("from3"):
        _from = request.form.get("from3")
        from_today = _from
```

```
    q_temp_hum = Kpon.query.with_entities(Kpon.temp, Kpon.hum, Kpon.temp_in,
Kpon.hum_in, Kpon.date_created). \
        filter(func.date(Kpon.date_created) == from_today).all()
```

```
    for i in q_temp_hum:
        kpon_temp = float(i.temp)
        kpon_hum = float(i.hum)
        kpon_temp_in = float(i.temp_in)
        kpon_hum_in = float(i.hum_in)
```

```

date = str(i.date_created.time()).strftime("%H:%M")
r = int(i.date_created.time()).strftime("%M")
list_kpon.append(r)
list_field1.append(date)
list_field2.append(kpon_temp)
list_field3.append(kpon_hum)
list_temp_in.append(kpon_temp_in)
list_hum_in.append(kpon_hum_in)
k = str(i.date_created.date()).strftime("%d-%m-%Y")
b_li.append(k)

page_title = "Διάγραμμα ενεργοποίησης λέβητα"
page_title_up = "Ενεργοποίηση λέβητα"
sunolo = ""

a=list_field1
b=list_field2
c=list_field3
d=list_temp_in
ee=list_hum_in

for i in range(len(list_kpon)-1):
    if ((list_kpon[i+1]) - (list_kpon[i])) == 5:
        list_kpon_count += 1

sun_min_kpon = list_kpon_count*5
kpon_h = sun_min_kpon // 60
kpon_m = sun_min_kpon % 60
e = b_li[0]
sunolo_h_m_kpon = "Συνολικές ώρες λειτουργίας λέβητα στις " + str(e) + ": " \
    + str(kpon_h) + ":" + str(kpon_m)

```



```

# h timh 9 proerxetai apo to menou
elif temp_value_id == 9:

    page_title="Συνολικές ώρες λειτουργίας λέβητα"
    page_title_up = "Ωρες λειτουργίας λέβητα"

    if request.form.get("from4") or request.form.get("to4"):
        _from = request.form.get("from4")
        _to = request.form.get("to4")
        from_today_r = _from
        to_today_r = _to

        range_date = pd.date_range(start=from_today_r, end=to_today_r)

        for i in range_date:
            if from_today_r == to_today_r:
                q_temp_hum = Kpon.query.with_entities(Kpon.date_created). \
                    filter(func.date(Kpon.date_created) == from_today_r).count()
                x = Kpon.query.with_entities(Kpon.date_created). \
                    filter(func.date(Kpon.date_created) == from_today_r).all()
                range_list_kpon_count = q_temp_hum

                for i in x:
                    k = str(i.date_created.date().strftime("%d-%m-%Y"))
                    b_li.append(k)

            else:
                q_temp_hum = Kpon.query.with_entities(Kpon.date_created). \
                    filter(func.date(Kpon.date_created) == i).count()
                x = Kpon.query.with_entities(Kpon.date_created). \
                    filter(func.date(Kpon.date_created) == i).all()
                range_list_kpon_count += q_temp_hum

```

```

    for i in x:
        k = str(i.date_created.date().strftime("%d-%m-%Y"))
        b_li.append(k)

else:
    q_temp_hum = Kpon.query.with_entities(Kpon.date_created). \
        filter(func.date(Kpon.date_created) == from_today_r).count()
    x = Kpon.query.with_entities(Kpon.date_created). \
        filter(func.date(Kpon.date_created) == from_today_r).all()
    range_list_kpon_count = q_temp_hum

    for i in x:
        k = str(i.date_created.date().strftime("%d-%m-%Y"))
        b_li.append(k)

range_sun_min_kpon = range_list_kpon_count * 5

range_kpon_h = range_sun_min_kpon // 60
range_kpon_m = range_sun_min_kpon % 60
f = b_li[0]
g = b_li[-1]
if from_today_r == to_today_r:
    range_sunolo_h_m_kpon = "Συνολικές ώρες λειτουργίας λέβητα στις " + \
        str(f) + ": " + str(range_kpon_h) + ":" + str(range_kpon_m)
else:
    range_sunolo_h_m_kpon = "Συνολικές ώρες λειτουργίας λέβητα μεταξύ " \
        + str(f) + " ~ " + str(g) \
        + ": " + str(range_kpon_h) + ":" + str(range_kpon_m)

```

```

    return render_template('temp_hum.html', field1=a, field2=b, field3=c, field4=d, field5=ee,
from_today=from_today,

        from_today_r=from_today_r, to_today_r=to_today_r,

        page_title=page_title, page_title_up=page_title_up, sunolo=sunolo_h_m_kpon,

        range_sunolo=range_sunolo_h_m_kpon,                                max=100,
temp_value_id=temp_value_id)

    except:

        return render_template('temp_hum.html', page_title="Υπήρξε κάποιο σφάλμα στο διάστημα
ημερομηνιώνΑ",

            from_today=from_today)

```

```

@app.route('/total_h_hours/', methods=["GET", "POST"]) # sunolies wres diamerismatwn

@login_required

def total_h_hours():

    global my_list, today, _from, _to, z, q, from_today, to_today, wres, date, sunolo_wres,
temp_value_id, wres_all\

        ,tank_diff

    z = datetime.now()

    _from = z.date()

    _to = z.date()

    from_today = z.date()

    to_today = z.date()

    wres = 0

    sunolo_wres = 0

    wres_all = 0

    q = 0

    total_h_id1_1 = 0

    total_h_id1_2 = 0

    total_h_id2_3 = 0

    try:

        my_list.clear()

```

```

# ta h_id twn diamerismatwn
x = ["h_id1_1", "h_id1_2", "h_id2_3"]

if request.form.get("from5") or request.form.get("to5"):
    _from = request.form.get("from5")
    _to = request.form.get("to5")
    from_today = _from
    to_today = _to
    range_date = pd.date_range(start=from_today, end=to_today)

    if from_today == to_today:

        for h_id in x:
            for i in range_date:
                q = Diamerismata.query.filter(func.date(Diamerismata.date_created)
from_today).filter_by(
                    h_id=h_id).all()

                for b in q:
                    wres = float(b.wres)
                    sunolo_wres += wres

                my_list.append(round(sunolo_wres,2))

                sunolo_wres=0
                wres=0

    else:
        wres = 0
        sunolo_wres = 0
        for h_id in x:
            for i in range_date:
                q = Diamerismata.query.filter(func.date(Diamerismata.date_created)
i).filter_by(h_id=h_id).all()

```

```

        for b in q:
            wres = float(b.wres)
            sunolo_wres += wres
        my_list.append(round(sunolo_wres,2))
        wres = 0
        sunolo_wres = 0

else:
    for h_id in x:
        q = Diamerismata.query.filter(func.date(Diamerismata.date_created) == from_today)\
            .filter_by(h_id=h_id).all()

        for i in q:
            wres = float(i.wres)
            sunolo_wres += round(wres,2) # strogulopoihsh se ena dekadiko
        c = sunolo_wres
        my_list.append(round(sunolo_wres,2))
        sunolo_wres=0
        wres=0

page_title = "Συνολικές ώρες λειτουργίας ενοίκων"

diamer = ["Διαμέρισμα 1_1" , "Διαμέρισμα 1_2", "Διαμέρισμα 2_3"]
li=[]

for i in range(len(diamer)):
    a = (diamer[i], my_list[i])
    li.append(a)
sunolo=li

return render_template('total_h_hours.html', from_today=from_today, to_today=to_today,

```

```

        page_title=page_title, sunolo=sunolo, diamer=diamer) # emfanizei to html arxeio

    except:

        return render_template('total_h_hours.html', page_title="Υπήρξε κάποιο σφάλμα στο διάστημα
ημερομηνιών",

            from_today=from_today, to_today=to_today)

h = 0
d = 0
c = 0
distance = 0
part = 0
list_for_db = []
apodosh=0
tempr_flag = 0
tempr_in_flag = 0
event_flag = 0
temp=0
hum=0
temp_in=0
hum_in=0
check_uic_h = ""
check_uic_m = ""

@app.route('/device_post/',methods=["POST"])
def device_post():

    global h, d, c, distance, part, h_id1_1, h_id1_2, h_id2_3, check_uic_h, check_uic_m, apodosh,
list_for_db, \

        temp, hum, temp_in, hum_in, tempr_flag, tempr_in_flag, event_flag

    data = request.get_json()

    token = data["token"]

    if token == 'bH5GxzFoj4':

```

```

check_uic_h = datetime.now().hour
check_uic_m = datetime.now().minute

# apodosh tou lebhta
if "apod" in data:
    apodosh = data["apod"]

if "f_dist_tank" in data:
    t_d = round(data["f_dist_tank"], 2) #1 dekadiko
    distance = t_d*10 #gia thn prosomoiwsh
    distance = round(((distance*1500)/150), 2) #150εκ = 1500λιτρα
    #print(f"Μέτρηση δεξαμενής πριν και μετά την πλήρωση {distance} λίτρα")
    dh = Diamerismata(h_id="", wres="", tank=distance, part="", date_created=datetime.now())

    my_db.session.add(dh)
    my_db.session.commit()

if "dist" in data:
    t_d = round(data["dist"], 2) #1 dekadiko
    distance = t_d*10 #gia thn proswmoiwsh
    distance = round(((distance*1500)/150), 2) #150εκ = 1500λιτρα
    #print(f"Λίτρα δεξαμενής= {distance} λίτρα")
    d = 1 # flaf se "1"

if "h_id1_1" and "h_id1_2" and "h_id2_3" in data:
    list_for_db.clear()
    h_id1_1_wres = data["h_id1_1"]
    h_id1_2_wres = data["h_id1_2"]
    h_id2_3_wres = data["h_id2_3"]
    list_for_db.append(h_id1_1_wres)
    list_for_db.append(h_id1_2_wres)
    list_for_db.append(h_id2_3_wres)

```

```

        #print(f"Όρες: Διαμέρισμα 1_1= {h_id1_1_wres}, Διαμέρισμα 1_2= {h_id1_2_wres},
        Διαμέρισμα 2_3= {h_id2_3_wres}")

        h = 1 # flag se "1"

# otan kai oi 2 shmaiees ginoun "1" ginetai heggraph sth vash dedomenwn
if d == 1 and h == 1 :
    c += 1
    part = str(c) + '/4' # gia na grafei kai poia eggraph apo tis 4 egine
    if c == 4:
        c = 0

for x, y in data.items(): # epeidh oi wres erxontai apo to postman tis pernow kateutheian kai edw
    if x != "token" and x != "dist": # otan erxontai mazi
        dh = Diamerismata(h_id = x, wres = y, tank = distance, part = part, date_created =
datetime.now())
        my_db.session.add(dh)
        my_db.session.commit()

#mhdenismos flags
d = 0
h = 0

if "temp" and "hum" in data:
    temp = round(data["temp"], 2)
    hum = round(data["hum"], 2)
    tempr_flag = 1

if "temp_in" and "hum_in" in data:
    temp_in = round(data["temp_in"], 2)
    hum_in = round(data["hum_in"], 2)
    tempr_in_flag = 1

if tempr_flag == 1 and tempr_in_flag == 1:

```



```

th = Kpon(temp = temp, hum = hum, temp_in = temp_in, hum_in = hum_in, date_created =
datetime.now())

my_db.session.add(th)

my_db.session.commit()

#print(f"Θερμοκρασία εκτός: {temp} - Υγρασία εκτός= {hum} ### Θερμοκρασία εντός:
{temp_in} - Υγρασία εντός= {hum_in}")

temp_in_flag = 0

temp_flag = 0

if "event" in data:
    event = data["event"]
    #print(f"Συμβάν= {event}")
    ev = Eventsb(eventb = event, date_created = datetime.now())
    my_db.session.add(ev)
    my_db.session.commit()
    event_flag=1
    body=event

#apostolh email gia sumvan h xamhllh statmh deksmamenhs
if distance <= 15.3 or event_flag == 1:
    with scheduler.app.app_context():
        if event_flag == 1:
            admin = Users.query.all() # se olous
            event_flag=0
            subject = "Ενεργοποίηση αισθητήρα-ων ασφαλείας στο λεβητοστάσιο"
        else:
            admin = Users.query.filter_by(admin_key=12123).all() # mono ston/stous admin
            subject = "Ανάγκη πλήρωσης δεξαμενής πετρελαίου"
            body="Το πετρέλαιο της δεξαμενής είναι λιγότερο από 150 λίτρα. Παρακαλώ επιληφθείτε
περί " \
                "της πλήρωσης δεξαμενής."

for a in admin:

```

```

        msg = Message(subject=subject,
                      sender=app.config.get("MAIL_USERNAME"),
                      recipients=[a.email],
                      body=body)
        mail.send(msg)

# an einai lathos to token
else:
    return "Unauthorized"

return "Success"

@app.route('/announcements/') # anakoinwseis
@login_required
def announcements():
    page = request.args.get("page", 1, type=int) # an dwsw kati pou den isxuei by default paei sthn 1
    #selidopoihsh stis anakoinwseis

    announcements =
New_announcement.query.order_by(New_announcement.date_created.desc()).paginate(per_page=5,page=page)

    # h lista me oles tis anakoinwseis pernaei sto template mesw tou tou announcements

    return render_template('announcements.html', announcements=announcements) #
announcements.html

@app.route('/new_announcement/', methods=["GET", "POST"]) # dhmiourgia neas anakoinwshs
@login_required
def new_announcement():
    new_announc_form = NewAnnouncementForm()

    if request.method == 'POST' and new_announc_form.validate_on_submit():

```

```

    annc_title = new_announc_form.announc_title.data
    annc_main = new_announc_form.announc_main.data

    new_full_announc = New_announcement(announc_title=annc_title, announc_main=annc_main,
user_id=current_user.id)
    my_db.session.add(new_full_announc)
    my_db.session.commit()

    flash(f'Επιτυχής δημοσίευση της ανακοίνωσης', "success")
    return redirect(url_for("announcements"))

return render_template('new_announcement.html', new_annc_form=new_announc_form,
                        page_title="Νέα ανακοίνωση")

@app.route('/edit_announcement/<int:announc_id>', methods=["GET", "POST"]) # epeksergia
anakoinwshs
@login_required
def edit_announcement(announc_id):
    edit_announc = New_announcement.query.filter_by(id=announc_id,
build_admin=current_user).first_or_404()

    edit_announc_form = NewAnnouncementForm(announc_title=edit_announc.announc_title,
announc_main=edit_announc.announc_main)

    if request.method == 'POST' and edit_announc_form.validate_on_submit():
        edit_announc.announc_title = edit_announc_form.announc_title.data
        edit_announc.announc_main = edit_announc_form.announc_main.data

        my_db.session.commit() # me add prosthetw sth vash me commit kanw update ta uparxonta
dedomena

        flash(f'Επιτυχής επεξεργασία της ανακοίνωσης', "success")
        return redirect(url_for("announcements"))

```

```

return render_template('new_announcement.html', new_annnc_form=edit_announc_form,
                      page_title="Τροποποίηση ανακοίνωσης")

@app.route("/delete_announcement/<int:announc_id>", methods=["GET", "POST"]) #diagrafh
anakoinwshs
@login_required
def delete_announcement(announc_id):
    announcement = New_announcement.query.filter_by(id=announc_id,
build_admin=current_user).first_or_404()

    # me to filter by elegxo kai oti o current user einai autos ths anakoinwshs. auto ginetai gia na mporei
o users

    # na diagrafei mono tis dikes tou anakoinwseis

    if announcement:
        my_db.session.delete(announcement)
        my_db.session.commit()

        flash("Η ανακοίνωση διαγράφηκε με επιτυχία.", "success")
        return redirect(url_for("announcements"))

    flash("Η ανακοίνωση δε βρέθηκε.", "warning")
    return redirect(url_for("announcements"))

# plhrhs anakoinwshs kai to id gia na kresei poia anakoinwsh na deiksei
@app.route('/full_announcement/<int:announc_id>', methods=["GET"])
@login_required
def full_announcement(announc_id):
    full_announcement = New_announcement.query.filter_by(id=announc_id).first_or_404()
    return render_template('full_announcement.html', full_announcement=full_announcement)

```

```
@app.route('/build_admin_announc/<int:build_admin_id>', methods=["GET"]) # oi anakoinwseis enos diaxeiristh
```

```
@login_required
```

```
def build_admin_announc(build_admin_id): # oles oi anakoinwseis tou enos admin
```

```
    user = Users.query.get_or_404(build_admin_id)
```

```
    page = request.args.get("page", 1, type=int)
```

```
    announc_by_build_admin = New_announcement.query.filter_by(build_admin=user).order_by(
        New_announcement.date_created.desc()).paginate(per_page=5, page=page)
```

```
    return render_template('build_admin_annc.html',
        announcs_by_build_admin=announc_by_build_admin, build_admin=user)
```

```
temp_value_id=0
```

```
@app.route('/history/<int:value_id>', methods=["GET", "POST"]) # istoriko
```

```
@login_required
```

```
def history(value_id):
```

```
    global temp_value_id
```

```
    temp_value_id = value_id # h timh apo to menou gia kathe epilogh
```

```
    page = request.args.get("page", 1, type=int)
```

```
    if value_id == 1:
```

```
        history = Eventsb.query.paginate(per_page=50, page=page)
```

```
        title = 'Ιστορικό συμβάντων'
```

```
        return render_template('people.html', people=history, page_title=title, val_id=temp_value_id)
```

```
    elif value_id == 2:
```

```
        return redirect(url_for("index_petrol")) # anakateuthunsh sth methodo index_petrol
```

```
    elif value_id == 3 or value_id == 9:
```

```
        return redirect(url_for("temp_hum")) # anakateuthunsh sth methodo temp_hum
```

```

elif value_id == 4 or value_id == 5 or value_id == 6 or value_id == 7 or value_id == 8:
    if value_id == 4:
        x = 'deh' # category gia thn eggraph sth vash dedomenwn
        title = 'Ιστορικό λογαριασμών ρεύματος' # o titlos metaferetai dunamika
    elif value_id == 5:
        x = 'petrol'
        title = 'Ιστορικό λογαριασμών πετρελαίου'
    elif value_id == 6:
        x = 'fuel_tank'
        title = 'Ιστορικό πλήρωσης δεξαμενής πετρελαίου'
    elif value_id == 7:
        x = 'koinox'
        title = 'Ιστορικό κοινοχρήστων'
    elif value_id == 8:
        x = 'alloi'
        title = 'Ιστορικό λοιπών λογαριασμών'

    page = request.args.get("page", 1, type=int)
    history = Upload_files.query.filter_by(category=x).order_by(Upload_files.date_created.desc()).paginate(
        per_page=5, page=page) # selidopoihsh otan gientai to erwthmas th vash

    return render_template('people.html', people=history, page_title=title, val_id=temp_value_id)
    return render_template('history.html', history=history, page=page, value_id=temp_value_id,
        page_title=title,
        temp_value_id=temp_value_id)

@app.route('/contact_form/', methods=["GET", "POST"]) # forma epikoinwnias metaksu diaxeiristwn
kai enoikwn
@login_required
def contact_form():
    contact_form = ContactForm() # h klash ContactForm

```

```

if current_user.admin_key == 12123:
    page_title = 'ένοικους'
else:
    page_title = 'διαχειριστές'

if request.method == 'POST' and contact_form.validate_on_submit():
    con_title = contact_form.contact_title.data
    con_main = contact_form.contact_main.data
    if current_user.admin_key == 12123:
        admin = Users.query.filter(Users.id != current_user.id).all() # se olous ekτος ton eauto tou

    else:
        admin = Users.query.filter_by(admin_key=12123).all() # mono ston/stous admin

    with app.app_context():
        for a in admin:
            msg = Message(subject=con_title,
                          sender=app.config.get("MAIL_USERNAME"),
                          recipients=[a.email],
                          body=con_main)
            mail.send(msg)
            admin = Users.query.filter_by(admin_key=12123).all() # mono ston/stous admin

    flash(f"Επιτυχής αποστολή email", "success")
    return redirect(url_for("index"))

return render_template('contact_form.html', contact_form=contact_form, page_title=page_title)

@app.route('/upload_files/', methods=["GET", "POST"]) # anevasma arxeiou
@login_required # gia na mpei mono an exei ginei login alla douleuei gia ola ta login

```

```

def upload_files():
    test = Upload_files()

def allowed_file(filename):
    return '.' in filename and \
        filename.rsplit('.', 1)[1].lower() in ALLOWED_EXTENSIONS

if request.method == 'POST':

    file = request.files['file']
    if file.filename == "":
        flash('Παρακαλώ επιλέξτε αρχείο', "warning")
    value = request.form.get('timologia')
    if value == "":
        flash('Παρακαλώ επιλέξτε φάκελο', "warning")

    if file and allowed_file(file.filename) and value != "":

        filename = file.filename
        x, y = os.path.splitext(file.filename)

        if y == ".jpeg" or y == ".jpg" or y == ".png" or y == ".pdf":
            value = request.form.get('timologia') # se poio fakelo tha apothikeutei
            upload = Upload_files(filename=filename, filedata=file.read(), category=value)
            my_db.session.add(upload)
            my_db.session.commit()
            flash(f"Το αρχείο {file.filename} ανέβηκε με επιτυχία.", "success")
            return redirect(url_for("announcements"))

return render_template('upload_files.html')

```



```

@app.errorhandler(413)
def request_entity_too_large(error):
    flash(f"Το αρχείο υπερβαίνει τα 4MB.", "warning")
    return redirect(url_for("upload_files"))

@app.route('/download_files/<int:upload_id>') # katevasma arxeiou
@login_required
def download_files(upload_id):
    upload = Upload_files.query.filter_by(id=upload_id).first()
    return send_file(BytesIO(upload.filedata), download_name=upload.filename, as_attachment=True)

@app.route('/people/') # emfanish kapoiwn stoixeiwn twn enoikwn sthn efarmogh
@login_required # gia na mpei mono an exei ginei login alla douleuei gia ola ta login
def people():
    page = request.args.get("page", 1, type=int)
    people = Users.query.paginate(per_page=25, page=page)
    title="Στοιχεία ενοίκων"
    return render_template('people.html', people=people, page_title=title)

@app.route("/reset_password/", methods=["GET", "POST"])
def reset_password():
    reset_passw_email_form = ResetPasswEmailForm()
    if reset_passw_email_form.validate_on_submit():
        user = Users.query.filter_by(email=reset_passw_email_form.email.data).first()
        if user:
            try:
                msg = Message(subject="Reset Password link",
                               sender=app.config.get("MAIL_USERNAME"),
                               recipients=[user.email])

```

```

link = url_for('reset_password_token',
              token=password_reset_serializer.dumps(user.email, salt='7c2189ed0c264d49'),
              _external=True) # fortwnetai sto link to sugkekrimeno salt
msg.body = "Your link is {}".format(link) # thelei "" to body
mail.send(msg)

flash(f"Έχει αποσταλεί στο email <b>{user.email}</b> ο σύνδεσμος αλλαγής του κωδικού
σας.", "success")

return redirect(url_for("login"))

except:

    flash("Αποτυχία αποστολής του email. Παρακαλώ προσπαθήστε ξανά", "warning")

    return redirect(url_for("reset_password"))

else:

    flash(f"Το email δεν βρέθηκε</b>", "warning")

    return redirect(url_for("reset_password"))

return render_template("reset_passw_email.html", reset_passw_email=reset_passw_email_form)

@app.route("/reset_password/<token>", methods=["GET", "POST"])
def reset_password_token(token):

    try:

        email = password_reset_serializer.loads(token, salt='7c2189ed0c264d49', max_age=1800) #
max_age σε seconds

        #elegxetai an uparxei to salt

    except:

        flash("The password reset link is invalid or has expired.", 'error')

        return redirect(url_for('login'))

reset_passw_form = ResetPasswordForm()

if reset_passw_form.validate_on_submit():

    user = Users.query.filter_by(email=email).first_or_404()

    user.password

my_encrypt.generate_password_hash(reset_passw_form.password.data).decode("utf-8")

```

```

my_db.session.commit()

flash('Επιτυχής αλλαγή κωδικού πρόσβασης!', 'success')

return redirect(url_for('login'))

return render_template('reset_password.html', token=token, reset_passw_form=reset_passw_form)

check_device_min = 0
diff_minutes = 0
@scheduler.task("interval", id='check_device', minutes=5)
def interval_task(): # elegxos an h suskeuh vrisketai entos h ektos leitourgias
    global check_device_min, diff_minutes, check_uic_h, check_uic_m
    check_device_min += 1
    #print(f"Counter check_device_min {check_device_min}")

    #apothikeuontai oi teleutaies times pou egine to post
    last_check_h = check_uic_h
    last_check_m = check_uic_m

    #oi trexouses xronikes times sth methodo
    current_check_h = datetime.now().hour
    current_check_m = datetime.now().minute

    if check_device_min == 2: # 10 lepta
        check_device_min = 0

    #otan oi times einai apo wra se wra, px 11:57 kai 12:03
    if current_check_h != last_check_h:
        last_check_m = 59 - int(last_check_m)
        diff_minutes = last_check_m + current_check_m
    else:
        diff_minutes = current_check_m - last_check_m

```

```

if diff_minutes <= 10:
    pass
    #print(f"H suskeuh einai OK. Ekane POST se xrono mikrotero twn 10 leptwn")
else:
    #print(f"H suskeuh einai EKTOS SYNDESHS. To teleutatio POST egine to ligotero prin 10
lepta")

with scheduler.app.app_context():
    admin = Users.query.filter_by(admin_key=12123).all() # mono ston/stous admin
    for a in admin:
        #print(f" id: {a.id} email: {a.email}")
        msg = Message(subject="Device - No internet connection",
            sender=app.config.get("MAIL_USERNAME"),
            recipients=[a.email],
            body="Η συσκευή καταγραφής είναι αποσυνδεδεμένη από το internet "
                "για περισσότερο από 10 λεπτά. Παρακαλώ ελέγξτε και αποκαταστήστε τη
σύνδεση.")
        mail.send(msg)

```

## ΠΑΡΑΡΤΗΜΑ Η : HTML αρχεία

-----base\_html.html-----

```
<!--O vasikos kormos tis html pou den allazei se kamia selida-->
<!DOCTYPE html>
<html lang="en" xmlns="http://www.w3.org/1999/html">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>
    {% block title %}
    {% endblock title %}
  </title>

  <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha1/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-
GLhLTQ8iRABdZLI6O3oVMWSktQOp6b7In1Zl3/Jr59b6EGGoI1aFkw7cmDA6j6gD"
crossorigin="anonymous"> <!--link gia sundesh me to bootstrap-->

  <link rel="stylesheet" type="text/css" href="{{ url_for('static', filename= 'css/my_styles.css')}}"> <!--fortwnei to css arxeio me urr_for. To static einai to default pou prepei na vlepei to flask, to path to dinw sto filename-->

  <link rel="shortcut icon" href="{{ url_for('static', filename= 'images/favicon.png')}}">

  <script src="https://cdnjs.cloudflare.com/ajax/libs/Chart.js/2.5.0/Chart.min.js"></script>

</head>

<body>
```

```

<div class="my_container">
  <div class="my_container_out_div">
    <div class="my_container_div">
      <header>
        <div class="pic">
          <a href="/index/">
             <!--allazei megethos kathos allazoume tis selides-->
          </a>
        </div>

        {% include "navbar.html" %} <!--prostheti th selida navbar.html (einai to menu)-->

      </header>

      <main class="">
        <div class="flash_msg">
          <div>
            {% with messages = get_flashed_messages(with_categories=True) %}
              {% if messages %}
                {% for categ,msg in messages %} <!-- an periexei polla mhnumata -->
                  <div class="alert alert-{{ categ }} alert-dismissible fade show" role="alert">
                    {{ msg | safe }} {# thelei kai | safe gia na deixnei swsta ta tags tou flash px
to usernae ths eggrafhs xwris ta html tags an ta xrhsimopoiw mesa sto flash#}
                    <button type="button" class="btn-close" data-bs-dismiss="alert" aria-
label="Close"></button>
                  </div>
                {% endfor %}
              {% endif %}
            {% endwith %}
          </div>
        </div>
      </main>
    </div>
  </div>
</div>

```

```

    {% block body % } <!--tha periexei to body tis kathe selidas-->

    {% endblock body % }

</main>

<footer>
    {% include "footer.html" % }
</footer>

    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-
alpha1/dist/js/bootstrap.bundle.min.js" integrity="sha384-
w76AqPfDkMBDXo30jS1Sgez6pr3x5MlQ1ZAGC+nuZB+EYdgRZgiwxhTBTkF7CXvN"
crossorigin="anonymous"></script> <!--link gia sundesh me to bootstrap-->

    <script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/2.9.2/umd/popper.min.js"></script>

    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.3/jquery.min.js"></script>

</div>

</div>

</div>

</body>
</html>

-----announcements.html-----

{% extends "base_html.html" % } <!--epekteinei oti exei h base_html kai tis prostethei epipleonoti einai
apo katw-->

```

```
{% block title % }
```

```
Ανακοινώσεις
```

```
{% endblock title % }
```

```
{% block body % }
```

```
<div class="announc_title">
```

```
<div>
```

```
<h2>Ανακοινώσεις</h2>
```

```
</div>
```

```
</div>
```

```
{% if announcements % } {#to if an se periptwsh einai adeia h lista kai den uparxoun anakoinwseis#}
```

```
{% for announcement in announcements.items % } {# .items logw tou paginate#}
```

```
<div class="div_announc">
```

```
<div class="announc">
```

```
<div >
```

```
<a class="announc_title" href="{{ url_for('full_announcement',  
announc_id=announcement.id) }}">{{ announcement.announc_title | safe }}</a>
```

```
</div>
```

```
<p class="announc_date">
```

```
{{ announcement.date_created.strftime("%d-%m-%Y" "%H:%M") }}
```

```
</p>
```

```
<p class="text_announc">
```

```
{{ announcement.announc_main | truncate(300) | safe }}
```

```
</p>
```

```
<p class="admin">Διαχειριστής:
```



```

        {% if current_user.admin_key != 12123 %}
            {{ announcement.build_admin.first_name+"
"+announcement.build_admin.last_name }}
            {% else %}
                <span>
                    <a class="admin_name" href="{% url_for('build_admin_announc',
build_admin_id=announcement.build_admin.id) %}"> {{ announcement.build_admin.first_name+"
"+announcement.build_admin.last_name }}</a>
                </span>
            {% endif %}
        </p>

    </div>
</div>

{% endfor %}

{% include "pagination.html" %}

{% endif %}

{% endblock body %}

-----build_admin_annonc.html-----

{% extends "base_html.html" %}

{% block title %}
    Ανακοινώσεις του ενός διαχειριστή
{% endblock title %}

```

```
{% block body %}
```

```
<div class="announc_title">
```

```
<div>
```

```
<h2>Όλες οι ανακοινώσεις του διαχειριστή: </h2>
```

```
<h2>{{ build_admin.first_name }} {{ build_admin.last_name }}</h2>
```

```
</div>
```

```
</div>
```

```
{% if announces_by_build_admin %}
```

```
{% for announc_by_build_admin in announces_by_build_admin.items %}
```

```
<div class="div_announc">
```

```
<div class="announc">
```

```
<div>
```

```
<a class="announc_title" href="{{ url_for('full_announcement',  
announc_id=announc_by_build_admin.id) }}">{{ announc_by_build_admin.announc_title | safe  
}}</a>
```

```
</div>
```

```
<p class="announc_date">[{{ announc_by_build_admin.date_created.strftime('%d-%m-  
%Y' - '%H:%M') }}]</p>
```

```
<p class="text_announc">
```

```
{{ announc_by_build_admin.announc_main | truncate(300) | safe }}
```

```
</p>
```

```
<p class="admin">Διαχειριστής:
```

```
<span class="by_admin_name">
```

```
{{ announc_by_build_admin.build_admin.first_name+"  
"+announc_by_build_admin.build_admin.last_name }}
```

```
</span>
```

```
</p>
```

```

    </div>
</div>

{% endfor %}

<div class="main pagin">
    {% for page in announs_by_build_admin.iter_pages(left_edge=2, right_edge=2,
left_current=1, right_current=1) %}
        {% if page %}
            {% if announs_by_build_admin.page == page %}
                <a class="btn btn-primary" href="{{ url_for('build_admin_announc',
build_admin_id=build_admin.id, page=page) }}"> {{ page }} </a>

            {% else %}
                <a class="btn btn-outline-primary" href="{{ url_for('build_admin_announc',
build_admin_id=build_admin.id, page=page) }}"> {{ page }} </a>

            {% endif %}

        {% else %}

            <span>...</span>

        {% endif %}

    {% endfor %}
</div>

{% endif %}

{% endblock body %}

```

-----contact\_form.html-----

```
{% extends "base_html.html" %}
```

```
{% block title %}
```

```
    Φόρμα επικοινωνίας
```

```
{% endblock title %}
```

```
{% block body %}
```

```
    <div class="announc_title">
```

```
        <div>
```

```
            <h2>Φόρμα επικοινωνίας με τους {{ page_title }}</h2>
```

```
        </div>
```

```
    </div>
```

```
    <div class="div_new_announc">
```

```
        <div class="forms">
```

```
            <form method="POST" novalidate>
```

```
                {{ contact_form.csrf_token }}
```

```
            <div class="mb-3">
```

```
                {{ contact_form.contact_title.label(class="form-label") }}
```

```
                {{ contact_form.contact_title(class="form-control") }}
```

```
            {% if contact_form.contact_title.errors %}
```

```
                <ul class=errors>
```

```
                    {% for error in contact_form.contact_title.errors %}
```

```
                        <li class="alert text-danger">{{ error }}</li>
```

```
                    {% endfor %}
```

```

        </ul>
        {% endif %}
    </div>

    <div class="mb-3">
        {{ contact_form.contact_main.label(class="form-label") }}
        {{ contact_form.contact_main(class="form-control", rows="5") }} <!--rows=5 tha
fainontai 5 grammes-->

        {% if contact_form.contact_main.errors %}
            <ul class=errors>
                {% for error in contact_form.contact_main.errors %}
                    <li class="alert text-danger">{{ error }}</li>
                {% endfor %}
            </ul>
        {% endif %}
    </div>

    <div class="mb-3">
        {{ contact_form.submit_btn(class="btn btn-primary") }}
    </div>
</form>
</div>
</div>

```

```
{% endblock body %}
```

```
-----footer.html-----
```

```
<footer>
```

```
<div>
```

```
Copyright: Chris Akriviadis -
```

```

    <script>document.write(new Date().getFullYear())</script>
</div>

<div> version 1.1 </div>
</footer>

-----full_announcement.html-----

{% extends "base_html.html" %}

{% block title %}
    Ανακοινώσεις
{% endblock title %}

{% block body %}

<div class="announc_title">
    <div>
        <h2>Πλήρης ανακοίνωση</h2>
    </div>
</div>

{% if full_announcement %}
    <div class="div_announc">
        <div class="announc">

            <div>
                {% if full_announcement.build_admin == current_user %}

```

```
<a class="btn btn-primary btn-sm mt-1 mb-1" href="{{ url_for('edit_announcement',
announc_id=full_announcement.id) }}">Τροποποίηση</a>
```

```
<button type="button" class="btn btn-danger btn-sm m-1" data-bs-toggle="modal" data-
bs-target="#deleteModal">Διαγραφή</button>
```

```
{% endif % }
```

```
</div>
```

```
<div class="announc_title_full_announc">
```

```
  {{ full_announcement.announc_title | safe }}
```

```
</div>
```

```
<p class="announc_date">{{ full_announcement.date_created.strftime('%d-%m-%Y' -
"%H:%M') }}</p>
```

```
<p class="text_announc">
```

```
  {{ full_announcement.announc_main | safe }}
```

```
</p>
```

```
<p class="admin">Διαχειριστής:
```

```
{% if current_user.admin_key != 12123 % }
```

```
  {{ full_announcement.build_admin.first_name+"
"+full_announcement.build_admin.last_name }}
```

```
{% else % }
```

```
<span>
```

```
<a class="admin_name" href="{{ url_for('build_admin_announc',
build_admin_id=full_announcement.build_admin.id) }}">{{
full_announcement.build_admin.first_name+" "+full_announcement.build_admin.last_name }}</a>
```

```
</span>
```

```
{% endif % }
```

```
</p>
```

```
</div>
```

```
</div>
```

```

{% endif % }

<!-- Modal -->
<div class="modal fade" id="deleteModal" tabindex="-1" aria-labelledby="deleteModalLabel" aria-
hidden="true">
  <div class="modal-dialog">
    <div class="modal-content">
      <div class="modal-header">
        <h5 class="modal-title" id="exampleModalLabel">Διαγραφή ανακοίνωσης</h5>
        <button type="button" class="btn-close" data-bs-dismiss="modal" aria-
label="Close"></button>
      </div>
      <div class="modal-body">
        Είστε σίγουρος/η ότι θέλετε να διαγράψετε την ανακοίνωση;
      </div>
      <div class="modal-footer">
        <button type="button" class="btn btn-secondary" data-bs-dismiss="modal">Όχι</button>
        <form action="{{ url_for('delete_announcement', announc_id=full_announcement.id) }}"
method="POST">
          <input type="submit" class="btn btn-danger" value="Διαγραφή Ανακοίνωσης">
        </div>
      </div>
    </div>
  </div>
</div>
{% endblock body % }

```

-----history.html-----



```
{% extends "base_html.html" %}
```

```
{% block title %}
```

```
    Ιστορικό
```

```
{% endblock title %}
```

```
{% block body %}
```

```
    <p class="main-article-title">Ιστορικό {{ page_title }}</p>
```

```
    {% if history %}
```

```
        {% for h in history.items %}
```

```
            <div class="logar">
```

```
                <a class="" href="{{ url_for('download_files', upload_id=h.id) }}">{{ h.filename | safe }}</a>
```

```
            </div>
```

```
        {% endfor %}
```

```
    {% for page in history.iter_pages(left_edge=2, right_edge=2, left_current=1, right_current=1) %}
```

```
        {% if page %}
```

```
            {% if history.page == page %}
```

```
                <a class="btn btn-primary" href="{{ url_for('history', page=page, value_id=value_id) }}"> {{ page }} </a>
```

```
            {% else %}
```

```
                <a class="btn btn-outline-primary" href="{{ url_for('history', page=page, value_id=value_id) }}"> {{ page }} </a>
```

```
            {% endif %}
```

```
        {% else %}
```

```
            <span>...</span>
```

```
{% endif % }
```

```
{% endfor % }
```

```
{% endif % }
```

```
{% endblock body % }
```

```
-----index.html-----
```

```
{% extends "base_html.html" % }
```

```
{% block title % }
```

```
    Αρχική σελίδα
```

```
{% endblock title % }
```

```
{% block body % }
```

```
    <div class="announc_title">
```

```
        <div>
```

```
            <h2>{{ page_title }}</h2>
```

```
        </div>
```

```
    </div>
```

```
<div class="div_diast">
```

```
    <div>
```

```
        <form method="POST" action="{{ url_for('index') }}">
```

```
            <label>Διάστημα </label>
```

```
            <label >από:</label>
```

```
            <input type="date" id="" name="from" value="{{ from_today }}">
```

```

    <label >έως:</label>
    <input type="date" id="" name="to" value="{{to_today}}">
    <button class="">Αποστολή</button>
  </form>
</div>
</div>

<div class="">
  <canvas id="current_day" width="900" height="350"></canvas>
</div>

<script>

var ctx = document.getElementById("current_day").getContext("2d");
var current_day = new Chart(ctx,
  {
    type: 'line',
    data:
    {
      labels: {{field1 | safe }},
      datasets:
      [
        {
          label: "Wres",
          data: {{field2 | safe }},
          fill: false,
          borderColor: "rgb(255,0,0)",
          lineTension: 0.1
        }
      ]
    },
    options:

```

```
{
    responsive: false
}
});
```

```
</script>
```

```
<div class="div_sun">
```

```
<div>
```

```
<h3>{{sunolo}}</h3>
```

```
</div>
```

```
<div>
```

```
<h3>{{pososto}}</h3>
```

```
</div>
```

```
</div>
```

```
{% endblock body % }
```

```
-----index_petrol.html-----
```

```
{% extends "base_html.html" % }
```

```
{% block title % }
```

```
    Στάθμη δεξαμενής
```

```
{% endblock title % }
```

```
{% block body % }
```

```
<div class="announc_title">
```

```
<div>
```

```

        <h2>{{ page_title }}</h2>
    </div>
</div>

<div class="div_diast">
    <div>
        <form method="POST" action="{{ url_for('index_petrol') }}">
            <label >Διάστημα </label>
            <label >από:</label>
            <input type="date" id="" name="from1" value="{{ from_today }}">
            <label >Έως:</label>
            <input type="date" id="" name="to1" value="{{ to_today }}">
            <button class="">Αποστολή</button>
        </form>
    </div>
</div>

<div>
    <canvas id="current_day" width="900" height="400"></canvas>
</div>

<script>

var ctx = document.getElementById("current_day").getContext("2d");
var current_day = new Chart(ctx,
    {
        type: 'line',
        data:
        {
            labels: {{ field1 | safe }},
            datasets:
            [
                {

```

```

        label: "liter",
        data: {{field2 | safe}},
        fill: false,
        borderColor: "rgb(255,0,0)",
        lineTension: 0.1
    }
]
},
options:
{
    responsive: false
}
});

```

```
</script>
```

```
<div class="div_sun">
```

```
<div>
```

```
<h3>{{sunolo}}</h3>
```

```
</div>
```

```
<div>
```

```
<h3>{{pososto}}</h3>
```

```
</div>
```

```
{% if bef_tit != "" %}
```

```
<span class="grammh"></span>
```

```
{% endif %}
```

```
<div class="keno">
```

```
<h3>{{bef_tit}}</h3>
```

```
</div>
```

```
<div>
  <h3>{{bfr}}</h3>
</div>
```

```
<div>
  <h3>{{sun}}</h3>
</div>
```

```
</div>
{% endblock body %}
```

-----login.html-----

```
<!DOCTYPE html>
<html lang="en" xmlns="http://www.w3.org/1999/html">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>
    Είσοδος
  </title>

  <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha1/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-
GLhlTQ8iRABdZL16O3oVMWSktQOp6b7In1Zl3/Jr59b6EGGoI1aFkw7cmDA6j6gD"
crossorigin="anonymous"> <!--link gia sundesh me to bootstrap-->

  <link rel="stylesheet" type="text/css" href="{{ url_for('static', filename= 'css/my_styles.css')}}">
  <link rel="shortcut icon" href="{{ url_for('static', filename= 'images/favicon.png')}}">
</head>
```

```

<body class="login">

<div class="flash_msg">
  <div class="">
    {% with messages = get_flashed_messages(with_categories=True) %}
    {% if messages %}
      {% for categ,msg in messages %}
        <div class="alert alert-{{ categ }} alert-dismissible fade show" role="alert">
          {{ msg | safe }} {# thelei kai | safe gia na deixnei swsta ta tags tou flash px to usernmae
ths eggrafhs xwris ta html tags an ta xrhsimopoiw mesa sto flash#}
          <button type="button" class="btn-close" data-bs-dismiss="alert" aria-
label="Close"></button>
        </div>
      {% endfor %}
    {% endif %}
    {% endwith %}
  </div>
</div>

<div class="login_box">
  <h2>Είσοδος</h2>
  <div>

    <form method="POST" novalidate>
      {{ loginform.csrf_token }}

      <div class="email_pass">
        <div class="email">
          {{ loginform.email(class="form-control") }}

          {% if loginform.email.errors %}

```



```

        <ul class=errors>
            {% for error in loginform.email.errors %}
                <li class="alert text-danger">{{ error }}</li>
            {% endfor %}
        </ul>
    {% endif %}
</div>

<div>
    {{ loginform.password(type="password", class="form-control") }}
    {% if loginform.password.errors %}
        <ul class=errors>
            {% for error in loginform.password.errors %}
                <li class="alert text-danger">{{ error }}</li>
            {% endfor %}
        </ul>
    {% endif %}
</div>
</div>

<div class="remem-forg">
    <span >
        {{ loginform.remember_me(class="check form-check-input") }}
        {{ loginform.remember_me.label(class="remem-forg_a") }}
    </span>

    <span class="">
        <a class="forgot_pass" href="{{ url_for('reset_password') }}">Ξεχάσατε τον
κωδικό σας;</a>
    </span>
</div>

<div>

```

```

        <div>
            {{ loginform.submit_btn(class="btn btn_ok btn-primary") }}
        </div>
    </div>

    <div>
        <a class="" href="/signup/">Εγγραφή</a>
    </div>
</form>
</div>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-
alpha1/dist/js/bootstrap.bundle.min.js" integrity="sha384-
w76AqPfDkMBDXo30jS1Sgez6pr3x5MlQ1ZAGC+nuZB+EYdgRZgiwxhTBTkF7CXvN"
crossorigin="anonymous"></script>

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.3/jquery.min.js"></script>

</div>

</body>
</html>

```

-----navbar.html-----

```
{% if current_user.is_authenticated %}
```

```

<nav class="navbar navbar-dark bg-dark"> <!-- fixed-top-->
    <div class="container-fluid">
        <a class="navbar-brand">Παρακολούθηση κοινόχρηστων ενεργειακών πόρων!</a>
        <button class="navbar-toggler" type="button" data-bs-toggle="offcanvas" data-bs-
target="#offcanvasDarkNavbar" aria-controls="offcanvasDarkNavbar">
            <span class="navbar-toggler-icon"></span>

```

```

</button>

<div class="offcanvas offcanvas-end text-bg-dark" tabindex="-1" id="offcanvasDarkNavbar" aria-
labelledby="offcanvasDarkNavbarLabel">

  <div class="offcanvas-header">

    <h5 class="offcanvas-title" id="offcanvasDarkNavbarLabel">Menu</h5>

    <button type="button" class="btn-close btn-close-white" data-bs-dismiss="offcanvas" aria-
label="Close"></button>

  </div>

  <div class="offcanvas-body">

    <ul class="navbar-nav justify-content-end flex-grow-1 pe-3">

      <li class="nav-item dropdown">

        <a class="nav-link dropdown-toggle" href="#" role="button" data-bs-toggle="dropdown"
aria-expanded="false">

          Welcome {{ current_user.username }}

        </a>

        <ul class="dropdown-menu dropdown-menu-dark">

          <li><a class="dropdown-item" href="/update_account/">Λογαριασμός</a></li>

          <li><hr class="dropdown-divider"> <!--οριζοντια γραμμη--></li>

        </ul>

      </li>

      <li class="nav-item">

        <a class="nav-link" aria-current="page" href="/index/">Αρχική σελίδα</a>

      </li>

      <li class="nav-item">

        <a class="nav-link" href="/announcements/">Ανακοινώσεις</a>

      </li>

      <li class="nav-item dropdown">

        <a class="nav-link dropdown-toggle" href="#" role="button" data-bs-toggle="dropdown"
aria-expanded="false">

```

Ιστορικό

</a>

<ul class="dropdown-menu dropdown-menu-dark">

<li><a class="dropdown-item" href="/history/1">Συμβάντα</a></li>

<li><hr class="dropdown-divider"> <!--οριζοντια γραμμη--></li>

<li><a class="dropdown-item" href="/history/2">Στάθμη δεξαμενής</a></li>

<li><hr class="dropdown-divider"> <!--οριζοντια γραμμη--></li>

<li><a class="dropdown-item" href="/history/3">Ενεργοποίηση λέβητα</a></li>

<li><hr class="dropdown-divider"> <!--οριζοντια γραμμη--></li>

<li><a class="dropdown-item" href="/history/9">Ωρες λειτουργίας λέβητα</a></li>

<li><hr class="dropdown-divider"> <!--οριζοντια γραμμη--></li>

<li><a class="dropdown-item" href="/history/4">Λογαριασμός ρεύματος</a></li>

<li><hr class="dropdown-divider"> <!--οριζοντια γραμμη--></li>

<li><a class="dropdown-item" href="/history/5">Λογαριασμός πετρελαίου</a></li>

<li><hr class="dropdown-divider"> <!--οριζοντια γραμμη--></li>

<li><a class="dropdown-item" href="/history/6">Λογαρ. πλήρωσης δεξαμενής</a></li>

<li><hr class="dropdown-divider"> <!--οριζοντια γραμμη--></li>

<li><a class="dropdown-item" href="/history/7">Κοινόχρηστα</a></li>

<li><hr class="dropdown-divider"> <!--οριζοντια γραμμη--></li>

<li><a class="dropdown-item" href="/history/8">Άλλοι λογαριασμοί</a></li>

</ul>

</li>

<li class="nav-item">

<a class="nav-link" href="/contact\_form/">Φόρμα επικοινωνίας</a>

</li>

{% if current\_user.admin\_key == 12123 % }

<li class="nav-item">

<a class="nav-link" href="/new\_announcement/">Νέα ανακοίνωση</a>

</li>

```
<li class="nav-item">
  <a class="nav-link" href="/upload_files/">Ανέβασμα αρχείου</a>
</li>
```

```
<li class="nav-item">
  <a class="nav-link" href="/people/">Στοιχεία Ενοίκων</a>
</li>
```

```
<li class="nav-item">
  <a class="nav-link" href="/total_h_hours/">Ωρες λειτουργίας ενοίκων</a>
</li>
```

```
{% endif % }
```

```
<div class="my_nav_exit">
```

```
  <a class="nav-link" href="/logout/">Εξοδος</a>
```

```
</div>
```

```
</ul>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
</nav>
```

```
{% endif % }
```

```
-----new_announcement.html-----
```

```
{% extends "base_html.html" %}
```

```
{% block title %}
```

```
    {{ page_title }}
```

```
{% endblock title %}
```

```
{% block body %}
```

```
<div class="announc_title">
```

```
    <div>
```

```
        <h2>{{ page_title }}</h2>
```

```
    </div>
```

```
</div>
```

```
<div class="div_new_announc">
```

```
    <div class="forms">
```

```
        <form method="POST" novalidate>
```

```
            {{ new_annc_form.csrf_token }}
```

```
            <div class="mb-3">
```

```
                {{ new_annc_form.announc_title.label(class="form-label") }}
```

```
                {{ new_annc_form.announc_title(class="form-control") }}
```

```
            {% if new_annc_form.announc_title.errors %}
```

```
                <ul class="errors">
```

```
                    {% for error in new_annc_form.announc_title.errors %}
```

```
                        <li class="alert text-danger">{{ error }}</li>
```

```
                    {% endfor %}
```

```
                </ul>
```

```
            {% endif %}
```

```
</div>
```

```
<div class="mb-3">
```

```
  {{ new_annc_form.announc_main.label(class="form-label") }}
```

```
  {{ new_annc_form.announc_main(class="form-control", rows="5") }}
```

```
  {% if new_annc_form.announc_main.errors % }
```

```
    <ul class=errors>
```

```
      {% for error in new_annc_form.announc_main.errors % }
```

```
        <li class="alert text-danger">{{ error }}</li>
```

```
      {% endfor % }
```

```
    </ul>
```

```
  {% endif % }
```

```
</div>
```

```
<div class="mb-3">
```

```
  {{ new_annc_form.submit_btn(class="btn btn-primary") }}
```

```
</div>
```

```
</form>
```

```
</div>
```

```
</div>
```

```
{% endblock body % }
```

```
-----pagination.html-----
```

```
<div class="main pagin">
```

```
  {% for page in announcements.iter_pages(left_edge=2, right_edge=2, left_current=1,  
right_current=1) % }
```

```
    {% if page % }
```

```
      {% if announcements.page == page % }
```

```
    <a class="btn btn-primary btn-sm" href="{{ url_for('announcements', page=page) }}"> {{
page }} </a>
```

```
    {% else % }
```

```
    <a class="btn btn-outline-primary btn-sm " href="{{ url_for('announcements', page=page)
}}"> {{ page }} </a>
```

```
    {% endif % }
```

```
{% else % }
```

```
<span>...</span>
```

```
{% endif % }
```

```
{% endfor % }
```

```
</div>
```

```
-----people.html-----
```

```
{% extends "base_html.html" % }
```

```
{% block title % }
```

```
    {{ page_title }}
```

```
{% endblock title % }
```

```
{% block body % }
```

```
<div class="announc_title">
```

```
<div>
```

```
<h2>{{ page_title }}</h2>
```



</div>

</div>

<div class="div\_new\_announc">

<table>

<colgroup>

<col>

</colgroup>

{% if val\_id == 1 % }

<tr>

<th class="th\_center">Α/Α</th>

<th class="font\_people\_pad">Συμβάν</th>

<th class="font\_people\_pad">Ημερομηνία/Ωρα</th>

</tr>

{% elif val\_id == 4 or val\_id == 5 or val\_id == 6 or val\_id == 7 or val\_id == 8 % }

<tr>

<th class="th\_center">Α/Α</th>

<th class="font\_people\_pad">Λογαριασμός</th>

<th class="font\_people\_pad">Ημερομηνία/Ωρα</th>

</tr>

{% else % }

<tr>

<th class="th\_center">Α/Α</th>

<th class="font\_people\_pad">Επώνυμο</th>

<th class="font\_people\_pad">Όνομα</th>

<th class="font\_people\_pad">Email</th>

<th class="font\_people\_pad">Όροφος</th>

<th class="font\_people\_pad">Διαμέρισμα</th>

</tr>

{% endif % }

```

{% if people %}
  {% for p in people.items %}
    {% if val_id == 1 %}
      <tr>
        <th class="th_center">{{ p.id | safe }}</th>
        <th class="font_people"> {{ p.eventb | safe }}</th>
        <th class="font_people">{{ p.date_created.strftime("%d/%m/%Y %H:%M") | safe
}}</th>
      </tr>
    {% elif val_id == 4 or val_id == 5 or val_id == 6 or val_id == 7 or val_id == 8 %}
      <tr>
        <th class="th_center">{{ p.id | safe }}</th>
        <th class="font_people"> <a class="admin_name" href="{{ url_for('download_files',
upload_id=p.id) }}"> {{ p.filename | safe }}</a> </th>
        <th class="font_people">{{ p.date_created.strftime("%d/%m/%Y %H:%M") | safe
}}</th>
      </tr>
    {% else %}
      <tr>
        <th class="th_center">{{ p.id | safe }}</th>
        <th class="font_people"> {{ p.last_name | safe }}</th>
        <th class="font_people">{{ p.first_name | safe }}</th>
        <th class="font_people">{{ p.email | safe }}</th>
        <th class="th_center font_people">{{ p.floor | safe }}</th>
        <th class="th_center font_people">{{ p.house_number | safe }}</th>
      </tr>
    {% endif %}
  {% endfor %}
{% endif %}

</table>

```

```
</div>
```

```
<div class="main pagin">
```

```
  {% for page in people.iter_pages(left_edge=2, right_edge=2, left_current=1, right_current=1) %}
```

```
    {% if page %}
```

```
      {% if people.page == page %}
```

```
        {% if val_id == 1 %}
```

```
          <a class="btn btn-primary" href="{{ url_for('people', page=page) }}"> {{ page }} </a>
```

```
        {% else %}
```

```
          <a class="btn btn-primary" href="{{ url_for('people', build_admin_id=people.id, page=page) }}"> {{ page }} </a>
```

```
        {% endif %}
```

```
      {% else %}
```

```
        {% if val_id == 1 %}
```

```
          <a class="btn btn-outline-primary" href="{{ url_for('people', page=page) }}"> {{ page }} </a>
```

```
        {% else %}
```

```
          <a class="btn btn-outline-primary" href="{{ url_for('people', people=people.id, page=page) }}"> {{ page }} </a>
```

```
        {% endif %}
```

```
      {% endif %}
```

```
    {% else %}
```

```
      <span>...</span>
```

```
    {% endif %}
```

```
  {% endfor %}
```

```
</div>
```

```
{% endblock body %}
```

```
{% if myvar == 1 %}  
    {{ myvar }}  
{% else %}  
    {{ None }}  
{% endif %}
```

-----reset\_passw\_email.html-----

```
<!--O vasikos kormos tis html pou den allazei se kamia selida-->  
<!DOCTYPE html>  
<html lang="en" xmlns="http://www.w3.org/1999/html">  
<head>  
    <meta charset="UTF-8">  
    <meta http-equiv="X-UA-Compatible" content="IE=edge">  
    <meta name="viewport" content="width=device-width, initial-scale=1.0">  
    <title>  
        Αλλαγή κωδικού πρόσβασης  
    </title>  
  
    <link rel="stylesheet"  
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">  
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha1/dist/css/bootstrap.min.css"  
rel="stylesheet" integrity="sha384-  
GLhlTQ8iRABdZLI6O3oVMWSktQOp6b7In1Zl3/Jr59b6EGGoI1aFkw7cmDA6j6gD"  
crossorigin="anonymous">  
  
    <link rel="stylesheet" type="text/css" href="{{ url_for('static', filename= 'css/my_styles.css')}}">  
    <link rel="shortcut icon" href="{{ url_for('static', filename= 'images/favicon.png')}}">  
  
</head>  
  
<body class="body_reset_passw_email">
```

```

<h2>Αλλαγή κωδικού πρόσβασης</h2>
<!-- to ksexasate ton kwdiko sas sto login -->
<div class="reset_passw_email">

<form method="POST" novalidate>
  {{ reset_passw_email.csrf_token }}

  <div>
    {{ reset_passw_email.email.label(class="form-label") }}
    {{ reset_passw_email.email(class="form-control") }}

    {% if reset_passw_email.email.errors %}
      <ul class=errors>
        {% for error in reset_passw_email.email.errors %}
          <li class="alert text-danger"> {{ error }} </li>
        {% endfor %}
      </ul>
    {% endif %}
  </div>

  <div>

    <span>
      {{ reset_passw_email.submit_btn(class="btn btn-primary") }}
    </span>
  </div>

</form>
</div>

</body>
</html>

```

-----reset\_password.html-----

```
<!DOCTYPE html>
<html lang="en" xmlns="http://www.w3.org/1999/html">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>
    Αλλαγή κωδικού πρόσβασης
  </title>

  <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha1/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-
GLhITQ8iRABdZLI6O3oVMWSktQOp6b7In1Zl3/Jr59b6EGGoI1aFkw7cmDA6j6gD"
crossorigin="anonymous">

  <link rel="stylesheet" type="text/css" href="{ {url_for('static', filename= 'css/my_styles.css')}}">
  <link rel="shortcut icon" href="{ {url_for('static', filename= 'images/favicon.png')}}">

</head>

<body class="body_reset_passw_email">
  <h2>Ορισμός νέου κωδικού πρόσβασης</h2>
  <!-- to link apo to email gia allagh kwdikou otan to ksexasw -->
  <div class="reset_passw_email">
```

<!--to kathe form prepei na exei ena action gia na leei pia selda na kalesei kai na asteilei ta dedomena

tha stelnei me post request-->

```
<form method="POST" action="{{ url_for('reset_password_token', token=token) }}"
novalidate>
```

```
    {{ reset_passw_form.csrf_token }} <!--methodos tou Flaskform gia na perasei sth selida to
Secret Key-->
```

```
    <div>
```

```
        {{ reset_passw_form.password.label(class= "form-label") }} <!--to form einai apo to
render to label apo to forms-->
```

```
        {{ reset_passw_form.password(class= "form-control", type="password") }}
```

```
        {% if reset_passw_form.password.errors %}
```

```
        <ul class="errors">
```

```
            {% for error in reset_passw_form.password.errors %} <!--an uparxoun polla errors-->
```

```
>
```

```
                <li class="alert text-danger">{{ error }}</li>
```

```
            {% endfor %}
```

```
        </ul>
```

```
        {% endif %}
```

```
    </div>
```

```
    <div class="">
```

```
        {{ reset_passw_form.confirm_password.label(class= "form-label") }} <!--to form einai
apo to render to label apo to forms-->
```

```
        {{ reset_passw_form.confirm_password(class= "form-control", type="password") }} <!--
oi klaseis mpainoun mesa sth methodo-->
```

```
        {% if reset_passw_form.confirm_password.errors %} <!--an uparxoun erros sto username
emfanizontai se lista-->
```

```
        <ul class="errors">
```

```
            {% for error in reset_passw_form.confirm_password.errors %} <!--an uparxoun polla
errors-->
```

```
                <li class="alert text-danger">{{ error }}</li>
```

```
            {% endfor %}
```

```
        </ul>
```

```

        {% endif % }
    </div>

    <div>
        <span>
            {{ reset_passw_form.submit_btn(class="btn btn-primary") }}
        </span>
    </div>

</form>
</div>

</body>
</html>

-----signup.html-----

<!DOCTYPE html>
<html lang="en" xmlns="http://www.w3.org/1999/html">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>
        Εγγραφή νέου χρήστη
    </title>

    <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">

    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha1/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-
GLhlTQ8iRABdZL16O3oVMWSktQOp6b7In1Zl3/Jr59b6EGGoI1aFkw7cmDA6j6gD"
crossorigin="anonymous">

```



```
<link rel="stylesheet" type="text/css" href="{ {url_for('static', filename= 'css/my_styles.css')}}">
```

```
<link rel="shortcut icon" href="{ {url_for('static', filename= 'images/favicon.png')}}">
```

```
</head>
```

```
<body>
```

```
<div class="signup_out_div">
```

```
<div>
```

```
<div class="signup_title">
```

```
<h2>Σελίδα εγγραφής νέου χρήστη</h2>
```

```
</div>
```

```
<form method="POST" action="#" novalidate>
```

```
{ { signupform.csrf_token } }
```

```
<!--
```

```
Fist_Name
```

```
-->
```

```
<div class="mb-3">
```

```
{ { signupform.first_name.label(class= "form-label") } }
```

```
{ { signupform.first_name(class= "form-control") } }
```

```
{% if signupform.first_name.errors % }
```

```
<ul class="errors">
```

```
{% for error in signupform.first_name.errors % }
```

```
<li class="alert text-danger">{ { error } }</li>
```

```
{% endfor % }
```

```
</ul>
```

```
{% endif % }
```

```
</div>
```

```

<!--                                LAst_Name                                -->
<div class="mb-3">
    {{ signupform.last_name.label(class= "form-label") }}
    {{ signupform.last_name(class= "form-control") }}

    {% if signupform.last_name.errors %}
        <ul class="errors">
            {% for error in signupform.last_name.errors %}
                <li class="alert text-danger">{{ error }}</li>
            {% endfor %}
        </ul>
    {% endif %}
</div>

```

```

<!--                                EMAIL                                -->
<div class="mb-3">
    {{ signupform.email.label(class= "form-label") }}
    {{ signupform.email(class= "form-control") }}
    {% if signupform.email.errors %}
        <ul class="errors">
            {% for error in signupform.email.errors %}
                <li class="alert text-danger">{{ error | safe }}</li>
            {% endfor %}
        </ul>
    {% endif %}
</div>

```

```

<!--                                USERNAME                                -->
<div class="mb-3">
    {{ signupform.username.label(class= "form-label") }}
    {{ signupform.username(class= "form-control") }}

```

```

    {% if signupform.username.errors % }
    <ul class="errors">
        {% for error in signupform.username.errors % }
            <li class="alert text-danger">{{ error |safe }}</li>
        {% endfor % }
    </ul>
    {% endif % }
</div>

```

```

<!--          PASSWORD          -->

```

```

<div class="mb-3">
    {{ signupform.password.label(class= "form-label") }}
    {{ signupform.password(class= "form-control", type="password") }}

    {% if signupform.password.errors % }
    <ul class="errors">
        {% for error in signupform.password.errors % }
            <li class="alert text-danger">{{ error }}</li>
        {% endfor % }
    </ul>
    {% endif % }
</div>

```

```

<!--          CONFIRM PASSWORD          -->

```

```

<div class="mb-3">
    {{ signupform.confirm_password.label(class= "form-label") }}
    {{ signupform.confirm_password(class= "form-control", type="password") }}

    {% if signupform.confirm_password.errors % }
    <ul class="errors">
        {% for error in signupform.confirm_password.errors % }

```

```

        <li class="alert text-danger">{{ error }}</li>
    {% endfor %}
</ul>
{% endif %}
</div>

```

```

<!--                                OROFOS-FLOOR                                -->

```

```

<div class="mb-3">
    {{ signupform.floor.label(class="form-label") }}
    {{ signupform.floor(class="form-control") }}

    {% if signupform.floor.errors %}
        <ul class="errors">
            {% for error in signupform.floor.errors %}
                <li class="alert text-danger">{{ error }}</li>
            {% endfor %}
        </ul>
    {% endif %}
</div>

```

```

<!--                                HOUSE NUMBER                                -->

```

```

<div class="mb-3">
    {{ signupform.house_number.label(class="form-label") }}
    {{ signupform.house_number(class="form-control") }}

    {% if signupform.house_number.errors %}
        <ul class="errors">
            {% for error in signupform.house_number.errors %}
                <li class="alert text-danger">{{ error }}</li>
            {% endfor %}
        </ul>
    {% endif %}

```

```

</div>

<!--                ADMIN KEY                -->
<div class="mb-3">
    {{ signupform.admin_key.label(class= "form-label") }}
    {{ signupform.admin_key(class= "form-control") }}

    {% if signupform.admin_key.errors %}
        <ul class="errors">
            {% for error in signupform.admin_key.errors %}
                <li class="alert text-danger">{{ error }}</li>
            {% endfor %}
        </ul>
    {% endif %}
</div>

<!--                BUTTON                -->
<div>
    <div class="mb-3">
        {{ signupform.submit_btn(class="btn btn-primary") }}
    </div>
</div>

</form>
</div>
</div>

</body>
</html>

-----temp_hum.html-----

```

```
{% extends "base_html.html" %}
```

```
{% block title %}
```

```
    {{ page_title_up }}
```

```
{% endblock title %}
```

```
{% block body %}
```

```
    <script src='https://cdnjs.cloudflare.com/ajax/libs/Chart.js/1.0.2/Chart.min.js'></script>
```

```
    <div class="announc_title">
```

```
        <div>
```

```
            <h2>{{ page_title }}</h2>
```

```
        </div>
```

```
    </div>
```

```
{% if temp_value_id == 3 %}
```

```
<div class="div_diast">
```

```
    <div>
```

```
        <form method="POST" action="{{ url_for('temp_hum') }}">
```

```
            <label >Ημερομηνία: </label>
```

```
            <input type="date" id="" name="from3" value="{{ from_today }}">
```

```
            <button class="">Αποστολή</button>
```

```
        </form>
```

```
    </div>
```

```
</div>
```

```
<div class="fl_can">
```

```
    <span class="humid">
```

```
        <h3>Εξωτερική θερμοκρασία σε οC</h3>
```

```
        <canvas id="temp" width="900" height="400"></canvas>
```

```
</span>
```

```
<span class="humid1">
```

```
  <h3>Ποσοτό εξωτερικής υγρασίας %</h3>
```

```
  <canvas id="hum" width="900" height="400"></canvas>
```

```
</span>
```

```
<span class="humid">
```

```
  <h3>Εσωτερική θερμοκρασία σε οC</h3>
```

```
  <canvas id="temp_in" width="900" height="400"></canvas>
```

```
</span>
```

```
<span class="humid1">
```

```
  <h3>Ποσοτό εσωτερικής υγρασίας %</h3>
```

```
  <canvas id="hum_in" width="900" height="400"></canvas>
```

```
</span>
```

```
</div>
```

```
<script>
```

```
// temperature
```

```
  var barData1 = {
```

```
    labels : [
```

```
      {% for item in field1 %}
```

```
        "{{ item }}",
```

```
      {% endfor %}
```

```
    ],
```

```
    datasets : [{
```

```
      fillColor: "rgba(255,148,112,0.2)",
```

```
      strokeColor: "rgba(255,148,112,1)",
```

```

    pointColor: "rgba(255,148,112,1)",
    pointStrokeColor: "#fff",
    pointHighlightFill: "#fff",
    pointHighlightStroke: "rgba(255,148,112,1)",
    bezierCurve : false,
    data : [
        {% for item in field2 %}
        {{ item }},
        {% endfor %}
    ],

    ]
}

Chart.defaults.global.animationSteps = 50;
Chart.defaults.global.tooltipYPadding = 16;
Chart.defaults.global.tooltipCornerRadius = 0;
Chart.defaults.global.tooltipTitleFontStyle = "normal";
Chart.defaults.global.tooltipFillColor = "rgba(0,0,0,0.8)";
Chart.defaults.global.animationEasing = "easeOutBounce";
Chart.defaults.global.responsive = false;
Chart.defaults.global.scaleLineColor = "black";
Chart.defaults.global.scaleFontSize = 16;

// get bar chart canvas
var mychart1 = document.getElementById("temp").getContext("2d");

steps = 10
max = {{ max }}
// draw bar chart
var LineChartDemo1 = new Chart(mychart1).Line(barData1, {

```



```
scaleOverride: true,  
scaleSteps: steps,  
scaleStepWidth: Math.ceil(max / steps),  
scaleStartValue: 0,  
scaleShowVerticalLines: true,  
scaleShowGridLines : true,  
barShowStroke : true,  
scaleShowLabels: true,  
bezierCurve: false,  
});
```

```
</script>
```

```
<script>
```

```
// humidity
```

```
var barData2 = {  
  labels : [  
    {% for item in field1 %}  
    "{{ item }}",  
    {% endfor %}  
  ],  
  
  datasets : [{  
    fillColor: "rgba(151,187,205,0.2)",  
    strokeColor: "rgba(151,187,205,1)",  
    pointColor: "rgba(151,187,205,1)",  
    pointStrokeColor: "#fff",  
    pointHighlightFill: "#fff",  
    pointHighlightStroke: "rgba(151,187,205,1)",  
    bezierCurve : false,  
    data : [  

```

```

        {% for item in field3 % }
        {{ item }},
    {% endfor % }}
    },
]
}

```

```

Chart.defaults.global.animationSteps = 50;
Chart.defaults.global.tooltipYPadding = 16;
Chart.defaults.global.tooltipCornerRadius = 0;
Chart.defaults.global.tooltipTitleFontStyle = "normal";
Chart.defaults.global.tooltipFillColor = "rgba(0,0,0,0.8)";
Chart.defaults.global.animationEasing = "easeOutBounce";
Chart.defaults.global.responsive = false;
Chart.defaults.global.scaleLineColor = "black";
Chart.defaults.global.scaleFontSize = 16;

// get bar chart canvas
var mychart2 = document.getElementById("hum").getContext("2d");

steps = 10
max = { { max } }
// draw bar chart
var LineChartDemo2 = new Chart(mychart2).Line(barData2, {
    scaleOverride: true,
    scaleSteps: steps,
    scaleStepWidth: Math.ceil(max / steps),
    scaleStartValue: 0,
    scaleShowVerticalLines: true,
    scaleShowGridLines : true,

```

```
barShowStroke : true,  
scaleShowLabels: true,  
bezierCurve: false,  
});
```

```
</script>
```

```
<script>
```

```
// temperature_in  
var barData4 = {  
  labels : [  
    {% for item in field1 %}  
    "{{ item }}",  
    {% endfor %}  
  ],  
  
  datasets : [{  
    fillColor: "rgba(255,148,112,0.2)",  
    strokeColor: "rgba(255,148,112,1)",  
    pointColor: "rgba(255,148,112,1)",  
    pointStrokeColor: "#fff",  
    pointHighlightFill: "#fff",  
    pointHighlightStroke: "rgba(255,148,112,1)",  
    bezierCurve : false,  
    data : [  
      {% for item in field4 %}  
      {{ item }},  
      {% endfor %}]  
    ],  
  },
```

```
    ]  
}
```

```
Chart.defaults.global.animationSteps = 50;  
Chart.defaults.global.tooltipYPadding = 16;  
Chart.defaults.global.tooltipCornerRadius = 0;  
Chart.defaults.global.tooltipTitleFontStyle = "normal";  
Chart.defaults.global.tooltipFillColor = "rgba(0,0,0,0.8)";  
Chart.defaults.global.animationEasing = "easeOutBounce";  
Chart.defaults.global.responsive = false;  
Chart.defaults.global.scaleLineColor = "black";  
Chart.defaults.global.scaleFontSize = 16;
```

```
// get bar chart canvas
```

```
var mychart4 = document.getElementById("temp_in").getContext("2d");
```

```
steps = 10
```

```
max = { { max } }
```

```
// draw bar chart
```

```
var LineChartDemo4 = new Chart(mychart4).Line(barData4, {  
  scaleOverride: true,  
  scaleSteps: steps,  
  scaleStepWidth: Math.ceil(max / steps),  
  scaleStartValue: 0,  
  scaleShowVerticalLines: true,  
  scaleShowGridLines : true,  
  barShowStroke : true,  
  scaleShowLabels: true,  
  bezierCurve: false,  
});
```

```
</script>
```

```
<script>
```

```
// humidity_in
```

```
var barData5 = {  
  labels : [  
    {% for item in field1 %}  
    "{{ item }}",  
    {% endfor %}  
  ],  
  
  datasets : [{  
    fillColor: "rgba(151,187,205,0.2)",  
    strokeColor: "rgba(151,187,205,1)",  
    pointColor: "rgba(151,187,205,1)",  
    pointStrokeColor: "#fff",  
    pointHighlightFill: "#fff",  
    pointHighlightStroke: "rgba(151,187,205,1)",  
    bezierCurve : false,  
    data : [  
      {% for item in field5 %}  
      {{ item }},  
      {% endfor %}]  
    ],  
  }  
}
```

```
Chart.defaults.global.animationSteps = 50;
```

```
Chart.defaults.global.tooltipYPadding = 16;
```

```
Chart.defaults.global.tooltipCornerRadius = 0;
```

```

Chart.defaults.global.tooltipTitleFontStyle = "normal";
Chart.defaults.global.tooltipFillColor = "rgba(0,0,0,0.8)";
Chart.defaults.global.animationEasing = "easeOutBounce";
Chart.defaults.global.responsive = false;
Chart.defaults.global.scaleLineColor = "black";
Chart.defaults.global.scaleFontSize = 16;

// get bar chart canvas
var mychart5 = document.getElementById("hum_in").getContext("2d");

steps = 10
max = { { max } }
// draw bar chart
var LineChartDemo5 = new Chart(mychart5).Line(barData5, {
  scaleOverride: true,
  scaleSteps: steps,
  scaleStepWidth: Math.ceil(max / steps),
  scaleStartValue: 0,
  scaleShowVerticalLines: true,
  scaleShowGridLines : true,
  barShowStroke : true,
  scaleShowLabels: true,
  bezierCurve: false,
});

```

```
</script>
```

```

<div class="div_sun">
  <span class="grammh"></span>
<div>
  <h3>{{ sunolo }}</h3>

```

```
</div>
```

```
</div>
```

```
{% elif temp_value_id == 9 %}
```

```
<div class="div_diast">
```

```
<div>
```

```
<form method="POST" action="{{ url_for('temp_hum') }}">
```

```
<label>Διάστημα </label>
```

```
<label>από:</label>
```

```
<input type="date" id="" name="from4" value="{{ from_today_r }}">
```

```
<label>Έως:</label>
```

```
<input type="date" id="" name="to4" value="{{ to_today_r }}">
```

```
<button class="">Αποστολή</button>
```

```
</form>
```

```
</div>
```

```
</div>
```

```
<div class="div_sun">
```

```
<div>
```

```
<h3>{{ range_sunolo }}</h3>
```

```
</div>
```

```
</div>
```

```
{% endif %}
```

```
{% endblock body %}
```

```
----total_h_hours.html----
```

```
{% extends "base_html.html" %}
```

```
{% block title %}
```

```
    Ωρες λειτουργίας ενοίκων
```

```
{% endblock title %}
```

```
{% block body %}
```

```
    <div class="announc_title">
```

```
        <div>
```

```
            <h2>{{ page_title }}</h2>
```

```
        </div>
```

```
    </div>
```

```
<div class="div_diast">
```

```
    <div>
```

```
        <form method="POST" action="{{ url_for('total_h_hours') }}">
```

```
            <label >Διάστημα </label>
```

```
            <label >από:</label>
```

```
            <input type="date" id="" name="from5" value="{{ from_today }}">
```

```
            <label >Έως:</label>
```

```
            <input type="date" id="" name="to5" value="{{ to_today }}">
```

```
            <button class="">Αποστολή</button>
```

```
        </form>
```

```
    </div>
```

```
</div>
```

```
<div class="div_new_announc">
```

```
    <table>
```

```
        <colgroup>
```

```
            <col>
```



```
</colgroup>
```

```
<tr>
```

```
<th class="th_center">Διαμέρισμα</th>
```

```
<th class="th_center">Συνολικές ώρες</th>
```

```
</tr>
```

```
{% if sunolo % }
```

```
{% for s in sunolo % }
```

```
<tr>
```

```
<th class="th_center">{{ s[0] | safe }}</th>
```

```
<th class="font_people k"> {{ s[1] | safe }}</th>
```

```
</tr>
```

```
{% endfor % }
```

```
{% endif % }
```

```
</table>
```

```
</div>
```

```
{% endblock body % }
```

```
-----update_account.html-----
```

```
{% extends "base_html.html" % }
```

```
{% block title % }
```

```
    Ενημέρωση λογαριασμού
```

```
{% endblock title % }
```

```
{% block body % }
```

```
<div class="announc_title">
```

```
    <div>
```

```
        <h2>Ενημέρωση στοιχείων λογαριασμού</h2>
```

```
    </div>
```

```
</div>
```

```
<div class="forms_upd">
```

```
    <div class="forms_upd_div">
```

```
        <form method="POST" action="#" novalidate>
```

```
            {{ updateaccount.csrf_token }}
```

```
<!--
```

```
        EMAIL
```

```
-->
```

```
    <div class="mb-3">
```

```
        {{ updateaccount.email.label(class="form-label") }}
```

```
        {{ updateaccount.email(class="form-control") }}
```

```
    {% if updateaccount.email.errors % }
```

```
        <ul class="errors">
```

```
            {% for error in updateaccount.email.errors % }
```

```
                <li class="alert text-danger">{{ error | safe }}</li>
```

```
            {% endfor % }
```

```
        </ul>
```

```
    {% endif % }
```

```
</div>
```

```
<!--                                USERNAME                                -->
```

```
<div class="mb-3">

    {{ updateaccount.username.label(class= "form-label") }}
    {{ updateaccount.username(class= "form-control") }}

    {% if updateaccount.username.errors %}
        <ul class="errors">
            {% for error in updateaccount.username.errors %}
                <li class="alert text-danger">{{ error |safe }}</li>
            {% endfor %}
        </ul>
    {% endif %}
</div>
```

```
<!--                                PASSWORD                                -->
```

```
<div class="mb-3">

    {{ updateaccount.password.label(class= "form-label") }}
    {{ updateaccount.password(class= "form-control", type="password") }}

    {% if updateaccount.password.errors %}
        <ul class="errors">
            {% for error in updateaccount.password.errors %}
                <li class="alert text-danger">{{ error }}</li>
            {% endfor %}
        </ul>
    {% endif %}
</div>
```

```
<!--                                Fist_Name                                -->
```

```
<div class="mb-3">
```

```

    {{ updateaccount.first_name.label(class= "form-label") }}
    {{ updateaccount.first_name(class= "form-control") }}

    {% if updateaccount.first_name.errors %}
        <ul class="errors">
            {% for error in updateaccount.first_name.errors %}
                <li class="alert text-danger">{{ error }}</li>
            {% endfor %}
        </ul>
    {% endif %}
</div>

<!--                                LAsT_Name                                -->
<div class="mb-3">
    {{ updateaccount.last_name.label(class= "form-label") }}
    {{ updateaccount.last_name(class= "form-control") }}

    {% if updateaccount.last_name.errors %}
        <ul class="errors">
            {% for error in updateaccount.last_name.errors %}
                <li class="alert text-danger">{{ error }}</li>
            {% endfor %}
        </ul>
    {% endif %}
</div>

<!--                                OROFOS-FLOOR                                -->
<div class="mb-3">
    {{ updateaccount.floor.label(class= "form-label") }}
    {{ updateaccount.floor(class= "form-control") }}

    {% if updateaccount.floor.errors %}

```

```

    <ul class="errors">
      {% for error in updateaccount.floor.errors %}
        <li class="alert text-danger">{{ error }}</li>
      {% endfor %}
    </ul>
  {% endif %}
</div>

```

```

<!--          HOUSE NUMBER          -->

```

```

<div class="mb-3">
  {{ updateaccount.house_number.label(class="form-label") }}
  {{ updateaccount.house_number(class="form-control") }}

  {% if updateaccount.house_number.errors %}
    <ul class="errors">
      {% for error in updateaccount.house_number.errors %}
        <li class="alert text-danger">{{ error }}</li>
      {% endfor %}
    </ul>
  {% endif %}
</div>

```

```

<!--          ADMIN KEY          -->

```

```

<div class="mb-3">
  {{ updateaccount.admin_key.label(class="form-label") }}
  {{ updateaccount.admin_key(class="form-control") }}

  {% if updateaccount.admin_key.errors %}
    <ul class="errors">
      {% for error in updateaccount.admin_key.errors %}
        <li class="alert text-danger">{{ error }}</li>
      {% endfor %}
    </ul>
  {% endif %}
</div>

```

```

        </ul>
        {% endif %}
    </div>

<!--          BUTTON          -->

    <div class="mb-2">
        {{ updateaccount.submit_btn(class="btn btn-primary") }}
    </div>

    </form>
</div>
</div>

{% endblock body %}

-----upload_files.html-----

{% extends "base_html.html" %}

{% block title %}
    Ανέβασμα αρχείου
{% endblock title %}

{% block body %}

    <div class="announc_title">
        <div>
            <h2>Ανέβασμα αρχείου</h2>
        </div>

```

```
</div>
```

```
<div class="upload_files_out_div">
```

```
<div class="upload_files">
```

```
<form method="POST" action="#" enctype="multipart/form-data">
```

```
<input class="" type="file" name="file">
```

```
<div>
```

```
<select name="timologia">
```

```
<option value="" >--Επιλέξτε φάκελο--</option>
```

```
<option value="deh">Λογαριασμός ρεύματος</option>
```

```
<option value="petrol">Λογαριασμός πετρελαίου</option>
```

```
<option value="fuel_tank">Λογαρ. πλήρωσης δεξαμενής</option>
```

```
<option value="koinox">Κοινόχρηστα</option>
```

```
<option value="alloi">Άλλοι λογαριασμοί</option>
```

```
</select>
```

```
</div>
```

```
<button class="btn btn-primary">Αποστολή</button>
```

```
</form>
```

```
</div>
```

```
</div>
```

```
{% endblock body % }
```

## ΠΑΡΑΡΤΗΜΑ Θ : CSS αρχείο

```
/* katholikos epilogeas */
*{
  box-sizing: border-box;
}
/* */
```

```
/* ola ta h2 */
h2{
  font-size: 1.8rem;
  text-shadow: 0 0 23px rgb(2, 73, 237);
  font-weight: 600;
}
/* */
```

```
/* ola ta h3 */
h3{
  font-size: 1.3rem;
  /*text-shadow: 0 0 23px rgb(2, 73, 237);*/
  font-weight: 600;
}
/* */
```

```
/* ola ta label */
label{
  font-size: 1.15rem;
  font-weight: 500;
}
/* */
```



```
/*                                     */

/* selida login                       */
.login{
  background-image: url("27272.jpg");
  background-size: cover;
  display: flex;
  flex-direction: column;
  align-items: center;
}

.login_box{
  order: 2;
  border: solid thin black;
  width: 550px;
  margin-top: 10rem;
  position: fixed;
  border-radius: 10px;
  box-shadow: 0px 0px 60px black;
  padding-top: 1.4rem;
  padding-left: 1.4rem;
  padding-right: 1.4rem;
}

.email_pass{
  margin-top: 20px;
}

.email_pass > div{
  margin-top: 10px;
}
```

```
.login_box > div > form > div:last-child{
    margin-top: 50px;
    margin-bottom: 15px;
    display: flex;
    justify-content: center;
    border-top: solid thin grey;
}
```

```
.login_box > div > form > div:last-child > a{
    color: black;
    text-decoration: none;
    font-size: medium;
}
```

```
.login_box > div > form > div:last-child > a:hover{
    text-decoration: none;
    font-weight: bolder;
}
```

```
.login_box > div > form > div:nth-last-child(2){
    display: flex;
    justify-content: center;
    margin-top: 1.3rem;
}
```

```
/*                                     */
```

```
/* remem-forg                         */
```

```
.remem-forg {
    font-size: 14.5px;
    margin-top: 5px;
```

```

}
.remem-forg_a {
    font-size: 15px;
}

.login_box div:nth-last-child(3) span{
    margin-left: 30px;
}

.login_box div:nth-last-child(3) span ~ span{
    margin-left: 178px;
}

.login_box div:nth-last-child(3){
    display: flex;
    align-items: flex-start;
}

.forgot_pass:link{
    color: blue;
    text-decoration: none;
}

.forgot_pass:hover{
    text-decoration: none;

    font-weight: 620;
}

/*
*/

/* flash messages sto login */

```

```

.flash_msg{
  order: 1;
}
/*                                     */

/* reset password email               */
.body_reset_passw_email{
  display: flex;
  flex-direction: column;
  align-items: center;
  margin-top: 12rem;
}

.body_reset_passw_email h2{
  font-size: 1.35rem;
  align-self: center;
  width: 550px;
  margin-bottom: 18px;
}

.reset_passw_email{
  display: flex;
  flex-direction: column;
  align-items: center;
  border: solid thin black;
  border-radius: .3rem;
  padding: 1%;
  width: 550px;
}

.reset_passw_email form div:first-of-type{

```

```

width: 25em;
margin-bottom: 20px;
}

.reset_passw_email form div:last-of-type{
width: 25em;
margin-bottom: 10px;
margin-top: 25px;
padding-top: 5px;
border-top: solid thin gray;
}
/*
*/

```

```

/* klash button apo bootstrap */

```

```

.btn_ok{
--bs-btn-padding-y: .45rem;
--bs-btn-padding-x: 8rem;
--bs-btn-font-size: 1rem;
font-weight: 500;
}
/*
*/

```

```

/* koutia container */

```

```

.my_container{
background-color: white;
background-image: url("49.jpg");
background-size: contain;
background-repeat: round;
display: flex;

```

```
flex-direction: column;
align-items: center;
}
```

```
.my_container_out_div{
  min-height: 1200px;
  background-color: rgba(245, 245, 245, 0.9);
  border-left: solid thin #0353a4;
  border-right: solid thin #0353a4;
  box-shadow: 0 0 35px 25px black;
  border-bottom-left-radius: 13px;
  border-bottom-right-radius: 13px;
}
```

```
.my_container_div{
  display: grid;
  justify-content: center;
  grid-template-columns: 900px;
  grid-template-rows: 275px auto 80px;
  grid-template-areas: "header"
                       "main"
                       "footer";
  min-height: 1200px;
}
```

```
.my_container_div header{
  grid-area: header;
}
```

```
.my_container_div main{
  grid-area: main;
}
```

```
/*                                     */

/* header tou html                       */
header{
    background-color: #e9ecef;
    grid-area: header; /* header tou grid */
    position: sticky;
    top : 0px;
}
/*                                     */
```

```
/* navbar                               */
.navbar-brand {
    font-size: 1.2rem;
}
```

```
.my_nav_exit{
    border-top: solid thin gray;
    margin-top: 20px;
}
```

```
/*                                     */
```

```
/* footer                               */
.my_container_div footer{
    grid-area: footer;
}
```

```
footer{
    color: #e0fbfc;
```

```
font-weight: 500;
background-color: rgb(29, 28, 29);
display: flex;
flex-direction: column;
justify-content: center;
border-bottom-left-radius: 10px;
border-bottom-right-radius: 10px;
}
```

```
footer > div:first-child{
display: flex;
flex-direction: row;
justify-content: center;
text-shadow: 0 0 25px white;
margin-top: 10px;
}
```

```
footer > div:last-child{
align-self: flex-start;
font-size: x-small;
text-shadow: none;
margin-left: 5px;
margin-top: 10px;
font-weight: 100;
}
```

```
/*
```

```
*/
```

```
/* selidopoihsh
```

```
*/
```

```
.pagin{
display : flex;
justify-content: center;
```



```
margin-top: 3%;
margin-bottom: 1%;
}
/*                                     */

/* onoma admin stis anakoinwseis      */
.admin{
    font-style: italic;
    font-size: 0.85rem;
}

.admin_name{
    font-size: 1rem;
    font-style: normal;
}

.admin_name:link{
    color: black;
    text-decoration: none;
}

.admin_name:visited{ /*otan paththei*/
    color: #696969;
}

.admin_name:hover{
    text-decoration: none;
    font-weight: bold;
}

.by_admin_name{
```

```

    font-size: 1rem;
    font-style: normal;
}
/*                                     */

/* anakoinwseis                       */
.div_new_announc{
    display: flex;
    flex-direction: row;
    justify-content: center;
    border: solid thin black;
    border-radius: .3rem;
    margin: 2.5rem 2rem;
    padding: 2%;
}

.div_announc{
    display: flex;
    flex-direction: row;
    justify-content: center;
    border-top: solid thin black;
    border-radius: .3rem;
    margin: 2.5rem 2rem;
    padding: 2%;
}

.announc{
    width: 950px;
    padding: 5px;
    margin: 10px;
    background-color: white;
}

```

```

    box-shadow : 5px 7px #c0c0c0;
    border-radius: 10px;
}

.announc_title{
    display: flex;
    flex-direction: column;
}

.announc_title > div{
    align-self: center;
    margin-top: 0.5rem;
}

/*
*/

/* formes
*/
.forms{ /* dhmiourgia nesas anakoinwshs */
    width: 900px;
}

.forms_upd{
    display: flex;
    flex-direction: row;
    justify-content: center;
    padding: 2%;
    width: 900px;
}

.forms_upd_div{
    width: 800px;
    border: solid thin black;
}

```

```

border-radius: .3rem;
padding: 20px 20px;
}
/*          */

/* anebasma arxeiwn          */
.upload_files{
border : solid thin ;
border-radius: 5px;
padding : 1rem;
background-color: rgba(255, 255, 255, 0.97);
margin: 0 0px;
}

.upload_files div{ /* oloi oi apogonoi tou upload_files */
margin: 10px 0px;
}

.upload_files div select option:first-child{ /*pseudoklash structural*/
text-align: center;
}

.upload_files_out_div{
display: flex;
flex-direction: column;
align-items: center;
}

.upload_files_title{
font-size: small;
margin-bottom: 12px;
}

```

```
}
```

```
.upload_files_out_div > div > div {  
    display: flex;  
    flex-direction: column;  
    align-items: center;  
}
```

```
.upload_files_out_div > div {  
    margin-top: 50px;  
}
```

```
/*                                     */
```

```
/* titlos anakoinwsewn                */
```

```
.announc_title {  
    font-size: 1.2rem;  
    font-weight: 600;  
    text-decoration: none;  
    color: black;  
    color: black;  
    margin-left: 5px;  
    margin-right: 5px;  
}
```

```
.announc_title:hover {  
    text-decoration: none;  
    font-weight: 900;  
    color: black;  
}
```

```
.text_announc {
```

```
font-size: 1.02rem;
font-weight: 400;
line-height: 1.5;
color: black;
margin: 20px 0px;
}
```

```
.announc_title_full_announc{
font-size: 1.2rem;
font-weight: 600;
text-decoration: none;
color: black;

}
/*
```

```
/* hmeromhnia anakoinwsewn */
.announc_date{
font-size: 0.83rem;
color: rgba(105, 105, 105, 0.8);
}
/*
```

```
/* signup form */
.signup_out_div{
display: flex;
flex-direction: column;
align-items: center;
}
```

```
.signup_title{  
  margin-top: 1rem;  
}
```

```
.signup_out_div form{  
  margin-top: 1rem;  
  display: flex;  
  flex-direction: column;  
  align-items: center;  
  flex-flow: wrap;  
  width: 635px;  
  justify-content: space-between;  
  border: solid thin black;  
  padding: 15px;  
  border-radius: 5px;  
}
```

```
.signup_out_div form > div:nth-child(2){  
  width: 295px;  
}
```

```
.signup_out_div form > div:nth-child(3){  
  width: 295px;  
}
```

```
.signup_out_div form > div:nth-child(4){  
  width: 600px;  
}
```

```
.signup_out_div form > div:nth-child(5){  
  width: 600px;  
}
```

```
.signup_out_div form > div:nth-child(6){
  width: 295px;
}
```

```
.signup_out_div form > div:nth-child(7){
  width: 295px;
}
```

```
.signup_out_div form > div:nth-child(8){
  width: 300px;
}
```

```
.signup_out_div form > div:nth-child(9){
  width: 295px;
}
```

```
.signup_out_div form > div:nth-child(10){
  width: 600px;
}
```

```
.signup_out_div form > div:last-of-type{
  border-top: solid thin gray;
  width: 600px;
  padding-top: 5px;
}
```

```
/*
```

```
/* people.html
```

```
table tr:nth-last-of-type(2n+1){
```



```
background-color: rgba(2, 73, 237, 0.2);  
}
```

```
.th_center{  
    text-align: center;  
}
```

```
col{  
    border-right: solid thin black;  
}
```

```
table tr:first-of-type{  
    border-bottom: solid thin black;  
}
```

```
.font_people{  
    font-weight: 500;  
    padding-left: 12px;  
  
}
```

```
.font_people_pad{  
    padding-left: 10px;  
}
```

```
.div_new_announc table{  
    width: 900px;  
}
```

```
/*
```

```
*/
```

```
.div_diast{
```

```
display: flex;
flex-direction: column;

}
```

```
.div_diast > div{
  align-self: center;
  margin: 1.6rem 1.3rem;
}
```

```
.div_sun{
  display: flex;
  flex-direction: column;

}
```

```
.div_sun > div{
  align-self: center;
  text-align:center;
}
```

```
.div_sun{
  align-self: center;
  margin-top: 1.5rem;
  margin-bottom: 1.5rem;
}
```

```
.fl_can{
display: flex;
flex-direction:column;
```

```
}
```

```
.humid h3{  
margin-top:20px;  
border-bottom:solid thin gray;  
  text-align: center;  
}
```

```
.humid1 h3{  
margin-top:80px;  
border-bottom:solid thin gray;  
text-align: center;  
}
```

```
.logar{  
border:solid thin red;  
text-align: center;  
}
```

```
.keno{  
margin-top:20px;  
}
```

```
.grammh{  
border-top:solid thin blue;  
padding-top:15px;  
}
```

```
.k{  
text-align: center;  
}
```