



ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΑΝΑΠΤΥΞΗ ΔΙΑΔΙΚΤΥΑΚΗΣ ΕΦΑΡΜΟΓΗΣ ΓΙΑ ΤΗ ΜΕΛΕΤΗ ΔΕΔΟΜΕΝΩΝ ΠΡΟΕΡΧΟΜΕΝΩΝ ΑΠΟ ΜΕΤΕΩΡΟΛΟΓΙΚΟΥΣ ΣΤΑΘΜΟΥΣ

Α.Τ.Ε.Ι. ΘΕΣΣΑΛΟΝΙΚΗΣ
Σταθμός Μέτρησης
Περιβαλλοντικής Μόλυνσης

Τμήμα Αυτοματισμού
Εργαστήριο Αυτομάτου Ελέγχου
Τηλ. 2310 031276

Αρχική Περιβάλλον Ιστορικό Ανδρομέδα Αρχιμήδης Online Μετρήσεις [Gr](#) [En](#)

Περιβαλλοντικές Μετρήσεις του τμήματος Αυτοματισμού

Copyright © Τμήμα Αυτοματισμού ΑΤΕΙ Θεσσαλονίκης

Όνοματεπώνυμο: Γλερίδης Χρήστος

Τμήμα Πληροφορικής

Α.Μ.: 042645

Εισηγητές - Επιβλέποντες: Δρ. Κλεφτούρης Δημήτριος

Δρ. Χατζίκος Ευάγγελος

Πρόλογος	4
Ευχαριστίες	5
Εισαγωγή	6
ΜΕΡΟΣ ΠΡΩΤΟ	8
ΚΕΦΑΛΑΙΟ 1^ο	8
1. Βασικά στοιχεία μετεωρολογίας	8
1.1 Εισαγωγή.....	8
1.2 Τι είναι μετεωρολογία.....	8
1.3 Ιστορικά στοιχεία.....	9
1.4 Προγνώσεις καιρού.....	11
1.5 Μοντέλα πρόγνωσης καιρού.....	12
1.6 Μετεωρολογικά φαινόμενα.....	13
1.6.1 Η θερμοκρασία.....	13
1.6.2 Η ατμοσφαιρική πίεση.....	14
1.6.3 Η ατμοσφαιρική υγρασία.....	14
1.7 Επίλογος.....	15
ΚΕΦΑΛΑΙΟ 2^ο	16
2. Ανάλυση σταθμού μέτρησης	16
2.1 Εισαγωγή.....	16
2.2 Κύρια λειτουργία σταθμού περιβαλλοντικού ελέγχου.....	16
2.3 Ο σταθμός μέτρησης Αρχιμήδης.....	17
2.4 Αισθητήρες του σταθμού Αρχιμήδης.....	19
2.4.1 Αισθητήρας κατεύθυνσης αέρα.....	19
2.4.2 Αισθητήρας ταχύτητας αέρα.....	19
2.4.3 Αισθητήρας θερμοκρασίας – υγρασίας αέρα.....	20
2.4.4 Αισθητήρας βροχόπτωσης.....	20
2.5 Δίκτυο Ανδρομέδα.....	21
2.6 PLC.....	23
2.7 Σύστημα SCADA.....	24
2.8 Επίλογος.....	25
ΜΕΡΟΣ ΔΕΥΤΕΡΟ	26
ΚΕΦΑΛΑΙΟ 3^ο	26
3. Τεχνολογίες διαδικτύου	26
3.1 Εισαγωγή.....	26
3.2 HTML.....	26
3.2.1 Χαρακτηριστικά της HTML.....	27
3.2.2 Εκδόσεις της HTML.....	27
3.3 HTTP.....	28
3.3.1 Apache Server.....	30
3.4 CSS.....	30
3.4.1 Αρχές του CSS.....	31
3.5 JavaScript.....	31
3.5.1 JQuery.....	32
3.6 Document Object Model.....	33
3.7 PHP.....	34
3.7.1 Βασικά χαρακτηριστικά της PHP.....	35
3.8 SQL Server.....	36
3.9 MySQL.....	38
3.10 Αρχιτεκτονική Βάσης Δεδομένων με PHP – MySQL.....	39
3.11 Επίλογος.....	40

ΜΕΡΟΣ ΤΡΙΤΟ	41
ΚΕΦΑΛΑΙΟ 4^ο	41
4. Βάση δεδομένων της εφαρμογής	41
4.1 Εισαγωγή	41
4.2 Κανονικοποίηση	41
4.2.1 Μη Κανονική Μορφή	41
4.3.2 1 ^η Κανονική Μορφή (1NF)	42
4.3.3 2 ^η Κανονική Μορφή (2NF)	42
4.3.4 3 ^η Κανονική Μορφή (2NF)	42
4.4 Σχεσιακό Μοντέλο	43
4.5 Οντότητες και Συσχετίσεις	43
4.5.1 Βάση δεδομένων στους σταθμούς μέτρησης	43
4.5.1 Βάση δεδομένων για τη δημιουργία της ιστοσελίδας	45
4.6 Επίλογος	49
ΚΕΦΑΛΑΙΟ 5^ο	50
5. Προγραμματισμός σε PHP	50
5.1 Εισαγωγή	50
5.2 Cookies και Sessions	50
5.2.1 Cookies	50
5.2.2 Sessions	51
5.3 Υπερκαθολικές μεταβλητές \$ _GET και \$ _POST	53
5.4 Σύνδεση με τη βάση δεδομένων	55
5.5 SQL Injection και XSS Attacks	56
5.5.1 SQL Injection	57
5.5.2 XSS Attack	58
5.5.3 Sanitize	58
5.6 Κώδικας εμφάνισης σελίδων	60
5.7 Επιλογή γλώσσας	61
5.8 Εύρεση τελευταίας μέτρησης	62
5.9 Δημιουργία γραφήματος	63
5.10 Δημιουργία αρχείου δεδομένων	63
5.11 Επίλογος	66
ΒΙΒΛΙΟΓΡΑΦΙΑ	67
Έντυπα	67
Ιστοσελίδες	67
Παράρτημα Ι	69
Εγχειρίδιο εγκατάστασης απαιτούμενων εργαλείων	69

Πρόλογος

Η εργασία αυτή υλοποιήθηκε το ακαδημαϊκό έτος 2010 – 2011 από το Χρήστο Γλερίδη, φοιτητή του τμήματος Πληροφορικής του Ανώτατου Τεχνολογικού Ιδρύματος Θεσσαλονίκης, υπό την επίβλεψη του καθηγητή του τμήματος Πληροφορικής κ. Δημήτρη Κλεφτούρη στο πλαίσιο εκπόνησης υποχρεωτικής πτυχιακής εργασίας κατά το τελευταίο εξάμηνο σπουδών, με τίτλο:

“Ανάπτυξη διαδικτυακής εφαρμογής για τη μελέτη δεδομένων προερχόμενων από μετεωρολογικούς σταθμούς.”

Η ιδέα του θέματος αυτού αποφασίστηκε όταν ήρθα σε επαφή με το καθηγητή του τμήματος Αυτοματισμού κ. Ευάγγελο Χατζίκο όπου μαζί με άλλους συναδέλφους είχαν ήδη υλοποιήσει σύστημα συλλογής μετεωρολογικών δεδομένων και μου εξέφρασαν την ανάγκη που παρουσιάστηκε για την εύκολη πρόσβαση στα δεδομένα αυτά μέσω του διαδικτύου. Φυσικά, σημαντικό ρόλο έπαιξε και η ραγδαία αύξηση της ζήτησης υλικοτεχνικών υποδομών σχετικών με το περιβάλλον λόγω κυρίως της ανάπτυξης των φωτοβολταϊκών συστημάτων που πλέον έχουν μπει στην καθημερινή ζωή πολλών νοικοκυριών αφού οι διάφορες επιδοτήσεις που έχουν θεσπιστεί κυρίως τα τέσσερα τελευταία χρόνια καθιστούν τη χρήση τέτοιων συστημάτων αρκετά προσοδοφόρα.

Επίσης, η εφαρμογή που ζητήθηκε λόγω της διαδικτυακής της φύσης, έκανε αρκετά ελκυστική την επιλογή του θέματος αφού η γνώση μίας γλώσσας προγραμματισμού που αφορά το διαδίκτυο όπως είναι η HTML και η PHP και η σύνδεσή της με μία βάση δεδομένων, αποτελούν πλέον απαραίτητο προσόν για την εύρεση εργασίας σε πολλές επιχειρήσεις οι οποίες δραστηριοποιούνται στο χώρο της Πληροφορικής.

Τέλος, η εφαρμογή αυτή με λίγες τροποποιήσεις μπορεί να χρησιμοποιηθεί σε κάθε μετεωρολογικό σταθμό συλλογής δεδομένων κάτι που δίνει ένα ανταγωνιστικό πλεονέκτημα όσον αφορά την εξέλιξή της σε ένα πιο ολοκληρωμένο σύστημα στο μέλλον.

Ευχαριστίες

Σε αυτό το σημείο θα ήθελα να ευχαριστήσω τους καθηγητές και επιβλέποντες της πτυχιακής εργασίας, κ. Ευάγγελο Χατζίκο και κ. Δημήτρη Κλεφτούρη που με την εμπειρία τους με βοήθησαν να ολοκληρώσω το συγκεκριμένο πόνημα.

Φυσικά, δε θα μπορούσα να παραλείψω τον κ. Νικόλαο Μπαρμπαλιό και τους συνάδελφους του τμήματος Αυτοματισμού οι οποίοι με βοήθησαν όποτε τους χρειάστηκα κατά την ανάπτυξη του προγράμματος και κυρίως στην επεξήγηση των δεδομένων που παράγουν οι μετεωρολογικοί σταθμοί αλλά και στην προμήθεια των απαραίτητων τεχνικών εγχειριδίων των μηχανημάτων που χρησιμοποιούνται στους σταθμούς αυτούς.

Τέλος, θέλω να ευχαριστήσω την οικογένειά μου για την ψυχολογική υποστήριξη καθ' όλη τη διάρκεια των σπουδών μου με αποτέλεσμα να φτάσω με τη συγγραφή της εργασίας αυτής στην ολοκλήρωσή τους.

Εισαγωγή

Ο κλάδος της Μετεωρολογίας αποτελεί μία επιστήμη η οποία συνδέεται άμεσα με την καθημερινότητα του ανθρώπου αφού επηρεάζει σε μεγάλο βαθμό τις δραστηριότητές του. Η πρόγνωση των καιρικών φαινομένων βρίσκεται ψηλά στις προτεραιότητες πολλών άλλων επιστημών όπως η γεωλογία, η γεωπονία, η εκμετάλλευση ορυκτού πλούτου και πλήθος άλλων εχόντων άμεση σχέση με το φυσικό περιβάλλον.

Εκτός από αυτό, η φυσική επαφή των ανθρώπων με την ατμόσφαιρα καθιστά τη μελέτη του περιβάλλοντος χώρου αποφασιστικής σημασίας για τη βελτίωση του βιοτικού επιπέδου όλων.

Στην εργασία αυτή καταγράφονται και αναλύονται βασικά θεωρητικά στοιχεία της μετεωρολογίας και συστημάτων και οργάνων μέτρησης στοιχείων της ατμόσφαιρας ενώ παράλληλα αναπτύσσεται διαδικτυακή εφαρμογή με στόχο την παρουσίαση των σταθμών ατμοσφαιρικής μέτρησης “Ανδρομέδα” και “Αρχιμήδης” καθώς και των μετρήσεων που πραγματοποιούνται από τους εν λόγω σταθμούς.

Σκοπός της εργασίας είναι κάθε ενδιαφερόμενος είτε επαγγελματίας φυσικός είτε ερασιτέχνης να έχει πρόσβαση μέσα από την εφαρμογή στα δεδομένα που συλλέγονται από τους παραπάνω σταθμούς ενώ στόχοι αποτελούν η εφαρμογή αυτή να είναι ανεξάρτητη λειτουργικού και φυλλομετρητή από την πλευρά των χρηστών ενώ παράλληλα η καθυστέρηση παρουσίασης των αποτελεσμάτων από τα ερωτήματα που θα εισάγει ο χρήστης να είναι αμελητέα ανεξαρτήτως bandwidth της σύνδεσης.

Η παρουσίαση της πτυχιακής χωρίζεται σε τρία διακριτά μέρη.

Στο πρώτο μέρος γίνεται αναφορά στο θεωρητικό μέρος της μετεωρολογίας καθώς και των οργάνων που χρησιμοποιούνται για τη μέτρηση των διάφορων ατμοσφαιρικών στοιχείων που υπόκεινται σε μελέτη.

Στο δεύτερο μέρος αναφέρονται οι γλώσσες προγραμματισμού και τα εργαλεία εκείνα τα οποία χρησιμοποιήθηκαν για την υλοποίηση της διαδικτυακής εφαρμογής.

Στο τρίτο μέρος της πτυχιακής γίνεται παρουσίαση και επεξήγηση τμημάτων κώδικα ο οποίος αναπτύχθηκε κατά την υλοποίηση του προγράμματος.

Τέλος, στο παράρτημα υπάρχουν επιπρόσθετα βοηθητικά στοιχεία σχετικά με την εφαρμογή.

Το σύστημα το οποίο δημιουργήθηκε είναι εγκατεστημένο στον server του ΑΤΕΙ και λειτουργεί ως υπηρεσία Online ανάκτησης ατμοσφαιρικών μετρήσεων σε πραγματικό χρόνο.

Η υλοποίηση της ιστοσελίδας έγινε με τη μικρή ή εκτεταμένη χρήση των τεχνολογιών HTML, PHP, CSS, JavaScript, JQuery, SQL Server και MySQL.

Σημαντικός περιορισμός ήταν η χρήση από τους σταθμούς μέτρησης της SQL Server βάσης δεδομένων αντί της MySQL που συνήθως χρησιμοποιείται σε εφαρμογές διαδικτύου που συνδυάζονται με PHP. Παρόλες τις δυσκολίες όμως το πεδίο αυτό προσφέρθηκε για έρευνα ενώ όλες οι σχετικές πληροφορίες για τα θέματα και τις δυσκολίες που προέκυψαν παρατίθενται στο μέρος της θεωρητικής ανάλυσης της πτυχιακής εργασίας.

ΜΕΡΟΣ ΠΡΩΤΟ

ΚΕΦΑΛΑΙΟ 1^ο

1. Βασικά στοιχεία μετεωρολογίας

1.1 Εισαγωγή

Στο κεφάλαιο αυτό περιγράφεται συνοπτικά ο κλάδος της μετεωρολογίας ως φυσική επιστήμη καθώς επίσης μερικά από τα βασικά χαρακτηριστικά που πρόσκεινται σε αυτήν.

1.2 Τι είναι μετεωρολογία

Η μετεωρολογία αποτελεί έναν από τους κλάδους των φυσικών επιστημών και έχει ως κύριο αντικείμενο την έρευνα της ατμόσφαιρας καθώς και των φαινομένων που λαμβάνουν χώρα σε αυτήν. Τα σημαντικότερα από αυτά είναι η ατμοσφαιρική πίεση, οι μεταβολές της θερμοκρασίας, οι μετακινήσεις αερίων μαζών, η υγρασία, η εξάτμιση, ο σχηματισμός και η εξέλιξη των νεφών, η συμπύκνωση και υγροποίηση των υδρατμών, τα ατμοσφαιρικά κατακρημνίσματα, οι μορφές απόθεσης καθώς και ο ατμοσφαιρικός ηλεκτρισμός.

Στη μελέτη αυτών των φαινομένων, η Μετεωρολογία αναζητά τις γενεσιουργές αιτίες αυτών καθώς και τους παράγοντες που συμβάλουν στη δημιουργία ή και τροποποίηση αυτών, φθάνοντας στο σημείο της σχετικής χάραξης αυτών σε χάρτες και από την μικρή χρονικά πρόβλεψη να φθάσει στην ασφαλή πρόβλεψη για όσο το δυνατόν πιο μακρυπρόθεσμο χρονικό διάστημα κάτι που αποτελεί και τον κυριότερο στόχο της.

Ιδιαίτεροι κλάδοι αυτής της Επιστήμης είναι η Πρακτική Μετεωρολογία, η Δυναμική Μετεωρολογία, η Αερολογία (που προβαίνει στις μετρήσεις), η Περιγραφική Μετεωρολογία, η Ναυτική Μετεωρολογία (που αφορά κυρίως τον τομέα της ναυσιπλοΐας), η Κλιματολογία, η Γεωργική Μετεωρολογία, η Φαινομενολογία και η Βιοκλιματολογία .

Η σύγχρονη μετεωρολογία διακρίνεται στις εξής τρεις κατηγορίες:

- 1) Η δυναμική μετεωρολογία που εξετάζει τις κινήσεις του αέρα και τις δυνάμεις που τις προκαλούν.
- 2) Η φυσική μετεωρολογία που εξετάζει τα καθαρώς φυσικά φαινόμενα όπως η ακτινοβολία, η θερμότητα, η εξάτμιση, τα οπτικά, ακουστικά και ηλεκτρικά φαινόμενα κ.ά.
- 3) Η κλιματολογία ή στατική μετεωρολογία που πεδίο ενασχόλησής της αποτελεί η συγκέντρωση και επεξεργασία των διαστημικών στοιχείων για το κλίμα διαφόρων περιοχών της γης.

1.3 Ιστορικά στοιχεία

Ιστορικά, το 1922 πρώτος ο μαθηματικός Λιούις Φράι Ρίτσαρντσον είχε την ιδέα της περιγραφής των φυσικών διαδικασιών οι οποίες συμβαίνουν στην ατμόσφαιρα χρησιμοποιώντας τις εξισώσεις της υδροδυναμικής και των ρευστών. Στο έργο του “Η πρόβλεψη του καιρού με αριθμητικές μεθόδους” περιέγραψε τις μαθηματικές μεθόδους μέσω των οποίων ο άνθρωπος θα μπορούσε να προβλέψει την εξέλιξη του καιρού με τα τεχνικά μέσα της εποχής εκείνης.

Η ιδέα του απαιτούσε την ύπαρξη ενός μεγάλου αριθμού ανθρώπων, οι οποίοι χρησιμοποιώντας χαρτί, μολύβι και αριθμομηχανές γραφείου για την εκτέλεση των μαθηματικών πράξεων, θα επέλυαν τις μαθηματικές εξισώσεις που περιγράφουν τις κινήσεις της ατμόσφαιρας και θα προέβλεπαν την εξέλιξη του καιρού. Υπολόγισε μάλιστα, ότι απαιτούνταν 64.000 άνθρωποι προκειμένου να επιτευχθεί η πρόβλεψη του καιρού με την ίδια ταχύτητα με την οποία αυτός εξελίσσεται στην πραγματικότητα. Είχε μάλιστα διατυπώσει την άποψη ότι ίσως στο μακρινό μέλλον το ανθρώπινο γένος θα είχε τη δυνατότητα να προχωρεί στους υπολογισμούς γρηγορότερα από ότι προχωρεί ο καιρός.

Προφητικά λόγια, μόνο που το μακρινό μέλλον ήταν μόλις 40 χρόνια μετά. Στα επόμενα χρόνια μέχρι και το 1960 ο επιστημονικός κόσμος περιφρονούσε την πρόγνωση και δεν εμπιστευόταν τους υπολογιστές. Για τους περισσότερους η πρόγνωση ήταν κάτι λιγότερο από επιστήμη. Ήταν μια εμπειρική δουλειά που γινόταν από πρακτικούς μετεωρολόγους, οι οποίοι μέσω εμπειρίας αντιλαμβάνονταν τον καιρό της επόμενης ημέρας.

Εκείνη την εποχή ο Λόρεντζ ο οποίος εργαζόταν στο Τεχνολογικό Ινστιτούτο της Μασαχουσέτης είχε μια τελείως διαφορετική άποψη η οποία περιλάμβανε τις ήδη γνωστές μαθηματικές θεωρίες που χρησιμοποιούνταν στη Διαστημική. Έτσι, μετά από συνεχείς προσπάθειες και με τα πενιχρά τεχνολογικά μέσα της εποχής κατάφερε με τον πρωτόγονο υπολογιστή του να δημιουργήσει ένα απλό μοντέλο της ατμόσφαιρας που έμελλε να αποτελέσει τον πρόγονο των σημερινών ατμοσφαιρικών μοντέλων.

Στα επόμενα χρόνια ο επιστημονικός κόσμος γέμισε ελπίδες για τη δυνατότητα πρόγνωσης του καιρού. Ακολούθησαν αρκετές προσπάθειες μέχρι που το 1980 η αριθμητική πρόγνωση του καιρού έγινε πλέον μία αναμφισβήτητη πραγματικότητα. Οι μεγαλύτερες επιτυχίες για την ανάπτυξη αριθμητικών μοντέλων καιρού εκείνη την περίοδο, είχαν σημειωθεί στο Ρέντινγκ της Αγγλίας, μια μικρή επαρχιακή πόλη λίγο έξω από το Λονδίνο όπου στεγαζόταν και συνεχίζει να στεγάζεται το Ευρωπαϊκό Κέντρο Μεσοπρόθεσμων Προγνώσεων (European Center for Mesoscale Weather Forecasts - ECMWF).

Οι Ευρωπαίοι έδειξαν την πρωτοπορία τους στο συγκεκριμένο τομέα εγκαθιστώντας τον υπερυπολογιστή CRAY και προσλαμβάνοντας το καλύτερο επιστημονικό προσωπικό των κρατών μελών του ECMWF. Στην άλλη άκρη του Ατλαντικού ανάλογες προσπάθειες κατέβαλαν και οι Αμερικάνοι. Ο υπερυπολογιστής που διέθεταν - ένας Cyber 205 της Control Data – εκτελούσε εκατομμύρια πράξεις το δευτερόλεπτο και επέλυε τα συστήματα των 500.000 εξισώσεων του μοντέλου σε σύντομο χρονικό διάστημα. Τα αποτελέσματα ήταν αρκετά ικανοποιητικά και τα μηνύματα συνεχώς πιο αισιόδοξα.

Από τότε μέχρι σήμερα έχουν περάσει σχεδόν 20 χρόνια συνεχών προσπαθειών και η πρόοδος που έχει σημειωθεί θεωρείται αλματώδης. Σ' αυτό έχει συμβάλει σημαντικά και η εξέλιξη των υπερυπολογιστών οι οποίοι σήμερα πλέον διαθέτουν ένα μεγάλο αριθμό επεξεργαστών τεραστίων δυνατοτήτων. Η ταχύτητα ενός μόνο επεξεργαστή μετρείται πλέον σε εκατομμύρια πράξεις το δευτερόλεπτο. Η ατμόσφαιρα προσομοιώνεται συνεχώς με πιο ρεαλιστικό τρόπο αφού οι αριθμητικές μέθοδοι διαρκώς βελτιώνονται, το πλέγμα των σημείων γίνεται ολοένα και πιο πυκνό, τα υπό μελέτη στρώματα της ατμόσφαιρας αυξάνονται, εισάγονται λεπτομερείς πληροφορίες ορογραφίας, βλάστησης και τύπου εδάφους κ.τ.λ. Με τον τρόπο αυτό τα αποτελέσματα που λαμβάνονται είναι συνεχώς πιο αξιόπιστα και πιο αναλυτικά. Βέβαια θα πρέπει να τονιστεί ότι με τα μοντέλα καιρού η

επιστήμη απλά προσπαθεί να προσεγγίσει ένα πολύ σύνθετο πρόβλημα όσο καλύτερα μπορεί και στο μικρότερο δυνατό χρονικό διάστημα .

1.4 Προγνώσεις καιρού

Η πρόγνωση του καιρού είναι ο κλάδος της Συνοπτικής Μετεωρολογίας που έρχεται να ικανοποιήσει την ανάγκη για την όσο το δυνατόν καλύτερη γνώση του καιρού που αναμένεται ότι θα επηρεάσει κάποια συγκεκριμένη περιοχή και για κάποιο ορισμένο χρονικό διάστημα. Είναι από την φύση της τόσο μαθηματικό όσο και φυσικό πρόβλημα, με εφαρμογή στη Μετεωρολογία. Για να επιλυθεί το πρόβλημα της πρόγνωσης γίνεται χρήση των θεμελιωδών αρχών της Φυσικής, ενώ απαιτείται ένα σύνολο επιστημονικών μεθόδων και αναλύσεων καθώς και ικανή υπολογιστική ισχύς. Ταυτόχρονα απαιτούνται παγκόσμιες μετεωρολογικές παρατηρήσεις του παρόντος καιρού.

Οι μετεωρολογικές παρατηρήσεις συλλέγονται παγκόσμια σε τακτά χρονικά διαστήματα από συγκεκριμένων προδιαγραφών όργανα και σύμφωνα με συγκεκριμένο πρωτόκολλο (κανονισμοί του Παγκόσμιου Μετεωρολογικού Οργανισμού - WMO) και αφορούν συγκεκριμένα επίπεδα σε διαφορετικά ύψη στην κατώτερη ατμόσφαιρα (τροπόςφαιρα). Οι παρατηρήσεις αυτές αποστέλλονται, με βάση συγκεκριμένο πρωτόκολλο, το οποίο ορίζεται επίσης από τον WMO, σε τράπεζες μετεωρολογικών δεδομένων για αξιοποίηση από μεγάλα προγνωστικά κέντρα του εξωτερικού. Τέτοια κέντρα είναι το προγνωστικό κέντρο της Τουλούζης στην Γαλλία, του Όφενμπαχ στην Γερμανία, του Έξετερ στην Αγγλία, της Ουάσιγκτον στις ΗΠΑ και αλλού. Αυτά τα προγνωστικά κέντρα διαθέτουν τεράστια υπολογιστική ισχύ (υπέρ υπολογιστές) και έχουν την δυνατότητα να επιλύσουν το μαθηματικό – φυσικό κομμάτι της πρόγνωσης καιρού σε σχετικά μικρό χρονικό διάστημα. Το φυσικομαθηματικό κομμάτι της πρόγνωσης καιρού έγκειται στην επίλυση συγκεκριμένων μαθηματικών εξισώσεων, σε ένα τεράστιο πλήθος σημείων πλέγματος, (εκεί προσαρμόζονται οι μετεωρολογικές παρατηρήσεις) σε συγκεκριμένα ύψη στην τροπόςφαιρα. Τα αποτελέσματα των υπολογισμών αυτών ονομάζονται προϊόντα αριθμητικής πρόγνωσης καιρού - NWP. Τα προϊόντα αυτά παράγονται σε γραφική μορφή έχουν χρόνο πρόγνωσης μέχρι και 10 μέρες και

διαφέρουν από τα κλιματικά μοντέλα που έχουν χρόνο πρόγνωσης της τάξης του έτους ή και περισσότερο.

Οι μετεωρολογικές υπηρεσίες όλων των χωρών συνεργάζονται μέσω του Παγκόσμιου Μετεωρολογικού Οργανισμού και εκτελούν τακτικές παρατηρήσεις του καιρού σε όλη τη γη σε συγκεκριμένες χρονικές στιγμές. Σε ώρα Greenwich εκτελούνται παρατηρήσεις στις 00.00, 03.00, 06.00, 09.00, 12.00, 18.00, 21.00 UTC. Επίσης κάθε 6 ώρες γίνονται παρατηρήσεις ανώτερης ατμόσφαιρας. Οι παρατηρήσεις αυτές καταγράφονται, κωδικοποιούνται και μέσω του Παγκόσμιου μετεωρολογικού συστήματος διακίνησης είναι διαθέσιμες σε κάθε μετεωρολογική υπηρεσία.

Οι παρατηρήσεις καιρού όταν συγκεντρωθούν ελέγχονται ως προς την ορθότητα μέσω προγραμμάτων Η/Υ και κωδικοποιούνται πάνω σε χάρτη καιρού. Στη συνέχεια ο μετεωρολόγος κάνει ανάλυση του χάρτη και προσδιορίζει τα βαρομετρικά χαμηλά, υψηλά και τα μέτωπα καιρού. Στη συνέχεια από τα στοιχεία των προγνωστικών μοντέλων καιρού, μελετά τις νέες θέσεις των συστημάτων καιρού και μετώπων σαν συνάρτηση του χρόνου.

Από την μελέτη αυτή ο μετεωρολόγος βγάζει χρήσιμα συμπεράσματα σχετικά με την μελλοντική εξέλιξη του καιρού και επομένως κάνει μία πρόγνωση. Η πρόγνωση μπορεί να γίνει μέχρι 10 ημέρες ενώ γίνεται μεγάλη προσπάθεια για εποχιακή πρόγνωση του καιρού βασισμένη σε αποτελέσματα συνδυασμένων μοντέλων καιρού και θαλασσίων μοντέλων.

1.5 Μοντέλα πρόγνωσης καιρού

Ένα ατμοσφαιρικό μοντέλο πρόγνωσης αποτελεί στην πραγματικότητα ένα σύστημα προσομοίωσης των φυσικών διαδικασιών που λαμβάνουν χώρα στην ατμόσφαιρα. Η ατμόσφαιρα προσομοιώνεται με διακριτά σημεία (grid points) σε συγκεκριμένα επίπεδα (levels) που αρχίζουν από το έδαφος και τελειώνουν στα ανώτερα σημεία της ατμόσφαιρας. Με τον τρόπο αυτό δημιουργείται ένα τρισδιάστατο πλέγμα σημείων πάνω στο οποίο πραγματοποιούνται όλοι οι απαιτούμενοι υπολογισμοί.

Επίσης, γίνεται η παραδοχή ότι ο χρόνος κυλά με πολύ μικρά χρονικά βήματα της τάξης των λίγων δευτερολέπτων ή ακόμα και χιλιοστών του δευτερολέπτου. Η

συνεχής επανάληψη των υπολογισμών σε κάθε χρονικό βήμα οδηγεί στην πρόγνωση του καιρού για την επόμενη ημέρα ή για την επόμενη εβδομάδα. Όσο πυκνότερο είναι το δίκτυο των σημείων και όσο πιο ρεαλιστική είναι η προσομοίωση της ατμόσφαιρας από το σύστημα των εξισώσεων που την περιγράφουν, τόσο πιο ικανοποιητικά είναι τα αποτελέσματα αλλά παράλληλα τόσο μεγαλύτερη είναι και η απαιτούμενη υπολογιστική ισχύς.

1.6 Μετεωρολογικά φαινόμενα

Τα βασικά μετεωρολογικά φαινόμενα, από τα οποία συνιστώνται όλα τα υπόλοιπα είναι η θερμοκρασία, η ατμοσφαιρική πίεση και η υγρασία της ατμόσφαιρας. Ακολουθεί μία συνοπτική αναφορά σε κάθε ένα από αυτά.

1.6.1 Η θερμοκρασία

Η θερμοκρασία είναι το κυριότερο χαρακτηριστικό της ατμόσφαιρας καθώς είναι ο κύριος συντελεστής, ο οποίος καθορίζει το κλίμα, τη βλάστηση και κατ' επέκταση τη δυνατότητα ανάπτυξης και ύπαρξης ζωής σε ένα τόπο. Η θερμοκρασία μετριέται σε βαθμούς Κελσίου ($^{\circ}\text{C}$). Η μετεωρολογία χρησιμοποιεί εκτός από τα κοινά ενδεικτικά θερμόμετρα ειδικά θερμόμετρα όπως τα θερμόμετρα μεγίστου και ελαχίστου, τα οποία δείχνουν τη μέγιστη και την ελάχιστη θερμοκρασία από την τελευταία τους ρύθμιση. Έτσι, αν ρυθμιστούν σε ημερήσιο μηδενισμό θα μετράνε τη μέγιστη και την ελάχιστη θερμοκρασία του 24ώρου. Η μετεωρολογία χρησιμοποιεί επίσης θερμογράφους οι οποίοι καταγράφουν συνεχώς την επικρατούσα θερμοκρασία επί μιας βαθμονομημένης χάρτινης ταινίας.

Η θερμοκρασία ελαττώνεται σε συνάρτηση με το ύψος κατά $0,6^{\circ}\text{C}$ ανά 100 μέτρα ύψους για τα πρώτα 10.000 μέτρα ενώ έπειτα από το ύψος αυτό η θερμοκρασία αρχίζει να αυξάνεται.

1.6.2 Η ατμοσφαιρική πίεση

Η γη περιβάλλεται από ένα λεπτό στρώμα αερίων το οποίο ονομάζεται ατμόσφαιρα το βάρος της οποίας πιέζει όλα τα σώματα που βρίσκονται μέσα σε αυτή. Η πίεση αυτή που ασκείται ονομάζεται ατμοσφαιρική πίεση.

Η πίεση μετριέται σε χιλιοστόμετρα στήλης υδραργύρου και σημείο αναφοράς αποτελεί η επιφάνεια της θάλασσας στην οποία επικρατεί πίεση περίπου 760 χιλιοστών Hg. Στη μετεωρολογία συνηθίζεται επίσης να χρησιμοποιείται η μονάδα μιλλιμπάρ (mb) (αντί για Pascal) για την οποία η πίεση στην επιφάνεια της θάλασσας είναι 1.013,25 mb. Η επικράτηση του μιλλιμπάρ ως μονάδα ατμοσφαιρικής πίεσης έγινε το 1929 και λόγω συνήθειας διατηρήθηκε και μετά το 1960 όταν αποφασίστηκε η αλλαγή των μονάδων και η εφαρμογή του συστήματος International System of Units (S.I.).

Όπως συμβαίνει και με τη θερμοκρασία η πίεση ελαττώνεται σε μεγαλύτερα ύψη ενώ παράλληλα όσο πιο θερμός είναι ο ατμοσφαιρικός αέρας τόσο μικρότερη θα είναι η πίεση και το αντίθετο. Επίσης, αν ο αέρας έχει κατεύθυνση προς το έδαφος έχει ως αποτέλεσμα την πίεση του εδάφους με αντίστοιχη αύξηση της πίεσης ενώ με κατεύθυνση προς τα πάνω η πίεση ελαττώνεται. Μάλιστα, στην ουσία ο άνεμος είναι η μετακίνηση του αέρα που προκαλείται από τη διαφορά της ατμοσφαιρικής πίεσης που υπάρχει σε δύο διαφορετικούς τόπους.

1.6.3 Η ατμοσφαιρική υγρασία

Ο ατμοσφαιρικός αέρας περιέχει υδρατμούς σε διαφορετική ποσότητα από τόπο σε τόπο και από ώρα σε ώρα. Ο αέρας όμως δεν είναι δυνατόν να περιέχει απεριόριστη ποσότητα υδρατμών, αλλά για κάθε θερμοκρασία υπάρχει μία μέγιστη δυνατή περιεκτικότητα υδρατμών. Όταν ο αέρας περιέχει τη μέγιστη τέτοια ποσότητα ονομάζεται κορεσμένος.

Όσο ψυχρότερος είναι ο αέρας τόσο μικρότερη ποσότητα υδρατμών μπορεί να συγκρατήσει. Αν λοιπόν μία μάζα υγρού και θερμού αέρα ψυχθεί θα φθάσει σε μια θερμοκρασία όπου δεν είναι δυνατόν πλέον να συγκρατήσει όλους τους υδρατμούς τους οποίους περιέχει. Οι υδρατμοί που περισσεύουν θα συμπυκνωθούν ως σταγονίδια πάνω στα αιωρούμενα μικροσωματίδια και θα δημιουργήσουν το νέφος.

Η υγρασία μετριέται με τα υγρόμετρα τα οποία δείχνουν το ποσοστό των υδρατμών που περιέχεται στην ατμόσφαιρα όπου το 100% θεωρούνται οι υδρατμοί οι οποίοι περιέχονται στην ίδια θερμοκρασία σε κατάσταση κορεσμού.

1.7 Επίλογος

Στο πρώτο κεφάλαιο περιγράφηκαν τα βασικά στοιχεία της ατμόσφαιρας τα οποία μελετώνται από τους δύο σταθμούς οι οποίοι είναι το αντικείμενο της πτυχιακής. Στο επόμενο κεφάλαιο θα περιγραφούν οι σταθμοί μέτρησης καθώς και τα τμήματα από τα οποία αυτοί αποτελούνται.

ΚΕΦΑΛΑΙΟ 2^ο

2. Ανάλυση σταθμού μέτρησης

2.1 Εισαγωγή

Στο κεφάλαιο αυτό αναλύονται οι σταθμοί Αρχιμήδης και Ανδρομέδα τους οποίους εξετάζει η παρούσα εργασία και παρουσιάζονται τα κύρια τεχνικά τμήματα και χαρακτηριστικά τους όπως αυτά είναι εγκατεστημένα στους φυσικούς τους χώρους.

2.2 Κύρια λειτουργία σταθμού περιβαλλοντικού ελέγχου

Η γενικές λειτουργίες κάθε σταθμού περιβαλλοντικού ελέγχου, χωρίζονται στους παρακάτω τομείς:

- Συλλογή δεδομένων από του αισθητήρες στον τοπικό σταθμό μέτρησης.
- Κατάλληλη επεξεργασία και μετατροπή των τιμών αυτών.
- Αποστολή των τιμών στον κεντρικό σταθμό ελέγχου.
- Λήψη των τιμών και απεικόνιση τους στο σύστημα Εποπτικού Ελέγχου και Απόκτησης Στοιχείων (Supervisory Control And Data Acquisition - SCADA).
- Συνεχής επιτήρηση των τιμών και άμεση ενημέρωση σε περίπτωση σφαλμάτων.
- Καταγραφή των τιμών σε βάση δεδομένων για περαιτέρω επεξεργασία και μελέτη.
- Γραφική απεικόνιση όλων των στοιχείων και καταγραφή ιστορικού για ολοκληρωμένη ενημέρωση του χρήστη.
- Δυνατότητα δημιουργίας συσχετισμένων διαγραμμάτων για εξαγωγή συμπερασμάτων σχετικά με την μεταβολή των τιμών αλληλοεξαρτώμενων στοιχείων.
- Ιστορικό αναφοράς όλων των συμβάντων για την συνεχή επιτήρηση του σταθμού περιβαλλοντικού ελέγχου.

2.3 Ο σταθμός μέτρησης Αρχιμήδης

Ο σταθμός μέτρησης Αρχιμήδης ο οποίος είναι αντικείμενο της παρούσας εργασίας έχει το ρόλο παρατήρησης του εναέριου χώρου της Θεσσαλονίκης όσον αφορά τις μεταβολές των στοιχείων του καιρού της ευρύτερης περιοχής. Πιο συγκεκριμένα, λαμβάνει ενδείξεις για τη θερμοκρασία, την υγρασία του αέρα, την ταχύτητα και τη φορά του ανέμου καθώς και το ποσοστό της βροχόπτωσης.

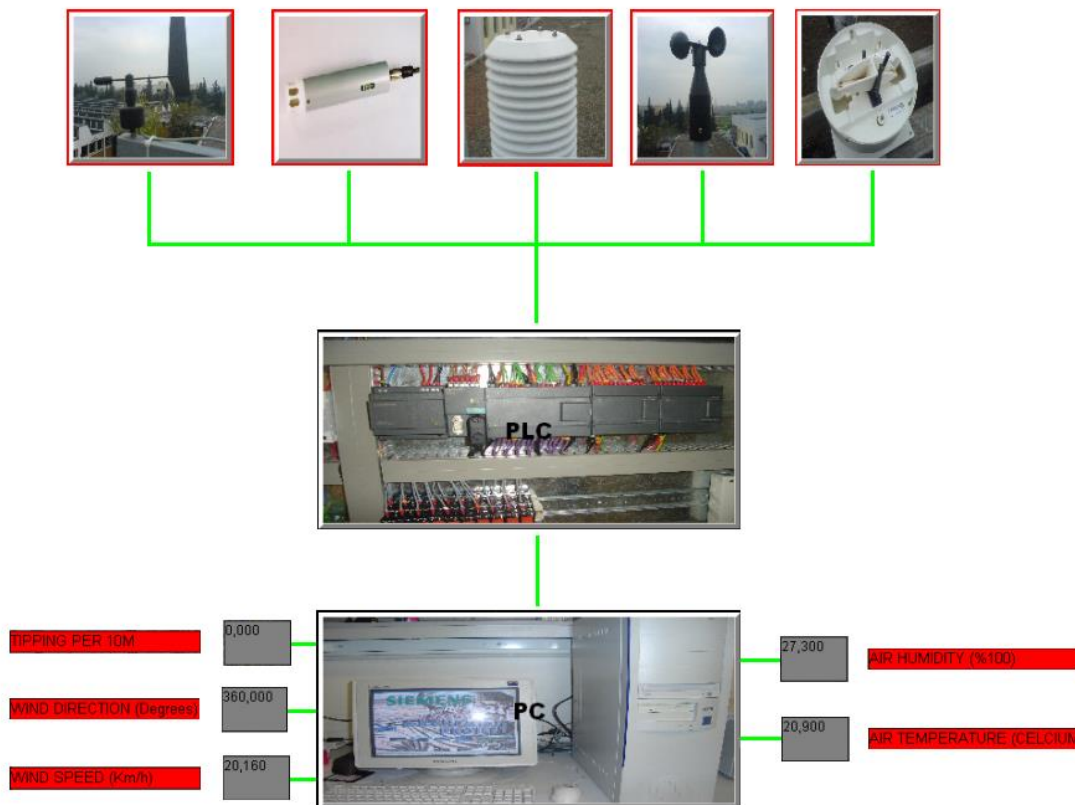
Ο σταθμός αυτός είναι εγκατεστημένος στο κτίριο της Ρομποτικής του τμήματος Αυτοματισμού του Α.Τ.Ε.Ι. Θεσσαλονίκης, όπου έχουν τοποθετηθεί τα αισθητήρια για τη μέτρηση των τιμών των παραπάνω στοιχείων.

Η Εικόνα 1 δείχνει τη φυσική εγκατάσταση του κεντρικού πίνακα που βρίσκεται στο χώρο του ΑΤΕΙ. Όλοι οι αισθητήρες συνδέονται σε αυτόν τον πίνακα ο οποίος βρίσκεται κάτω από την εγκατάσταση και ο οποίος στέλνει τα δεδομένα τα οποία συλλέγει σε έναν Προγραμματιζόμενο Λογικό Ελεγκτή (Programmable Logic Controller - PLC).



Εικόνα 1: Πίνακας ελεγκτή

Οι αισθητήρες οι οποίοι χρησιμοποιούνται για τη λήψη των ατμοσφαιρικών τιμών των στοιχείων που μελετώνται φαίνονται στην Εικόνα 2. Οι αισθητήρες βρίσκονται στο εξωτερικό περιβάλλον, λαμβάνουν τις τιμές τις οποίες προορίζονται και είναι προγραμματισμένοι να καταγράψουν, έπειτα στέλνουν τις τιμές αυτές στον κεντρικό ελεγκτή όπου γίνεται η μετατροπή τους και τέλος αυτές αποθηκεύονται σε μία βάση δεδομένων στον τοπικό ηλεκτρονικό υπολογιστή. Σημειώνεται ότι δεν είναι απαραίτητο η αποθήκευση να γίνεται τοπικά αλλά μέσω μίας απλής σύνδεσης Internet υπάρχει η δυνατότητα τα δεδομένα να αποστέλλονται σε απομακρυσμένη βάση δεδομένων.



Εικόνα 2: Αισθητήρια, πίνακας, Η/Υ και δεδομένα

Όπως φαίνεται και στην Εικόνα 2 οι ληφθέντες τιμές μετατρέπονται δυναμικά σε πραγματικό χρόνο σε τιμές αποδεκτές από τον υπολογιστή ώστε να μπορεί να γίνει η αποθήκευσή τους.

2.4 Αισθητήρες του σταθμού Αρχιμήδης

Παρακάτω γίνεται συνοπτική παρουσίαση των αισθητήρων του περιβαλλοντικού σταθμού Αρχιμήδης ώστε παρόλα τα ιδιαίτερα τεχνικά χαρακτηριστικά τους ο αναγνώστης να έχει μία γενική ιδέα σχετικά με τη λειτουργία του.

2.4.1 Αισθητήρας κατεύθυνσης αέρα

Στη βάση του αισθητηρίου κατεύθυνσης αέρα υπάρχει μία βάνα η οποία είναι πολύ ελαφριά και χαρακτηρίζεται από ταχεία ανταπόκριση σε μεταβολές. Μέσα στη βάνα βρίσκεται ένα ρουλεμάν από ανοξείδωτο χάλυβα. Ο αισθητήρας αυτός είναι αναλογικός και η ρύθμισή του γίνεται με τη βοήθεια πυξίδας, περιστρέφοντας την ακίδα που βρίσκεται στη βάση του αισθητηρίου προς το βορρά, σύμφωνα με την υπόδειξη της πυξίδας.

Το ποτενσιόμετρο, που μέσα στη βάση του αισθητήρα, είναι τύπου long-life και μπορεί να δεχθεί περισσότερες από $20 \cdot 10^6$ περιστροφές.

2.4.2 Αισθητήρας ταχύτητας αέρα

Ο αισθητήρας ταχύτητας αέρα μετράει την οριζόντια ταχύτητα του αέρα και η μετρούμενη τιμή περνάει ως έξοδος σε ψηφιακή μορφή.

Η βάση του είναι από αλουμίνιο και το επάνω μέρος του από συνθετικές ίνες. Τρία “πιάτα” μέσω βραχιόνων συνδέονται στον άξονα του αισθητήρα, ο οποίος αποτελείται από ρουλεμάν, και κατευθύνουν ένα μαγνήτη. Έτσι δημιουργείται μία κυκλική συχνότητα η οποία είναι ανάλογη του αριθμού περιστροφών των “πιάτων”. Η συχνότητα αυτή είναι σχεδόν γραμμική ως προς την ταχύτητα του ανέμου.

2.4.3 Αισθητήρας θερμοκρασίας – υγρασίας αέρα

Ο αισθητήρας θερμοκρασίας - υγρασίας του αέρα είναι ένας σύνθετος αισθητήρας, δεδομένου ότι μετράει ταυτόχρονα δύο διαφορετικά μεγέθη. Στο εσωτερικό της συσκευής βρίσκεται το όργανο μέτρησης. Ο λόγος που βρίσκεται στο εσωτερικό είναι ότι ο σκοπός του οργάνου είναι να μετράει τη θερμοκρασία και την υγρασία του αέρα. Σε περίπτωση που το όργανο αυτό βρισκόταν εκτεθειμένο στο εξωτερικό περιβάλλον, οι μετρήσεις δε θα ήταν αυτές που επιδιώκονται. Με τον όρο θερμοκρασία αέρα εννοείται η θερμοκρασία υπό σκιάν του περιβάλλοντος αέρα, η οποία διαφέρει από αυτή σε ένα μέρος που εκτίθεται στον ήλιο.

Το όργανο αυτό μπορεί να χρησιμοποιηθεί σε εσωτερικό και σε εξωτερικό περιβάλλον, με βροχή ή όχι, ακόμη και μέσα σε νερό, αν είναι μερικώς βυθισμένο. Είναι αρκετά γρήγορο και μπορεί να ανταποκριθεί σε μεταβολές με βήμα της τάξης του 50% έως 90% σε λιγότερο από 2 δευτερόλεπτα, αν και τόσο γρήγορες μεταβολές δε συμβαίνουν στη φύση.

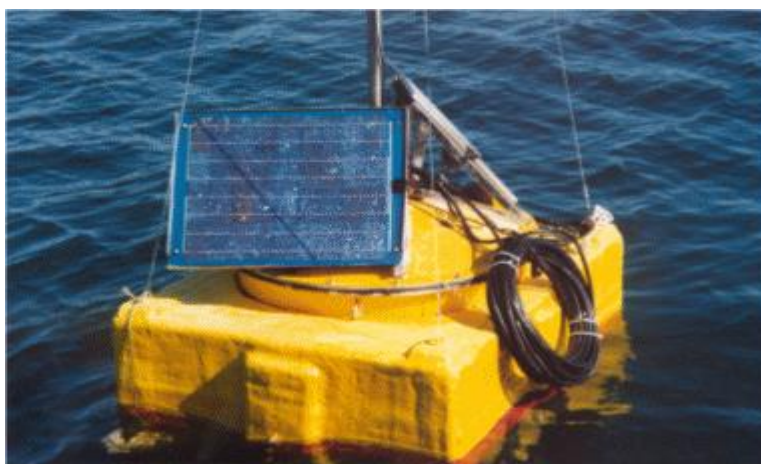
2.4.4 Αισθητήρας βροχόπτωσης

Η ποσότητα του βρόχινου νερού το οποίο συλλέγεται κατά τη διάρκεια της βροχής, οδηγείται μέσω μίας χοάνης στο εσωτερικό του οργάνου όπου υπάρχουν δύο συλλέκτες οι οποίοι είναι σε μία “τραμπάλα” και κάθε φορά που γεμίζει ο ένας, λόγω του βάρους του νερού γέρνει και ακουμπάει σε μία επαφή ανασηκώνοντας έτσι τον άλλον. Το νερό απομακρύνεται από μόνο του χωρίς να χρειαστεί κάποια εξωτερική παρέμβαση. Κάθε φορά που αδειάζει ένας συλλέκτης στέλνεται ηλεκτρονικό σήμα μέσω της επαφής κι έτσι μπορεί να υπολογιστεί το ύψος της βροχής μετρώντας τα αδειάσματα των συλλεκτών σε ένα δεκάλεπτο και πολλαπλασιάζοντας αυτά με την τιμή 0,25.

Για να είναι σωστές οι μετρήσεις θα πρέπει οι συλλέκτες να αδειάζουν την ίδια ποσότητα νερού. Για το λόγο αυτόν πρέπει το αισθητήριο να τοποθετηθεί σε οριζόντιο επίπεδο και να μη γέρνει προς κάποια κατεύθυνση. Έτσι, στο εσωτερικό του υπάρχουν ενδείκτες με σταγόνες, όπως στο αλφάδι που χρησιμοποιείται στις οικοδομές, για να είναι δυνατή η σωστή τοποθέτησή του καθώς και ο έλεγχος της σωστής του θέσης ανά πάσα στιγμή.

2.5 Δίκτυο Ανδρομέδα

Το σύστημα Ανδρομέδα είναι εγκαταστημένο στο Θερμαϊκό Κόλπο της Θεσσαλονίκης και αποτελείται από τρεις τοπικούς σταθμούς μέτρησης και από ένα κεντρικό σταθμό συλλογής δεδομένων. Η αναγκαία ισχύς στους αισθητήρες και τα ηλεκτρονικά κυκλώματα παρέχεται από μπαταρίες και ηλιακά κύτταρα. Οι αισθητήρες μετρούν θερμοκρασία νερού, pH, αγωγιμότητα αλμυρότητα, ποσότητα διαλυμένου οξυγόνου και θολότητα στο θαλασσινό νερό σε καθορισμένες χρονικές στιγμές. Το προγραμματιζόμενο λογικό κύκλωμα παρακολουθεί τη λειτουργία του σταθμού μέτρησης, αποθηκεύει τις μετρήσεις και ελέγχει τις επικοινωνίες δια μέσου ενός ασύρματου δικτύου κεντρικό σταθμό συλλογής δεδομένων.



Εικόνα 3: Ένας από τους τρεις σταθμούς μέτρησης του συστήματος Ανδρομέδα

Ο κεντρικός σταθμός συλλογής δεδομένων παρακολουθεί και ελέγχει την επικοινωνία με τους τοπικούς σταθμούς μέτρησης και συλλέγει τις μετρήσεις των αισθητήρων από όλους αυτούς. Τα δεδομένα αποθηκεύονται σε μία βάση δεδομένων για τους σκοπούς της μελλοντικής επεξεργασίας και ανάλυσης. Αποτελείται από έναν υπολογιστή Pentium που λειτουργεί σε περιβάλλον SCADA. Ο υπολογιστής παίζει τον ρόλο ενός master και ελέγχει την επικοινωνία με τους τοπικούς σταθμούς μέτρησης χρησιμοποιώντας το πρωτόκολλο αμφίδρομης χειραψίας. (handshake protocol). Ο ολικός αριθμός των μετρήσεων που μπορούν

να συλλεχθούν είναι μεταξύ 8 και 24 μετρήσεις ημερησίως. Η συχνότητα των μετρήσεων μπορεί να αυξηθεί σε περιπτώσεις επείγουσας ανάγκης. Η πολιτική της επικοινωνίας μειώνει την κατανάλωση ενέργειας των τοπικών σταθμών, αφού αυτοί λειτουργούν μόνο όταν πρέπει να στείλουν δεδομένα στον κεντρικό σταθμό.



Εικόνα 4: Διάγραμμα ροής δεδομένων του συστήματος Ανδρομέδα

Συνοπτικά ο Τοπικός Σταθμός Συλλογής Δεδομένων όπου γίνεται και η πρωταρχική επεξεργασία των μετρήσεων από τους αισθητήρες περιλαμβάνει τρεις βαθμίδες οι οποίες είναι οι εξής:

- Η βαθμίδα λήψης των δεδομένων.
- Η βαθμίδα επεξεργασίας – αποθήκευσης των δεδομένων.
- Η βαθμίδα αποστολής των δεδομένων.

2.6 PLC

Επίσης απαραίτητη κρίνεται η επεξήγηση του PLC αφού είναι πολύ σημαντικό στοιχείο για τους σταθμούς μέτρησης περιβαλλοντικών στοιχείων.

Μία προγραμματιζόμενη μονάδα λογικού ελέγχου (PLC) ή προγραμματιζόμενος ελεγκτής, είναι ένας ψηφιακός υπολογιστής ο οποίος χρησιμοποιείται για την υλοποίηση ενός αυτοματισμού με βάση ένα συγκεκριμένο πρόγραμμα γραμμένο σε μία γλώσσα προγραμματισμού. Χαρακτηριστικά παραδείγματα υλοποίησης αυτοματισμού είναι η αυτοματοποίηση πολυδιάστατων μορφών ηλεκτρομηχανολογικών διεργασιών, όπως ο έλεγχος των μηχανημάτων σε ένα εργοστάσιο συναρμολόγησης, του ελέγχου σε ένα πάρκο αναψυχής ή τον έλεγχο των φωτιστικών σωμάτων καθώς επίσης και σε μηχανές συσκευασίας ημιαγωγών και μηχανών.

Σε αντίθεση με τους υπολογιστές γενικής χρήσης, το PLC είναι σχεδιασμένο για ρύθμιση πολλαπλών εισόδων και εξόδων, ελαχιστοποίηση της διακύμανσης θερμοκρασία στο εσωτερικό όπως και σε ηλεκτρικό θόρυβο, καθώς και η ανθεκτικότητα σε κραδασμούς. Πρώτου σχεδιαστούν οι προγραμματιζόμενοι λογικοί ελεγκτές, η υλοποίηση εγκαταστάσεων αυτοματισμού στηρίζεται κατά το μεγαλύτερο μέρος τους στα συστήματα συρματωμένης λογικής. Τα διάφορα στοιχεία όπως, ρελέ, χρονικά, επαφές, τοποθετούνταν μέσα σε πίνακα και συρματώνονταν σύμφωνα με το συνδεσμολογικό σχέδιο. Σε περίπτωση όμως που ο αυτοματισμός έπρεπε να αλλάξει, τότε έπρεπε να αλλάξει και η συρμάτωση. Με τον σχεδιασμό όμως των PLC αντιμετωπίστηκαν τέτοιου είδους προβλήματα όπως και άλλων ειδών (αντοχή από θερμοκρασία, υγρασία). Η συρμάτωση όλων των εξωτερικών στοιχείων γίνεται πλέον μόνιμα στις κλέμμες του PLC, και όποιας μορφής αλλαγές θα θέλει κάποιος να κάνει αρκεί να αλλάξει το πρόγραμμα, αποφεύγοντας την αλλαγή συνδεσμολογίας του αυτοματισμού. Τα προγράμματα για τον έλεγχο της μηχανής αποθηκεύονται σε μία μπαταρία μεταβλητής ή μη μεταβλητής μνήμης. Ένα PLC είναι ένα παράδειγμα πραγματικού χρόνου του συστήματος, δεδομένου ότι η παραγωγή των αποτελεσμάτων πρέπει να παράγονται, σε απάντηση στην είσοδο, πάντα μέσα σε συνθήκες που οριοθετούνται κάθε φορά, αλλιώς το αποτέλεσμα θα είναι λανθασμένο.

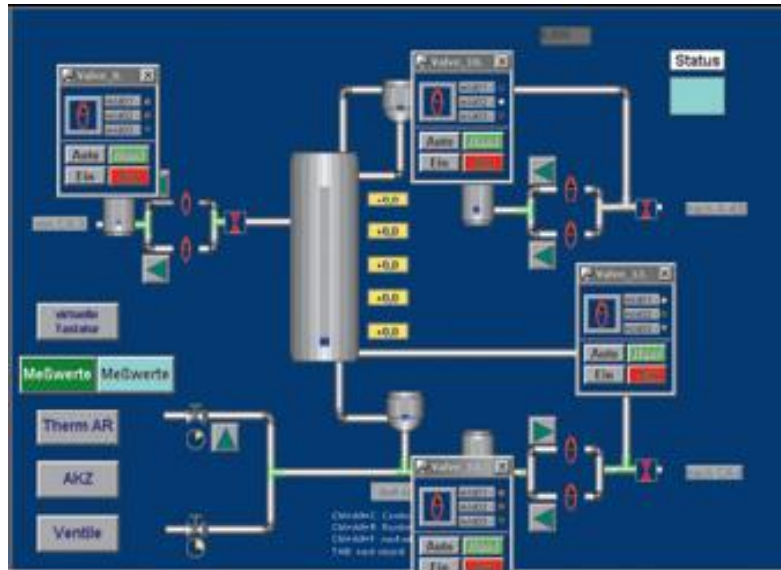
Η κύρια διαφορά από τους υπόλοιπους υπολογιστές είναι ότι το PLC είναι θωρακισμένο για ακραίες συνθήκες (όπως σκόνη, υγρασία, ζέστη, κρύο, κ.λ.π.) και έχουν τη δυνατότητα για εκτενείς ρυθμίσεις εισόδου / εξόδου (I / O), όπως η σύνδεση αισθητήρων και ενεργοποιητών με το PLC. Οι προγραμματιζόμενοι λογικοί ελεγκτές έχουν τη δυνατότητα ανάγνωσης του ορίου διακοπών, της θερμοκρασίας, τη πίεσης, τη θέση πολύπλοκων συστημάτων εντοπισμού θέσης, ακόμα και χρήσης για μηχανική όραση. Όσον αφορά την ενεργοποίηση, τα PLC λειτουργούν είτε σαν ηλεκτρικοί κινητήρες, είτε σαν πνευματικοί ή υδραυλικοί κύλινδροι, είτε σαν αναλογική έξοδος. Οι ρυθμίσεις εισόδων / εξόδων μπορούν να είναι ενσωματωμένες σε ένα απλό PLC, ή μπορεί το PLC να έχει εξωτερικές I / O ενότητες που συνδέονται με ένα δίκτυο υπολογιστών που αυτό με τη σειρά του συνδέεται με το PLC.

2.7 Σύστημα SCADA

Ο λόγος δημιουργίας του δικτύου των PLC και όλος ο προγραμματισμός που απαιτήθηκε, είναι η υλοποίηση ενός συστήματος SCADA. Το SCADA (Supervisory Control And Data Acquisition), σημαίνει Εποπτικός Έλεγχος και Απόκτηση Στοιχείων. Τα συστήματα SCADA αποτελούν εφαρμογή της βιομηχανικής πληροφορικής για την εποπτεία της παραγωγής. Με χρήση των συστημάτων SCADA επιτυγχάνεται online παρακολούθηση, μέσω μονάδων PLCs, και συνεχής καταγραφή σε ηλεκτρονικούς υπολογιστές όλων των κρίσιμων παραμέτρων της παραγωγικής διαδικασίας, για την επίτευξη εποπτείας σε πραγματικό χρόνο.

Ένα σύστημα SCADA περιλαμβάνει τα παρακάτω:

- Απεικόνιση σε μιμικά διαγράμματα όλων των διεργασιών παραγωγής.
- Ενδείξεις των τιμών των μετρούμενων μεγεθών.
- Διαρκή συλλογή και αποθήκευση δεδομένων σε Η/Υ .
- Γνωστοποίηση σφαλμάτων .
- Δυνατότητα παρακολούθησης του συστήματος μέσω κατάλληλα διαμορφωμένων σελίδων του διαδικτύου.



Εικόνα 5: Σύστημα SCADA

Η φιλοσοφία σχεδιασμού του συστήματος αποσκοπεί στα παρακάτω:

- Άμεση και ολοκληρωμένη επιτήρηση του συστήματος.
- Ενημέρωση του χειριστή για τυχόν υπερβάσεις των ορίων που ο ίδιος έχει θέσει.
- Αναδρομή σε όλο το ιστορικό των μετρήσεων.
- Καταγραφή των τιμών.
- Ευκολία και ευχρηστία στον χειρισμό.

2.8 Επίλογος

Στο κεφάλαιο αυτό έγινε η συνοπτική καταγραφή των χαρακτηριστικών αυτών που σχετίζονται με τους σταθμούς μέτρησης και κρίθηκε απαραίτητο να παρουσιαστούν στον αναγνώστη για την απόκτηση σφαιρικών γνώσεων σχετικά με το αντικείμενο της ατμοσφαιρικής μελέτης.

Με το κεφάλαιο αυτό ολοκληρώνεται το πρώτο μέρος της πτυχιακής και ακολουθεί το δεύτερο το οποίο εξετάζει τις τεχνολογίες που χρησιμοποιήθηκαν για την ανάπτυξη της διαδικτυακής εφαρμογής.

ΜΕΡΟΣ ΔΕΥΤΕΡΟ

ΚΕΦΑΛΑΙΟ 3^ο

3. Τεχνολογίες διαδικτύου

3.1 Εισαγωγή

Στο κεφάλαιο αυτό του δεύτερου μέρους της πτυχιακής εργασίας αναλύονται τα τεχνολογίες που χρησιμοποιήθηκαν κατά την ανάπτυξη της εφαρμογής, στις οποίες συγκαταλλέγονται οι PHP, HTML, JQuery, CSS και SQL. Ο αναγνώστης που δεν είναι εξοικειωμένος με το αντικείμενο καλείται να μελετήσει αυτό το κεφάλαιο ώστε να αποκτήσει τις απαραίτητες γνώσεις σχετικά με την εφαρμογή αφού η τεχνολογική πρόοδος των script γλωσσών προγραμματισμού έχει εξελιχθεί με μεγάλο ρυθμό τα τελευταία χρόνια. Ειδικότερα, αν και οι αρχές της βασίζονται στην τεχνολογία εμφάνισης κειμένου, με την πάροδο των χρόνων έχει έρθει σε σημείο να θεωρείται κάτι αναπόσπαστο κομμάτι για σχεδόν κάθε διαδικτυακή εφαρμογή όπως είναι και η παρούσα.

3.2 HTML

Η συντριπτική πλειοψηφία των ιστοσελίδων που είναι διαθέσιμες στον Παγκόσμιο Ιστό είναι γραμμένες στη γλώσσα προγραμματισμού HTML (Hypertext Markup Language). Η HTML είναι μια markup γλώσσα για την περιγραφή υπερκειμένων και πρωτοεμφανίστηκε στις αρχές του 1990. Η γλώσσα αυτή χρησιμοποιείται στο WWW (World Wide Web) και αποτελεί υποσύνολο της γλώσσας SGML (Standard Generalized Markup Language) που δημιουργήθηκε από την IBM για την επίλυση του προβλήματος της μη τυποποιημένης εμφάνισης κειμένων σε διάφορα υπολογιστικά συστήματα. Η γλώσσα αυτή έχει τυποποιηθεί από το World Wide Web Consortium (W3C) που είναι ο παγκόσμιος οργανισμός τυποποίησης τεχνολογιών του Web. Η SGML είναι μια μεταγλώσσα (metalanguage), μία γλώσσα δηλαδή που χρησιμοποιείται για την επίσημη περιγραφή markup γλωσσών. Με τη χρήση αυτής παρέχεται ένα σύστημα τυποποίησης κειμένων (document formatting system) το οποίο οι σχεδιαστές των Web σελίδων

χρησιμοποιούν για να περιγράψουν τη δομή και οργάνωση των σελίδων που κατασκευάζουν.

3.2.1 Χαρακτηριστικά της HTML

Η HTML επινοήθηκε το 1989 από τον Tim Berners Lee και στην αρχή χρησιμοποιούνταν μόνο για την δημιουργία στατικών ιστοσελίδων όπως επίσης και για την διάταξη των εγγράφων. Από την αρχή της χρήσης της μέχρι τώρα έχουν γίνει σχετικά λίγες αλλαγές στη γλώσσα ενώ παραμένει ιεραρχικά δομημένη και αποτελούμενη από ετικέτες οι οποίες ορίζουν το κάθε στοιχείο του εγγράφου. Οι ετικέτες αυτές (tags) περικλείουν τα στοιχεία υποδεικνύοντας την λειτουργία και τον σκοπό τους. Είναι γραμμένες απευθείας σε μορφή απλού κειμένου στο html έγγραφο όπου μπορούν να διερμηνευτούν από το λογισμικό του υπολογιστή ενώ δεν παρουσιάζονται στον browser κατά τη διάρκεια του rendering ενώ αποτελούν ξεχωριστό στοιχείο από το περιεχόμενο που περικλείουν.

Μέσα στην ιστοσελίδα περιλαμβάνονται ή και φορτώνονται script γλωσσών όπως JavaScript και Cascading Style Sheets (CSS) για τον καθορισμό, την εμφάνιση και τη διάταξη του κειμένου και του υπόλοιπου περιεχομένου.

3.2.2 Εκδόσεις της HTML

Το 1990 ο Tim Berners Lee από το Cern, το εργαστήριο φυσικής της Γενεύης, δημιούργησε ένα νέο πρωτόκολλο με το οποίο δίνονταν η δυνατότητα μεταφοράς κάθε είδους αρχείου και αντικειμένου μέσω του Internet. Το πρωτόκολλο αυτό ονομάστηκε HTTP (HyperText Transfer Protocol) και σηματοδότησε την αρχή του WWW με τη μορφή που έχει σήμερα. Οι σελίδες που ήταν η βάση του WWW ήταν γραμμένες στην πρώτη έκδοση της γλώσσα HTML.

Το 1994 αναπτύσσεται το πρότυπο HTML 2.0 από το διεθνή οργανισμό Internet Engineering Task Force. Η επόμενη έκδοση η 3.0 δεν έγινε αποδεκτή από τις Microsoft και Netscape οπότε γρήγορα αντικαταστάθηκε από την έκδοση 3.2 κάτι το οποίο έγινε το 1996. Η τελευταία περιελάμβανε πολλές από τις σημάνσεις (tags) που είχαν εισάγει οι δύο εταιρείες. Η έκδοση 4.0 παρουσιάστηκε τον Ιούνιο του 1997.

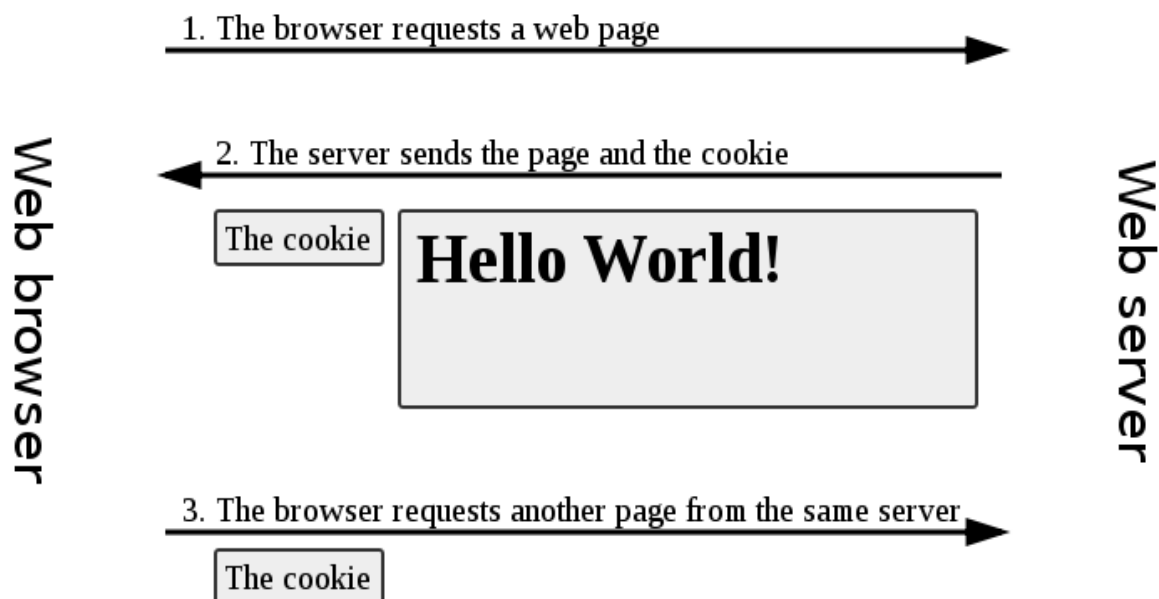
Η XHTML (Extensible HyperText MarkupLanguage) είναι ο διάδοχος της HTML, μία αυστηρότερη μορφή κώδικα, που δημιουργήθηκε με σκοπό την κατασκευή ιστοσελίδων που θα είναι προσβάσιμες και από άλλες συσκευές εκτός από τους browsers (π.χ. κινητά τηλέφωνα). Πάνω σε αυτή την τεχνολογία στηρίζονται οι browsers που έχουν σχεδιαστεί ειδικά για τα άτομα με ειδικές ανάγκες. Το αποτέλεσμα είναι να επιτρέπεται ένα μεγαλύτερο κοινό πρόσβασης στις ιστοσελίδες που σχεδιάστηκαν με XHTML.

Στην ουσία η XHTML είναι μία μορφή XML, κάτι που σημαίνει ότι χρησιμοποιεί συντακτικά σωστό κώδικα. Από πολλές απόψεις αποτελεί τη διασταύρωση της HTML και της XML. Επίσης, έχει τη δυνατότητα εκτενής χρήσης των εγγράφων CSS (Cascading Style Sheets) που είναι υπεύθυνα για την εμφάνιση και παρουσίαση του περιεχομένου.

Σημειώνεται ότι η XHTML προτάθηκε από το W3C στις 26 Ιανουαρίου του 2000.

3.3 HTTP

Κάθε ενέργεια των web εφαρμογών, έχει ως αποτέλεσμα μία κλήση στον εξυπηρετητή (server). Κάτι τέτοιο απαιτεί την ανανέωση της σελίδας για την ανάκτηση των απαραίτητων πληροφοριών από τον server και την αποστολή τους στο φυλλομετρητή (browser). Η απαραίτητη αυτή λειτουργία δεν έχει να κάνει με την αρχιτεκτονική του κάθε browser ξεχωριστά αλλά με το πρωτόκολλο HTTP request/response το οποίο υλοποιείται σε όλους τους browser ανεξαιρέτως (π.χ. Firefox, Internet Explorer, Safari, Chrome κτλ). Ο τρόπος με τον οποίο λειτουργεί το παραπάνω πρωτόκολλο φαίνεται στην παρακάτω εικόνα.



Εικόνα 5: Η λειτουργία του πρωτοκόλλου HTTP [10]

Όπως φαίνεται και στο σχήμα κάθε ερώτηση προς το server έχει και την αντίστοιχη απόκριση από αυτόν.

Το HTTP είναι το στάνταρ πρωτόκολλο μεταφοράς του Παγκόσμιου Ιστού και ξεκίνησε από την έκδοση HTTP/0.9 κατά τη διάρκεια της πρώιμης ανάπτυξης του διαδικτύου και ακολούθησε η έκδοση HTTP/1.0 το 1995. Η πιο πρόσφατη έκδοσή του, HTTP/1.1, παρέχει περισσότερη λειτουργικότητα και υποστηρίζει πολλαπλές συναλλαγές μεταξύ πελάτη και διακομιστή κατά τη διάρκεια της ίδιας αίτησης.

Κάθε ανταλλαγή πληροφορίας συνίσταται από μία αίτηση ASCII, ακολουθούμενη από μία απόκριση RFC 822 τύπου MIME, η οποία αποτελεί το de facto πρότυπο μορφής μηνύματος για το ηλεκτρονικό ταχυδρομείο. Παρά το γεγονός ότι είναι σύνηθες η χρησιμοποίηση του Πρωτοκόλλου Ελέγχου Μεταφοράς TCP (Transmission Control Protocol) για τη σύνδεση μεταφοράς, δεν απαιτείται επίσημα από το πρότυπο.

3.3.1 Apache Server

Ο Apache HTTP γνωστός και απλά σαν Apache είναι ένας εξυπηρετητής του παγκόσμιου ιστού. Όταν γίνεται μία επίσκεψη σε μία ιστοσελίδα ο browser επικοινωνεί με έναν διακομιστή HTTP. Ο Apache είναι ένας από τους δημοφιλέστερους, εν μέρει γιατί λειτουργεί σε διάφορες πλατφόρμες σαν τα Windows, το Linux, το Unix, και το Mac OS X. Τώρα, διατηρείται από μια κοινότητα ανοιχτού κώδικα με την επιτήρηση του Ιδρύματος Λογισμικού Apache (Apache Software Foundation).

Η πρώτη έκδοση του Apache, γνωστή ως NCSA HTTPd, δημιουργήθηκε από τον Robert McCool και κυκλοφόρησε το 1993 και θεωρείται ότι έπαιξε σημαντικό ρόλο στην αρχική επέκταση του ιστού. Ήταν η πρώτη βιώσιμη εναλλακτική επιλογή που παρουσιάστηκε απέναντι στον εξυπηρετητή http της εταιρείας Netscape και από τότε έχει εξελιχθεί στο σημείο να ανταγωνίζεται άλλους εξυπηρετητές βασισμένους στο Unix σε λειτουργικότητα και απόδοση.

3.4 CSS

Τα Cascading Style Sheets (CSS) είναι μία γλώσσα που χρησιμοποιείται από την HTML και την XHTML για τον ορισμό της παρουσίασης των ιστοσελίδων. Εφαρμόζεται σε κάθε στοιχείο της σελίδας ξεχωριστά και τροποποιεί τα χρώματα, τις γραμματοσειρές, την τοποθέτηση εικόνων και άλλων στοιχείων κ.λ.π.

Το CSS σχεδιάστηκε με σκοπό το διαχωρισμό της πληροφορίας και της εμφάνισης ώστε να μπορούν περισσότεροι χρήστες να έχουν πρόσβαση στη πληροφορία των ιστοσελίδων. Μειώνει κατά πολύ τον κώδικα των ιστοσελίδων, αφού εξαλείφει τον επαναλαμβανόμενο κώδικα, με αποτέλεσμα να βοηθάει στη δημιουργία πιο γρήγορων ιστοσελίδων, οι οποίες μάλιστα είναι και προσβάσιμες σε μεγαλύτερο βαθμό από τις μηχανές αναζήτησης.

Σημειώνεται ότι αν και πρωτοεμφανίστηκε το 1996, η ολοκληρωμένη υποστήριξη από τους πιο δημοφιλείς browsers δεν έγινε μέχρι το 2000.

3.4.1 Αρχές του CSS

Το CSS επιτρέπει την προσθήκη στυλ στις ιστοσελίδες, ενσωματώνοντας τη μορφή στο περιεχόμενο. Κάθε HTML στοιχείο προσδίδει κάποιο περιεχόμενο και οι ιδιότητες οι οποίες δηλώνονται με κάθε ένα από αυτά παρέχουν περισσότερη πληροφορία σε αυτά τα στοιχεία.

Το δομικό στοιχείο του CSS είναι ο κανόνας (rule) ο οποίος είναι ένα σύνολο από εντολές τις οποίες ο browser ακολουθεί ώστε να αλλάξει την προεπιλεγμένη παρουσίαση των HTML στοιχείων ανάλογα με τις τιμές που έχουν οριστεί. Ένας CSS κανόνας αποτελείται από έναν επιλογέα (selector) και μία δήλωση (declaration).

Ο επιλογέας είναι το κομμάτι αυτό του κανόνα το οποίο επεξηγεί στον browser ποιο θα είναι το στοιχείο το οποίο πρόκειται να μορφοποιηθεί. Η σκοπιά του μπορεί να είναι μεγάλου εύρους, επηρεάζοντας τη μορφή κάθε στοιχείου, ή ακόμα και πολύ μικρή και συγκεκριμένη, επηρεάζοντας για παράδειγμα μόνο ένα στοιχείο (ή και κανένα).

Η δήλωση είναι το μέρος εκείνο του κανόνα το οποίο βρίσκεται μέσα στις αγκύλες και αποτελείται από δύο ξεχωριστά στοιχεία τα οποία ακολουθούν πάντα συγκεκριμένη σειρά και είναι η ιδιότητα (property) και η τιμή (value).

Η ιδιότητα είναι το τμήμα εκείνο του στοιχείου το οποίο πρόκειται να μορφοποιηθεί, όπως είναι το χρώμα, το πλάτος, το ύψος, η γραμματοσειρά ή ακόμα και η θέση όπου πρόκειται να εμφανιστεί το στοιχείο.

Η τιμή της ιδιότητας (property value) προσδίδει το συγκεκριμένο στυλ το οποίο πρόκειται να εφαρμοστεί στο επιλεγμένο στοιχείο. Οι τιμές οι οποίες είναι αποδεκτές εξαρτώνται από την κάθε ιδιότητα, και μερικές ιδιότητες δέχονται πολλαπλές τιμές οι οποίες χωρίζονται με κόμμα.

3.5 JavaScript

Η JavaScript είναι μια γλώσσα script η οποία χρησιμοποιείται στην ανάπτυξη ιστοσελίδων και αναπτύχθηκε το 1995 από τις εταιρείες Netscape και τη Sun Microsystems. Είναι γλώσσα που εκτελείται στη μεριά του χρήστη (client-side) που σημαίνει ότι στέλνεται κατευθείαν στον browser του πελάτη και δε μεταγλωττίζεται

στον server όπως συμβαίνει με την PHP. Αυτό όμως σημαίνει ότι για να εκτελεστεί ο κώδικας της JavaScript θα πρέπει να υποστηρίζεται από τον browser του χρήστη.

Η JavaScript προσφέρει ένα πλήθος εντολών όπως if, while κ.ά. ενώ πλέον έχει γίνει τόσο ισχυρή ώστε υποστηρίζει και αντικείμενα. Επίσης έχει την δυνατότητα να κάνει υπολογισμούς, να αντιδράσει σε διάφορα σύμβαντα όπως onclick, onmouseover κ.ά., να ελέγξει και να αλλάξει τα πεδία μιας φόρμας, να στείλει μηνύματα στον χρήστη και δημιουργήσει διάφορα εφέ κίνησης.

Παρόλο που το όνομα της JavaScript μοιάζει με αυτό της Java, οι δύο γλώσσες δεν έχουν καμία σχέση μεταξύ τους εκτός του ότι και οι δυο είναι γραμμένες σε C και στο ότι η JavaScript δανείζεται να ονόματα κάποιων εντολών της Java.

3.5.1 JQuery

Η JQuery γράφτηκε από τον John Resig το 2006, τώρα βρίσκεται στην τέταρτη έκδοση και είναι διαθέσιμη από τη διεύθυνση <http://jquery.com/>. Αποτελεί μία πολύ ισχυρή βιβλιοθήκη της JavaScript η οποία πλέον αποτελεί αναπόσπαστο κομμάτι σχεδόν σε κάθε διαδικτυακή εφαρμογή καθώς προσφέρει μία γενικού σκοπού αφαίρεση (abstraction) κατά τη συγγραφή των προγραμμάτων, είναι πλήρως επεκτάσιμη και για αυτό καλύπτουν ευρύ φάσμα περιπτώσεων. Τα βασικά χαρακτηριστικά της βιβλιοθήκης είναι τα εξής:

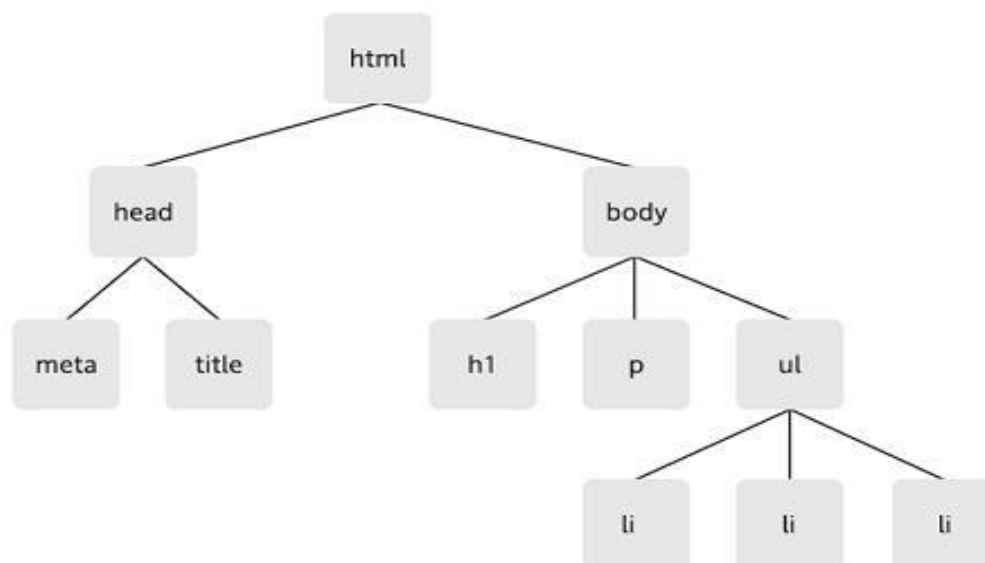
- Πρόσβαση στοιχείων στο έγγραφο (document): Με τη JQuery η δυνατότητα για εύρεση συγκεκριμένων στοιχείων στο Document Object Model (DOM) είναι εξαιρετικά εύκολη σε αντίθεση με την απλή χρήση της JavaScript όπου απαιτούνται πολλές γραμμές κώδικα για τέτοιου είδους χειρισμούς.
- Τροποποίηση της εμφάνισης της ιστοσελίδας: Ο συνδυασμός της JQuery με το CSS είναι ένα από τους κύριους λόγους χρήσης της αφού παρέχει λειτουργίες που υποστηρίζονται από όλους τους κύριους browser.
- Τροποποίηση του περιεχομένου του εγγράφου: Με τη χρήση της JQuery δίνεται η δυνατότητα τροποποίησης του περιεχομένου οποιουδήποτε στοιχείου του εγγράφου με το συμβάν πλήθους γεγονότων.

- Δημιουργία animation: Πιθανόν το πιο εντυπωσιακό χαρακτηριστικό όσον αφορά τον τελικό χρήστη είναι η χρήση της JQuery για τη δημιουργία animation αφού πολύ εύκολα μπορούν να δημιουργηθούν εφέ κίνησης, fade in και fadeout ώστε να δίνεται η αίσθηση οπτικής αλληλεπίδρασης.
- Ανάκτηση πληροφορίας από server χωρίς επαναφόρτωση της ιστοσελίδας: Η χρήση του AJAX είναι πλέον εξαιρετικά διαδεδομένη και ο συνδυασμός του με τη JQuery επιτρέπει στον προγραμματιστή να μην ανησυχεί σχετικά με την πολυπλοκότητα του AJAX σε διαφορετικούς browser καθώς παρέχει ενσωματωμένες λειτουργίες για εύκολο χειρισμό των γεγονότων.

3.6 Document Object Model

Για την ανάγνωση και επεξεργασία των εγγράφων μέσα από μια εφαρμογή έχει αναπτυχθεί το Document Object Model (DOM). Το DOM έχει συσταθεί και υποστηρίζεται σαν πρότυπο από το W3C (World Wide Web Consortium) και πλέον διαθέτει ευρεία υποστήριξη από όλους τους κύριους browser και εκτός από ανάγνωση προσφέρει δυνατότητες για τροποποίηση υπάρχοντων ή για δημιουργία νέων εγγράφων. Επίσης, είναι αντικειμενοστραφές και μπορεί να ενσωματωθεί σε ένα μεγάλο αριθμό από προγραμματιστικά περιβάλλοντα τόσο με γλώσσες προγραμματισμού όσο και με γλώσσες script

Σύμφωνα με τον ορισμό του W3C, το DOM είναι ένα ουδέτερο σε λειτουργικό και γλώσσα προγραμματισμού περιβάλλον το οποίο επιτρέπει σε προγράμματα και script να έχουν πρόσβαση και να ανανεώνουν το περιεχόμενο, τη δομή και το στυλ των εγγράφων. Το έγγραφο μπορεί να επεξεργαστεί περαιτέρω και τα αποτελέσματα αυτής της επεξεργασίας μπορούν να ενσωματωθούν στην σελίδα η οποία βρίσκεται υπό παρουσίαση.



Εικόνα 6: Αναπαράσταση ενός δέντρου DOM

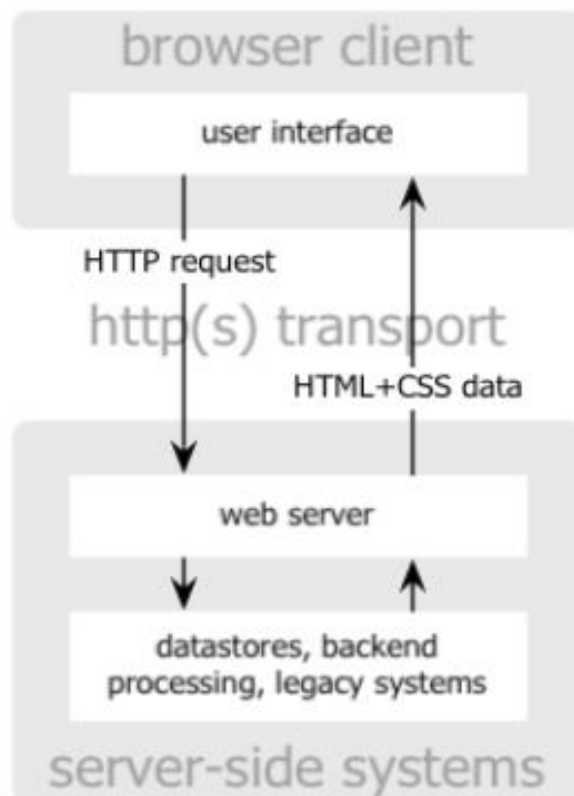
Η παραπάνω Εικόνα αναπαριστά το δέντρο ενός απλού εγγράφου HTML με μία ετικέτα h1, μία ετικέτα p και μία ετικέτα ul με ενσωματωμένες σε αυτήν τρεις ετικέτες li.

3.7 PHP

Η PHP (Hypertext Preprocessor) αποτελεί μία ευρέως χρησιμοποιούμενη, open source και γενικού σκοπού script γλώσσα η οποία είναι ειδικά σχεδιασμένη για την ανάπτυξη web εφαρμογών με δυνατότητα ενσωμάτωσης μέσα σε κώδικα HTML και εκτέλεσής της κάθε φορά που ο χρήστης επισκέπτεται την εκάστοτε σελίδα. Ο PHP κώδικας μεταφράζεται στον Web διακομιστή και κατ' αυτόν τον τρόπο δημιουργείται κώδικας HTML ή κάποια άλλη έξοδος.

Αυτό που διαχωρίζει την PHP από τα client-side JavaScripts είναι ότι ο κώδικας εκτελείται στον server (εξυπηρετητή). Αν υπήρχε ένα script PHP, ο browser θα έπαιρνε τα αποτελέσματα της εκτέλεσης αυτού του script, χωρίς να μπορεί να καταλάβει με κανένα τρόπο ποιος είναι κώδικας που εκτελείται στο παρασκήνιο.

Ένα τυπικό παράδειγμα μοντέλου χρήσης της PHP, HTML, CSS και HTTP request φαίνεται στην παρακάτω Εικόνα.



Εικόνα 6: Τυπικό μοντέλο web εφαρμογής

Αν και η ανάπτυξη της PHP εστιάζεται σε server-side scripting (scripting στην πλευρά του διακομιστή), μπορούνε να γίνουν πολύ περισσότερα με αυτήν.

3.7.1 Βασικά χαρακτηριστικά της PHP

Η PHP ενσωματώνει την ισχύ και τη δυναμικότητα σχετικά παλαιότερων γλωσσών όπως η Perl αλλά καταργώντας τις αδυναμίες τους. Παρακάτω αναφέρονται μερικά από τα βασικά χαρακτηριστικά της:

- Ο συντακτικός αναλυτής της, καθώς και ο πηγαίος κώδικάς της διανέμεται ελεύθερα στο διαδίκτυο δίνοντας την δυνατότητα σε όποιον θέλει να κατασκευάζει και να διανέμει εφαρμογές για εμπορική και μη χρήση.
- Μπορεί να μεταφραστεί και να τρέξει στα περισσότερα λειτουργικά συστήματα που κυκλοφορούν στην αγορά (Microsoft Windows, Linux, BSD, Solaris, Macintosh OS X, και UNIX servers).

- Συνεργάζεται χωρίς προβλήματα με τους πιο δημοφιλείς Web Servers που κυκλοφορούν όπως τον Apache και τον Microsoft IIS.
- Διαθέτει ενσωματωμένες εντολές υποστήριξης για ένα μεγάλο αριθμό βάσεων δεδομένων όπως MySQL, Sybase, Oracle, Ingres. Προσφέρει ένα σύνολο από Database API's τις ενοποιημένες ODBC συναρτήσεις (unified ODBC functions), που εξασφαλίζουν την προσπέλαση σε μια υποκείμενη βάση δεδομένων, χρησιμοποιώντας τις εγγενείς μεθόδους της εκάστοτε βάσης για να μεγιστοποιήσουν την απόδοση (IBM DB2).
- Είναι πιο απλό να συντάξει κάποιος κώδικα PHP από ότι σε οποιαδήποτε άλλη γλώσσα σεναρίου.
- Μπορεί να χρησιμοποιηθεί στη δημιουργία εικόνων, ανάγνωση / εγγραφή σε αρχεία και για αποστολή email. Για να προσφέρει αυτές τις υπηρεσίες, η PHP επικοινωνεί με αρκετά πρωτόκολλα όπως: HTTP (Ιστοσελίδες), POP3 (email), SNMP και LDAP.
- Υποστηρίζει τόσο τον διαδικαστικό προγραμματισμό όσο και τον αντικειμενοστραφή.

3.8 SQL Server

Ο SQL Server αποτελεί ένα από τα πολλά RDBMS που υπάρχουν στο χώρο των βάσεων δεδομένων και έχει αναπτυχθεί από τη Microsoft. Χρησιμοποιεί την T-SQL και ANSI SQL και αποτελεί ένα πολύ γρήγορο και δυνατό σύστημα διαχείρισης βάσεων δεδομένων με πολυνηματική δυνατότητα σύνδεσης πολλαπλών χρηστών με παράλληλη εξασφάλιση ότι μόνο πιστοποιημένοι χρήστες μπορούν να έχουν πρόσβαση.

Οι κυριότερες υπηρεσίες που προσφέρει είναι οι παρακάτω:

- Ο Service Broker ο οποίος τρέχει ως μέρος της μηχανής βάσεων δεδομένων, παρέχει μια αξιόπιστη πλατφόρμα μηνυμάτων και αναμονής μηνυμάτων για τις εφαρμογές κεντρικών SQL Server
- Οι υπηρεσίες συγχρονισμού (Replication) χρησιμοποιούνται από τον SQL Server το συγχρονισμό βάσεων δεδομένων, είτε ολοκληρωτικά είτε ως υποσύνολο των αντικειμένων. Υπάρχουν τουλάχιστον 4 διαφορετικοί

μέθοδοι (log shipping, mirroring, replication, back-ups, clustering) για το συγχρονισμό. Αυτό βοηθάει επίσης και σε περιπτώσεις ολοκληρωτικής καταστροφής της μητρικής βάσης δεδομένων, δίνοντας την δυνατότητα ακόμη και για αυτόματη λειτουργία της δευτερεύουσας συγχρονισμένης βάσης δεδομένων.

- Οι υπηρεσίες ανάλυσης (Analysis Services) SQL Server προσθέτουν τις ικανότητες OLAP και περίπλοκης ανάκτησης δεδομένων για τις βάσεις δεδομένων SQL, με περίπλοκη ανάλυση χωρίς την ανάγκη για τη συγγραφή περίπλοκων ερωτημάτων SQL.
- Ο SQL Server έχει υπηρεσίες Reporting για στοιχεία που συγκεντρώνονται από τις βάσεις δεδομένων SQL Server. Με αυτόν τον τρόπο είναι δυνατό να παραχθεί μια έκθεση Ιστού (Web Report) σχεδόν αυτόματα χωρίς προγραμματισμό HTML, CSS ή JavaScript, μέσα από το Visual Studio.
- Οι υπηρεσίες ανακοίνωσης (Notification Services) δίνουν την δυνατότητα στον χρήστη να παράγει μηνύματα ηλεκτρονικού ταχυδρομείου με πολύ λίγο κόπο.
- Το Integration Services του SQL χρησιμοποιούνται για να ενσωματώσουν στοιχεία από διαφορετικές πηγές δεδομένων. Χρησιμοποιεί τις ικανότητες ETL. Οι Integration Services περιλαμβάνουν εργαλεία GUI για να χτίσουν διάφορες λειτουργίες ροών δεδομένων, όπως η εξαγωγή στοιχείων από διάφορες πηγές, η συζήτηση στοιχείων, ο μετασχηματισμός των στοιχείων συμπεριλαμβανομένης της συνάθροισης, του διπλασιασμού και η συγχώνευσης.
- Ο SQL Server μπορεί επίσης να αναζητήσει μέσα σε κείμενα (Full Text Search Service). Ο δείκτης αναζήτησης μπορεί να δημιουργηθεί σε οποιαδήποτε στήλη και έτσι η αναζήτηση πραγματοποιείται άμεσα. Για παράδειγμα, κείμενα Word και έγγραφα pdf αποθηκευμένα στον SQL Server μπορούν αναζητηθούν πολύ γρήγορα.

3.9 MySQL

Στη συγκεκριμένη εργασία, χρησιμοποιείται εκτός από τον SQL Server το RDBMS MySQL. Ο λόγος είναι ότι η χρήση της MySQL είναι εξαιρετικά συνηθισμένη στις διαδικτυακές εφαρμογές κι έτσι κρίθηκε σκόπιμο να χρησιμοποιηθεί για την αποθήκευση των περιεχομένων της εκάστοτε σελίδας.

Η RDBMS το οποίο εκτελείται σε κάποιον εξυπηρετητή παρέχοντας multiuser πρόσβαση σε πολλαπλές βάσεις δεδομένων. Είναι συμβατή με ANSI-SQL και ακολουθεί το σχεσιακό μοντέλο. Αποτελεί έναν πολυνηματικό διακομιστή πολλαπλών χρηστών με δυνατότητα αυθεντικοποίησής τους. Η ιστορία της ξεκινάει από το 1979 αλλά είναι διαθέσιμη από το 1996 και είναι ανοικτού κώδικα.

Τα κυριότερα χαρακτηριστικά της είναι τα εξής:

- Ανοικτού κώδικα και δωρεάν (για επαγγελματικές εφαρμογές απαιτείται αγορά).
- Είναι ιδιαίτερα αποτελεσματική για διαδικτυακές εφαρμογές.
- Λόγω της συμβατότητάς της με την ANSI-SQL είναι συμβατή και μεταφέρσιμη σε διάφορες πλατφόρμες και άλλα RDBMS.
- Είναι εξαιρετικά γρήγορη και βελτιστοποιημένη όσον αφορά την ταχύτητα ανάκτησης δεδομένων.

Η MySQL έχει πολλές διαφορετικές μηχανές αποθήκευσης διαθέσιμες για αποθήκευση και ανάκτηση στοιχείων αλλά οι δύο πιο συνηθισμένες είναι η MyISAM και η InnoDB. Η MyISAM υποστηρίζεται ότι είναι λίγο γρηγορότερης ταχύτητας από την InnoDB όσον αφορά την εκτέλεση των ερωτημάτων αλλά η InnoDB έχει τη δυνατότητα υποστήριξης επιπλέον χαρακτηριστικών τα οποία δεν είναι διαθέσιμα από τη MyISAM. Το κυριότερο από αυτά είναι η δημιουργία ξένων κλειδιών με αναφορές σε άλλους πίνακες κάτι το οποίο δίνει τη δυνατότητα για την εισαγωγή περιορισμών κατά τα διάφορα γεγονότα που προκύπτουν στη διάρκεια της εκτέλεσης της εφαρμογής.

3.10 Αρχιτεκτονική Βάσης Δεδομένων με PHP – MySQL

Ένας server έχει ως αποστολή την εξυπηρέτηση αιτήσεων από τους client. Έτσι, όταν γίνεται μία αίτηση προς έναν server αυτός αφού ολοκληρώσει τις απαραίτητες ενέργειες αποστέλλει τα αποτελέσματα στον client.

Παρόμοια είναι και η αρχιτεκτονική που υποστηρίζει μία διαδικτυακή εφαρμογή που περιλαμβάνει με βάση δεδομένων, αν και είναι λίγο πιο περίπλοκη.

Έτσι, μία τυπική web συναλλαγή η οποία συμπεριλαμβάνει δεδομένα που βρίσκονται σε μία βάση δεδομένων, αποτελείται από τις παρακάτω φάσεις:

1. Ο επισκέπτης κάνει μία αίτηση στην ιστοσελίδα μέσα από μία τυπική URL διεύθυνση.
2. Ο server αναγνωρίζει ότι το αιτούμενο αρχείο είναι ένα PHP script και ξεκινάει την εκτέλεση του κώδικα..
3. Γίνεται η σύνδεση με τη βάση δεδομένων και η αίτηση για το περιεχόμενο που ανήκει στη συγκεκριμένη ιστοσελίδα.
4. Ο SQL Server απαντά στέλνοντας το αιτούμενο περιεχόμενο στο αντίστοιχο PHP script.
5. Το PHP script αποθηκεύει το περιεχόμενο μέσα σε μία ή περισσότερες μεταβλητές και τις επιστρέφει ως HTML σελίδα στον server.
6. Ο server αποθηκεύει ένα αντίγραφο της HTML σελίδας που δημιουργήθηκε.
7. Ο server στέλνει την HTML σελίδα στον browser και τα αποτελέσματα της επεξεργασίας γίνονται ορατά στον χρήστη.

Η διαδικασία είναι βασικά η ίδια, ανεξάρτητα από το ποια μηχανή script ή ποιος server βάσης δεδομένων χρησιμοποιείται. Συνήθως το πρόγραμμα του server, η PHP μηχανή και ο server της βάσης δεδομένων βρίσκονται στον ίδιο υπολογιστή. Ωστόσο, είναι πολύ συνηθισμένο ο server της βάσης δεδομένων να βρίσκεται σε διαφορετικό υπολογιστή. Αυτό μπορεί να γίνει για λόγους ασφάλειας, για μεγαλύτερη χωρητικότητα ή για κατανομή του φόρτου εργασίας.

3.11 Επίλογος

Στο κεφάλαιο αυτό παρουσιάστηκαν οι τεχνολογίες που χρησιμοποιήθηκαν στην εφαρμογή. Το κυριότερο μέρος είναι η PHP, και ο SQL Server που φυσικά δεν είναι δυνατό να γίνουν κατανοητά ως έννοιες μόνο με τη θεωρητική τους παρουσίαση. Για το λόγο αυτό, σε επόμενο κεφάλαιο αναλύεται τόσο η δομή της βάσης δεδομένων που χρησιμοποιείται από τους σταθμούς μέτρησης όσο και τμήματα κώδικα σε PHP που γράφτηκαν για την υλοποίηση της εφαρμογής και τη λήψη δεδομένων από τους σταθμούς μέσα από την ιστοσελίδα.

ΜΕΡΟΣ ΤΡΙΤΟ

ΚΕΦΑΛΑΙΟ 4^ο

4. Βάση δεδομένων της εφαρμογής

4.1 Εισαγωγή

Στο πρώτο κεφάλαιο του τρίτου και τελευταίου μέρους της εργασίας γίνεται η τεκμηρίωση της βάσης δεδομένων που χρησιμοποιείται από την εφαρμογή. Αρχικά, παρουσιάζεται η απαραίτητη θεωρία που σχετίζεται με τις βάσεις δεδομένων ενώ στη συνέχεια αναλύεται αυτή καθ' αυτή η βάση όπως χρησιμοποιήθηκε για την υλοποίηση της εφαρμογής.

4.2 Κανονικοποίηση

Η διαδικασία της σωστής σχεδίασης της δομής των πινάκων της βάσης, έτσι ώστε να ικανοποιεί τις προδιαγραφές που έχουν τεθεί και ταυτόχρονα να είναι ευέλικτη και αποδοτική στη χρήση της ονομάζεται λογικός ή εννοιολογικός σχεδιασμός (conceptual or logical design). Η θεωρία της κανονικοποίησης αναπτύχθηκε ώστε να γίνεται η ανίχνευση προβληματικών καταστάσεων όπως είναι για παράδειγμα η ασυνέπεια δεδομένων και οι διπλοεγγραφές. Η κανονικοποίηση προτάθηκε το 1970 από τον Codd και εισάγει τις έννοιες των κανονικών μορφών οι οποίες στη γενική περίπτωση είναι τρεις (1^η Κανονική Μορφή, 2^η Κανονική Μορφή, 3^η Κανονική Μορφή) ενώ τα αρχικά δεδομένα θεωρούνται ότι βρίσκονται σε Μη Κανονική Μορφή. Παρακάτω παρουσιάζεται η θεωρία για το πώς τα δεδομένα μεταφέρονται από τη Μη Κανονική Μορφή στην 3^η Κανονική Μορφή εξαλείφοντας τις συναρτησιακές εξαρτήσεις που υπάρχουν ανάμεσα στα πεδία των πινάκων.

4.2.1 Μη Κανονική Μορφή

Σε αυτό το στάδιο δεν υπάρχει ακόμα πίνακα με την σχεσιακή έννοια αλλά τα πεδία των πινάκων τοποθετούνται όλα το ένα δίπλα στο άλλο με μόνο επιπλέον στοιχείο πληροφορίας την ομαδοποίηση πεδίων που αντιστοιχούν σε χαρακτηριστικά που παίρνουν πολλαπλές τιμές. Η ομαδοποίηση των τελευταίων

αναπαριστάται με ζεύγη εσωτερικών παρενθέσεων στην περιγραφή της δομής του αντίστοιχου πίνακα.

4.3.2 1^η Κανονική Μορφή (1NF)

Ένας πίνακας βρίσκεται σε κανονική μορφή όταν έχουν απομακρυνθεί όλες οι επαναλαμβανόμενες ομάδες πεδίων κατά τέτοιο τρόπο ώστε η τομή μιας γραμμής και μιας στήλης του πίνακα, να αντιστοιχεί πάντα σε μια απλή τιμή. Έτσι επιτυγχάνεται η αποτροπή εμφάνισης πολλαπλών και σύνθετων τιμών στα πεδία ενός πίνακα δηλαδή όταν κάθε εγγραφή του πίνακα είναι ατομική κάτι που σημαίνει ότι δε μπορεί να γίνει διάσπασή της σε μικρότερες μονάδες πληροφορίας.

4.3.3 2^η Κανονική Μορφή (2NF)

Στη συνέχεια της διαδικασίας κανονικοποίησης, ένας πίνακας βρίσκεται σε δεύτερη κανονική μορφή όταν είναι σε πρώτη κανονική μορφή και ακολούθως απομακρυνθούν όλες οι μερικές συναρτησιακές εξαρτήσεις που υπάρχουν ανάμεσα στα πεδία που τον απαρτίζουν. Αυτό σημαίνει ότι όλα τα πεδία που δεν ανήκουν στο κύριο κλειδί του πίνακα έχουν πλήρη συναρτησιακή εξάρτηση μόνο από τα πεδία του κυρίως κλειδιού. Εδώ σημειώνεται ότι όταν το κύριο κλειδί ενός πίνακα είναι απλό τότε ένας πίνακας που είναι σε πρώτη κανονική μορφή είναι αυτόματα και σε δεύτερη κανονική μορφή.

4.3.4 3^η Κανονική Μορφή (3NF)

Εν τέλει, ένας πίνακας βρίσκεται σε τρίτη κανονική μορφή όταν είναι σε δεύτερη και εν συνεχεία απομακρυνθούν όλες οι μεταβατικές συναρτησιακές εξαρτήσεις που υπάρχουν μεταξύ των πεδίων του. Σημειώνεται ότι μια συναρτησιακή εξάρτηση $A \rightarrow B$ ονομάζεται μεταβατική, όταν υπάρχει ένα πεδίο C το οποίο δεν ανήκει στο κύριο κλειδί του πίνακα, τέτοιο ώστε $C = C(A)$ και $B = B(C)$.

4.4 Σχεσιακό Μοντέλο

Το σχεσιακό μοντέλο αποτελεί ένα από τα πιο απλά και ευέλικτα μοντέλα αυτού του είδους. Στην ουσία είναι ένα γράφημα στο οποίο απεικονίζονται οι οντότητες της βάσης και οι μεταξύ τους συσχετίσεις. Οι οντότητες αυτές αποτελούν τους πίνακες οι οποίοι περιέχουν ένα πλήθος γραμμών και στηλών και κάθε μία γραμμή ονομάζεται εγγραφή όπου περιέχει ένα σύνολο απλών πεδίων τα οποία συσχετίζονται μεταξύ τους. Κάθε πεδίο του πίνακα πρέπει να περιέχει τιμές οι οποίες ανήκουν σε ένα συγκεκριμένο σύνολο τιμών διαφορετικά δε γίνονται αποδεκτές ενώ οι τιμές που καταχωρούνται σε μία εγγραφή πρέπει να είναι ατομικές ώστε να μην παραβιάζονται οι κανόνες των τριών κανονικών μορφών.

4.5 Οντότητες και Συσχετίσεις

Παρακάτω αναλύονται οι οντότητες και συσχετίσεις οι οποίες πρόκειται να χρησιμοποιηθούν κατά τη διάρκεια της υλοποίησης για την αποθήκευση της απαιτούμενης πληροφορίας. Πρώτα γίνεται η γενική περιγραφή του πίνακα και ακολουθεί η περιγραφή των πεδίων από τα οποία αυτοί απαρτίζονται.

4.5.1 Βάση δεδομένων στους σταθμούς μέτρησης

Στο σημείο αυτό περιγράφεται η βάση δεδομένων που χρησιμοποιείται στους σταθμούς μέτρησης για την καταγραφή των τιμών.

Air_hum_data0: Στον πίνακα Air_hum αποθηκεύονται οι τιμές της υγρασίας του ατμοσφαιρικού αέρα.

Air_temp_data0: Στον πίνακα Air_temp αποθηκεύονται οι τιμές της θερμοκρασίας του ατμοσφαιρικού αέρα.

Conductivity_data0: Στον πίνακα αυτόν αποθηκεύονται οι τιμές της ηλεκτρικής αγωγιμότητας.

Depth_data0: Στον πίνακα αυτόν αποθηκεύονται οι τιμές του βάθους μέτρησης του οργάνου.

DO_data0: Στον πίνακα αυτόν αποθηκεύονται οι τιμές του διαλυμένου οξυγόνου στο νερό.

pH_data0: Στον πίνακα αυτόν αποθηκεύονται οι τιμές του pH του νερού.

Salt_data0: Στον πίνακα αυτόν αποθηκεύονται οι τιμές της αλατότητας του νερού.

Seawater_data0: Στον πίνακα αυτόν αποθηκεύονται οι τιμές του νερού της θαλάσσης.

TDS_data0: Στον πίνακα αυτόν αποθηκεύονται οι τιμές του διαλυμένων στερεών σωματιδίων.

Turbidity_data0: Στον πίνακα αυτόν αποθηκεύονται οι τιμές του θολότητας.

Water_temp_data0: Στον πίνακα αυτόν αποθηκεύονται οι τιμές θερμοκρασίας του θαλασσινού νερού.

Wind_direction_data0: Στον πίνακα αυτόν αποθηκεύονται οι τιμές της κατεύθυνσης του ανέμου.

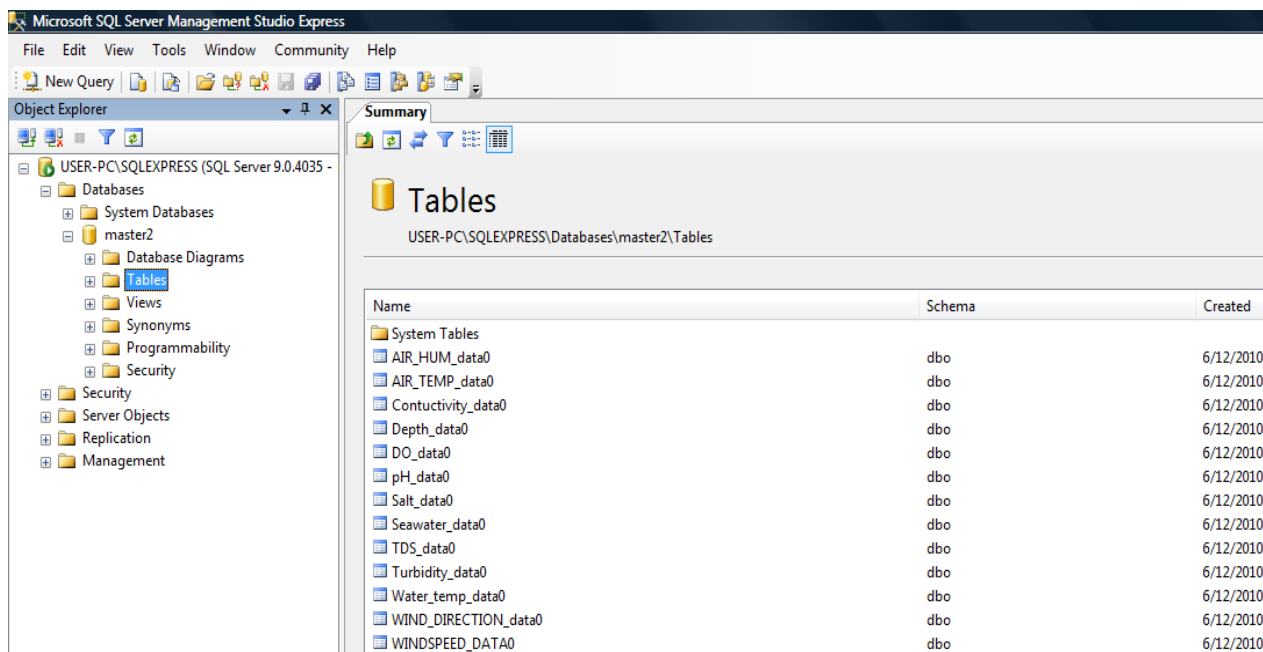
Windspeed_DATA0: Στον πίνακα αυτόν αποθηκεύονται οι τιμές της ταχύτητας του ανέμου.

Λόγω του γεγονότος ότι κάθε πίνακας στη βάση του SQL Server αποτελείται από τα ίδια πεδία, παρατίθεται η γενική τεκμηρίωση που ισχύει για κάθε έναν από αυτούς.

Πίνακας	
Πεδίο	Περιγραφή
VarName	Η κωδική ονομασία των δεδομένων.
TimeString	Η ημερομηνία και ώρα λήψης της τιμής.
VarValue	Η τιμή για το συγκεκριμένο στοιχείο.
Validity	Η εγκυρότητα της ληφθέντας τιμής.
Time_ms	Ο ρυθμός ανανέωσης της σάρωσης στον

	προγραμματιζόμενο λογικό ελεγκτή.
--	-----------------------------------

Σημειώνεται ότι Το πεδίο TimeString περιέχει την ημερομηνία και ώρα όπως αυτή εμφανίζεται στον Η/Υ του σταθμού μέτρησης που είναι εγκατεστημένο το λογισμικό παρακολούθησης τιμών.



Εικόνα 7: Η βάση δεδομένων αποθήκευσης των τιμών των στοιχείων προς καταγραφή όπως αυτή φαίνεται από το Microsoft SQL Server Management Studio Express

Στην Εικόνα 7 φαίνονται όλοι οι πίνακες που έχουν περιγραφεί παραπάνω και οι οποίοι αποτελούν τη βάση δεδομένων καταγραφής των περιβαλλοντικών στοιχείων.

4.5.1 Βάση δεδομένων για τη δημιουργία της ιστοσελίδας

Παρακάτω περιγράφεται η δομή της βάσης δεδομένων σύμφωνα με την οποία δημιουργείται το layout της ιστοσελίδας. Τα mods αποτελούν στην ουσία της σελίδες όλης της εφαρμογής.

Mods: Στον πίνακα αυτόν αποθηκεύονται τα λογικά τμήματα της εφαρμογής, κάθε ένα από τα οποία αντιστοιχεί σε μία ξεχωριστή ιστοσελίδα.

Πίνακας: Mods	
Πεδίο	Περιγραφή
id	Ο αύξων αριθμός του κάθε module της ιστοσελίδας.
mod_name	Το όνομα του module.
description_gr	Η περιγραφή του module στα ελληνικά.
description_en	Η περιγραφή του module στα αγγλικά.
title_gr	Ο τίτλος της ιστοσελίδας στα ελληνικά που αναγράφεται στο πεδίο της ετικέτας <title>.
title_en	Ο τίτλος της ιστοσελίδας στα αγγλικά που αναγράφεται στο πεδίο της ετικέτας <title>.
content_gr	Το περιεχόμενο του συγκεκριμένου module στα ελληνικά.
content_en	Το περιεχόμενο του συγκεκριμένου module στα αγγλικά.
rank	Η σειρά των modules όπως αυτά θα εμφανίζονται στην ιστοσελίδα.

Refs: Στον πίνακα αυτόν αποθηκεύονται τα λογικά υποτμήματα της εφαρμογής, κάθε ένα από τα οποία αντιστοιχεί σε ένα ξεχωριστό τμήμα κάθε ιστοσελίδας.

Πίνακας: Refs

Πεδίο	Περιγραφή
id	Ο αύξων αριθμός του κάθε reference της ιστοσελίδας.
mods_id	Το id του module στο οποίο αντιστοιχίζεται το reference.
ref	Το όνομα του reference.
description_gr	Η περιγραφή του reference στα ελληνικά.
description_en	Η περιγραφή του reference στα αγγλικά.
content_gr	Το περιεχόμενο του συγκεκριμένου reference στα ελληνικά.
content_en	Το περιεχόμενο του συγκεκριμένου reference στα αγγλικά.
rank	Η σειρά των reference όπως αυτά θα εμφανίζονται στην ιστοσελίδα.

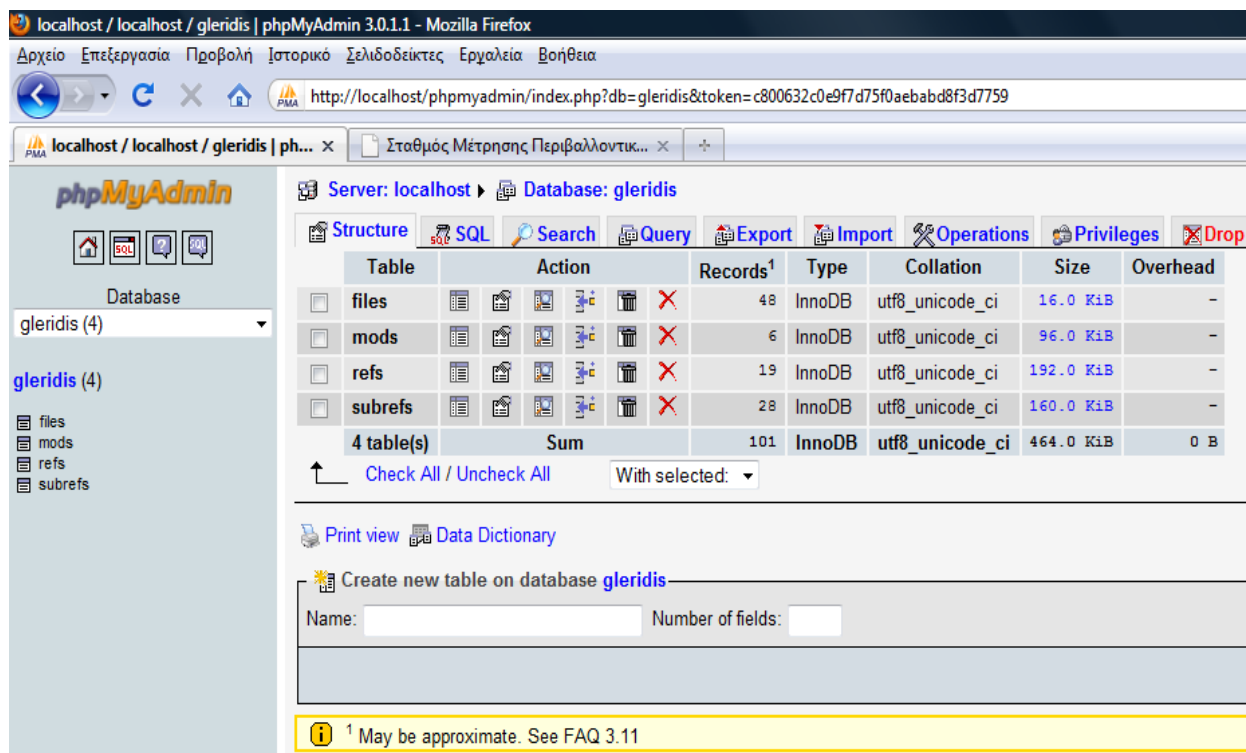
Subrefs: Στον πίνακα subrefs αποθηκεύονται τα υποτομήματα τα οποία αντιστοιχούν σε references.

Πίνακας: Subrefs	
Πεδίο	Περιγραφή
id	Ο αύξων αριθμός του κάθε reference της ιστοσελίδας.
refs_id	Το id του reference στο οποίο αντιστοιχίζεται το subreference.
subref	Το όνομα του subreference.

description_gr	Η περιγραφή του subreference στα ελληνικά.
description_en	Η περιγραφή του subreference στα αγγλικά.
content_gr	Το περιεχόμενο του συγκεκριμένου subreference στα ελληνικά.
content_en	Το περιεχόμενο του συγκεκριμένου subreference στα αγγλικά.
rank	Η σειρά των subreference όπως αυτά θα εμφανίζονται στην ιστοσελίδα.

Files: Στον πίνακα files αποθηκεύονται τα αρχεία και τα διαγράμματα τα οποία δημιουργούνται κατά τα ερωτήματα των χρηστών.

Πίνακας: Subrefs	
Πεδίο	Περιγραφή
id	Ο αύξων αριθμός του κάθε αρχείου της ιστοσελίδας.
name	Το όνομα του αρχείου.
time_created	Η στιγμή δημιουργίας του αρχείου.



Εικόνα 8: Η βάση δεδομένων αποθήκευσης των τμημάτων της ιστοσελίδας όπως αυτή φαίνεται από το phpMyAdmin

Στην παραπάνω Εικόνα, φαίνονται οι πίνακες της βάσης, η μηχανή engine (InnoDB) της δομής τους καθώς και η κωδικοποίηση της βάσης και των πινάκων (utf_unicode_ci).

4.6 Επίλογος

Με τη συμπλήρωση του κεφαλαίου αυτού, έχουν πλέον περιγραφεί επακριβώς και οι δύο τύποι βάσεων δεδομένων που χρησιμοποιούνται στην εφαρμογή. Στο επόμενο και τελευταίο κεφάλαιο θα επεξηγηθούν τα σημαντικότερα τμήματα του κώδικα.

5. Προγραμματισμός σε PHP

5.1 Εισαγωγή

Για την ολοκλήρωση της εργασίας, το μόνο που έχει απομείνει είναι η παρουσίαση των πιο σημαντικών τμημάτων κώδικα του προγραμματιστικού μέρους της εφαρμογής. Λόγω του μεγέθους του κώδικα δεν είναι δυνατόν να γίνει πλήρης παρουσίαση αυτού αλλά μόνο επιλεκτική επεξήγηση. Για μία πιο πλήρη αναφορά ο αναγνώστης καλείται να ανατρέξει στην ηλεκτρονική τεκμηρίωση.

5.2 Cookies και Sessions

Σε προηγούμενο κεφάλαιο έχει ήδη αναφερθεί ότι το πρωτόκολλο που βασίζεται η συγκεκριμένη εφαρμογή, δηλαδή το HTTP, είναι ένα ακαταστατικό stateless πρωτόκολλο κάτι που δεν του επιτρέπει να έχει γνώση του προηγούμενου ιστορικού του χρήστη. Έτσι, κατά τη διάρκεια της περιήγησής ενός χρήστη, το πρωτόκολλο αυτό θεωρεί ότι είναι η πρώτη φορά που εισέρχεται στο σύστημα και συνδέεται με αυτό. Αυτό φυσικά είναι πρόβλημα σε εφαρμογές στις οποίες απαιτείται η ταυτοποίηση του χρήστη, όπως για παράδειγμα σε ένα forum. Αν η γλώσσες προγραμματισμού δεν είχαν στη διάθεσή τους διάφορους τρόπους και εργαλεία για την αυθεντικοποίηση των χρηστών τότε θα έπρεπε κάθε σε κάθε σελίδα που απαιτούνταν αυθεντικοποίηση να υπάρχει και φόρμα εισόδου για την πιστοποίηση ότι ο χρήστης έχει το δικαίωμα να εισέλθει στο σύστημα. Αυτό το πρόβλημα λύνεται με τη χρήση των Cookies αλλά πολύ καλύτερα με τη χρήση των συνόδων Session.

5.2.1 Cookies

Τα Cookies αποθηκεύονται σε ένα αρχείο στον υπολογιστή του χρήστη. Με αυτόν τον τρόπο κάθε φορά που ο χρήστης κάνει μία αίτηση για τη λήψη πληροφορίας από κάποια ιστοσελίδα αποστέλλεται και το cookie που αντιστοιχεί σε αυτήν και γίνεται η ταυτοποίησή του.

Όσον αφορά την PHP, τα cookies μπορούν να δημιουργηθούν με τη χρήση της συνάρτησης `setcookie()`, όπως και φαίνεται στην παρακάτω γραμμή κώδικα:

```
setcookie("connected", "gleridis", time()+3600);
```

Εδώ δημιουργείται ένα cookie με το όνομα "connected" και τιμή "gleridis" για το οποίο ορίζεται να λήξει σε 3.600 δευτερόλεπτα (μία ώρα) μετά τη δημιουργία του.

Για την ανάκτηση και τη διαγραφή ενός cookie, αρκεί μόνο το παρακάτω τμήμα κώδικα:

```
setcookie("connected ", "", time()-3600);
```

Η γραμμή αυτή κώδικα θέτει την ημερομηνία λήξης του cookie στο παρελθόν (μία ώρα πριν) οπότε αυτό διαγράφεται από το σύστημα ενώ επίσης διαγράφει και την τιμή του.

Το πρόβλημα που έγκειται όσον αφορά τη χρήση των cookies για την πιστοποίηση των χρηστών είναι ότι τίθεται σημαντικό θέμα ασφάλειας. Αυτό γίνεται κατανοητό αν σκεφτεί κανείς ότι οι τιμές των cookies αποθηκεύονται στον υπολογιστή του client οπότε και οι χρήστες έχουν πρόσβαση από όλους όσους έχουν πρόσβαση και στο σύστημα. Πέρα όμως από το θέμα ασφάλειας, οι χρήστες μπορούν να επιλέξουν την απενεργοποίηση των cookies από το σύστημά τους οπότε κάθε εφαρμογή που βασίζεται σε αυτά για τη λειτουργία της είναι αυτομάτως προβληματική. Όσον αφορά το θέμα της ασφάλειας, δίνει τη λύση η χρήση των Sessions.

5.2.2 Sessions

Τα Sessions μοιάζουν αρκετά με τα cookies αλλά έχουν κάποιες αρκετές σημαντικές διαφορές με κυριότερο το σημείο ότι ενώ τα cookies αποθηκεύονται στους υπολογιστές των επισκεπτών μίας ιστοσελίδας, τα session αντίθετα αποθηκεύονται στον server.

Τα Sessions λειτουργούν δημιουργώντας ενός τυχαίου cookie ID για κάθε χρήστη και αποθηκεύοντάς το αυτόματα στον υπολογιστή του. Το cookie αυτό ονομάζεται PHPSESSID και στην ουσία υπενθυμίζει στον server ποιο είναι το Session που πρέπει να ενεργοποιήσει για το συγκεκριμένο χρήστη από όπου γίνεται η κλήση. Το cookie αυτό κλείνει αυτόματα με το κλείσιμο του browser ή αν υπάρξει κάποια άλλη επιλογή αποσύνδεσης μέσα από την εφαρμογή, όπως για παράδειγμα το logout.

Για τη χρήση ενός Session, δημιουργείται πρώτα ένα συγκεκριμένο αρχείο και το αντίστοιχο cookie αποθηκεύεται στον υπολογιστή του χρήστη. Αυτό γίνεται χρησιμοποιώντας τη συνάρτηση `session_start()`.

```
session_start();  
$_SESSION['login'] = 'online';
```

Το παραπάνω τμήμα κώδικα δημιουργεί μία μεταβλητή τύπου Session και ορίζει να έχει την τιμή “online”.

Με αυτόν τον τρόπο, υπάρχει η δυνατότητα πλοήγησης στις διάφορες σελίδες της εφαρμογής, αφού έτσι το σύστημα είναι σε θέση να γνωρίζει ότι ο συγκεκριμένος χρήστης έχει προηγουμένως συνδεθεί, κάτι που σημαίνει ότι του επιτρέπεται η πρόσβαση στις σελίδες αυτές.

Καλό είναι να σημειωθεί ότι η συνάρτηση `session_start()` πρέπει να χρησιμοποιείται στην αρχή του base αρχείου πριν γίνει η εκτύπωση οποιουδήποτε HTML κώδικα ενώ δεν επιτρέπεται να υπάρχει ούτε ένα κενό πριν τη συνάρτηση αυτή. Σε διαφορετική περίπτωση που έχει σταλεί οποιαδήποτε πληροφορία στο browser του χρήστη θα προκύψει σφάλμα και η εκτέλεση του προγράμματος θα τερματιστεί. Για το λόγο αυτό, η συνάρτηση `session_start()` είναι η πρώτη γραμμή του κώδικα.

Τα βασικά βήματα για την χρήση τους τα οποία είναι τα εξής:

- Εκκίνηση της συνόδου λειτουργίας.
- Εγγραφή των μεταβλητών της συνόδου λειτουργίας στον server και ανάθεση cookies.

- Χρήση των μεταβλητών της συνόδου λειτουργίας.
- Ακύρωση και διαγραφή της εγγραφής των μεταβλητών και καταστροφή της συνόδου.

Επίσης, πρέπει να σημειωθεί ότι δεν είναι αναγκαίο όλες οι παραπάνω ενέργειες να υλοποιηθούν στο ίδιο script. Παρόλα αυτά εξακολουθεί να είναι απαραίτητη η ενεργοποίηση των cookies στην πλευρά του χρήστη.

```
function create_session_key($id) {  
  
    $session_key =  
    sha1($id."my_secret_string".time().get_ip());  
  
    return $session_key;  
  
}
```

Στην εφαρμογή που αναπτύχθηκε, η συνάρτηση που αναλαμβάνει τη δημιουργία ενός κλειδιού για την ταυτοποίηση του κάθε χρήστη ώστε να μην υπάρχει δυνατότητα για αποθήκευση περισσότερων του ενός αρχείου από έναν χρήστη στον server κατά την εισαγωγή ερωτημάτων.

Όπως φαίνεται, το κλειδί αυτό περιέχει πολλές παραμέτρους οι οποίες είναι διαφορετικές για διαφορετικούς χρήστες και δεν είναι δυνατό να αναπαραχθούν περισσότερες από μία φορές αφού περιλαμβάνουν εκτός από την ip του χρήστη και μία χρονοσφραγίδα. Τέλος, το κλειδί περνάει από συνάρτηση κατακερματισμού για τη μονόδρομη κωδικοποίησή της. Η διαδικασία αυτή περιγράφεται ως salting.

5.3 Υπερκαθολικές μεταβλητές `$_GET` και `$_POST`

Οποιοσδήποτε έχει επισκεφθεί κάποια ιστοσελίδα που έχει δημιουργηθεί με php, ίσως να έχει παρατηρήσει ότι συχνά στη γραμμή διευθύνσεων URL, υπάρχει ο χαρακτήρας του αγγλικού ερωτηματικού “?” το οποίο ακολουθείται από άλλες παραμέτρους. Ένα τέτοιο παράδειγμα είναι το παρακάτω:

```
myscript.php?id=5&value=basket&time=25432&name=glelidis
```

Οι παράμετροι αυτές λαμβάνονται από τον web server και αφού ενσωματωθούν ως μεταβλητές στο αρχείο PHP, υφίστανται επεξεργασία ώστε η σελίδα να εμφανίσει συγκεκριμένες πληροφορίες. Πρέπει να σημειωθεί ότι το σύμβολο με το οποίο συνενώνονται οι τιμές στη γραμμή διευθύνσεων είναι το "&" για όλες τις παραμέτρους πέραν της πρώτης για την οποία χρησιμοποιείται το σύμβολο του αγγλικού ερωτηματικού "?". Ακολουθεί ένα παράδειγμα κώδικα όπου τα GET data χρησιμοποιούνται για να εμφανίσουν διαφορετικό κείμενο:

```
<?php
$id = $_GET['id'];
if ($id == 1) {
    print "Όνομα";
}
elseif ($id == 2) {
    print "Επίθετο";
}
else {
    print "Όνομα και Επίθετο";
}
```

Αν γίνει η παραδοχή ότι το αρχείο με τον παραπάνω κώδικα ονομάζεται script.php τότε ο επισκέπτης που θα ανοίξει τη σελίδα με τη μορφή script.php?id=1 θα δει το κείμενο "Όνομα". Αν ανοίξει τη σελίδα σαν script.php?id=2 θα δει στην οθόνη του τη φράση "Επίθετο", ενώ για οποιοδήποτε άλλο id, θα δει τη φράση "Όνομα και Επίθετο".

Αν ο αριθμός του id πρόκειται να σταλεί μέσω μια φόρμας που χρησιμοποιεί τη μέθοδο POST, τότε αρκεί να αντικατασταθεί η γραμμή:

```
$id = $_GET['id'];
```

με την ακόλουθη:

```
$id = $_POST['id'];
```

Πρέπει να σημειωθεί ότι ο παλαιότερος τρόπος λήψης μεταβλητών μέσω POST και GET data ήταν με τη χρήση των long arrays \$HTTP_POST_VARS και \$HTTP_GET_VARS αντίστοιχα. Αυτά όμως καταργούνται και ήδη δεν είναι

συμβατά με την νέα έκδοση 6 της PHP. Αυτό ισχύει ακόμη και αν στις ρυθμίσεις της PHP 6 ενεργοποιηθούν τα long arrays (register_long_arrays = On).

5.4 Σύνδεση με τη βάση δεδομένων

Ένα από τα πιο σημαντικά σημεία της εφαρμογής αποτελεί ο κώδικας με τον οποίο γίνεται η σύνδεση με τη βάση δεδομένων. Ο παρακάτω κώδικας υλοποιεί τη σύνδεση με τη MySQL.

```
function db_connection() {  
  
    $connection = mysqli_connect(SERVER, USER,  
MY_PASSWORD, MY_DATABASE);  
  
    $connection->query("SET NAMES 'utf8'");  
  
    if (!connection) {  
  
        die("Database connection failed: " .  
mysqli_error());  
  
    }  
  
}
```

Η συνάρτηση αυτή παίρνει σαν ορίσματα το όνομα του server στον οποίο φιλοξενείται η βάση δεδομένων, το όνομα του χρήστη της βάσης, τον κωδικό πρόσβασης και το όνομα της βάσης. Έπειτα, δημιουργεί έναν πόρο σύνδεσης (μεταβλητή \$connection) καλώντας τη συνάρτηση mysqli_connect() που παρέχεται από την PHP. Αν ο πόρος δημιουργηθεί σωστά τότε επιστρέφεται και είναι έτοιμος για χρήση, διαφορετικά θα υπάρχει θα τερματιστεί η λειτουργία του προγράμματος αφού καλείται η συνάρτηση die(). Μάλιστα, για να γίνει πιο εύκολη η εύρεση του λάθους, χρησιμοποιείται η συνάρτηση mysqli_error() η οποία επιστρέφει μία συμβολοσειρά με πληροφορίες για το σφάλμα που προέκυψε. Πάντως, καλή πρακτική είναι η συνάρτηση αυτή να χρησιμοποιείται μόνο όσο το πρόγραμμα είναι στο στάδιο της υλοποίησης αφού με την mysqli_error() παρέχονται πληροφορίες για τη βάση οι οποίες είναι καλό να κρατούνται μυστικές.

Όταν γίνει η σύνδεση με το διακομιστή βάσης δεδομένων θα πρέπει να επιλεγεί η βάση δεδομένων με την οποία θα λειτουργεί η εφαρμογή. Η δυνατότητα αυτή

δίνεται από την συνάρτηση `mysqli_select_db()`. Η συγκεκριμένη μέθοδος απαιτεί το όνομα μιας βάσης δεδομένων και προαιρετικά έναν αναγνωριστή συνδέσμου. Η `mysqli_select_db()` επιστρέφει `true` αν υπάρχει η βάση δεδομένων και αν μπορεί να γίνει χειρισμός αυτής.

Για βάσεις δεδομένων Microsoft SQL, PostgreSQL ή Oracle, οι παραπάνω εντολές αντικαθίστανται αντίστοιχα με τις:

```
mssql_connect και mssql_select_db για Microsoft SQL
pg_connect και pg_select_db για PostgreSQL
oci_connect για Oracle
```

Εναλλακτικά, μπορούν να χρησιμοποιηθούν οι εντολές `mysqli_pconnect`, `pg_pconnect`, `mssql_pconnect` και `oci_pconnect` που δημιουργούν `persistent` σύνδεση στην `database`. Όμως αυτή η πρακτική δεν συνιστάται, ιδιαίτερα σε `web sites` με αυξημένη κίνηση. Αυτό γιατί με `pconnect` οι συνδέσεις στη βάση δεδομένων παραμένουν ανοικτές όταν τερματιστεί η εκτέλεση του κώδικα PHP ή ακόμη και αν ο επισκέπτης κλείσει τον `web browser` και τον υπολογιστή του. Το αποτέλεσμα είναι η βάση δεδομένων να φτάσει πολύ σύντομα στο όριο ανοικτών συνδέσεων, αφήνοντας "εκτός" άλλους χρήστες που πλοηγούνται στο `web site`.

5.5 SQL Injection και XSS Attacks

Κάθε εφαρμογή η οποία είναι προσπελάσιμη από το Internet και χρησιμοποιεί κάποια βάση δεδομένων είναι αυτομάτως ευάλωτη σε επιθέσεις της μορφής SQL Injection και XSS Attack. Αυτό γίνεται λόγω των κενών ασφαλείας που μπορεί να υπάρχουν αν δε ληφθούν τα κατάλληλα μέτρα για την αποτροπή τους. Οι επιθέσεις αυτές αποτελούν κίνδυνο όταν το σύστημα λαμβάνει και χρησιμοποιεί δεδομένα τα οποία δεν έχουν πρώτα επικυρωθεί και αυτά είτε προορίζονται για χρήση στη βάση δεδομένων είτε για εμφάνιση σε κάποια από τις ιστοσελίδες της εφαρμογής. Η πρώτη περίπτωση αφορά το SQL Injection ενώ η δεύτερη το XSS Attack.

5.5.1 SQL Injection

Η επίθεση αυτή περιγράφει την τεχνική εισαγωγής κώδικα ο οποίος βρίσκεται ενσωματωμένος σε κάποιο από τα πεδία φορμών ή στις διευθύνσεις URL οι τιμές των οποίων προορίζονται να ενσωματωθούν σε ερωτήματα που γίνονται προς τη βάση.

```
INSERT INTO mods (name_gr) VALUES ('mods'); DROP TABLE
mods;--')
```

Με το παραπάνω τμήμα κώδικα, είναι δυνατή η διαγραφή του πίνακα που βρίσκονται αποθηκευμένα όλα τα modules της εφαρμογής. Το ερωτηματικό χρησιμοποιείται για να δηλώσει το τέλος ενός ερωτήματος κι έτσι τερματίζοντας το ερώτημα δίνεται η δυνατότητα να προστεθεί ένα ακόμα ερώτημα το οποίο είναι και αυτό που διαγράφει τελικώς τον πίνακα mods. Οι δύο χαρακτήρες “ -- ” στο τέλος του ερωτήματος δηλώνουν ότι ό,τι ακολουθεί αποτελεί σχόλιο κι έτσι με αυτόν τον τρόπο οι υπόλοιποι χαρακτήρες αγνοούνται. Σημειώνεται ότι η SQL δίνει τη δυνατότητα εκτέλεσης πολλαπλών ερωτημάτων τα οποία διαχωρίζονται με ερωτηματικό.

Για να αποφευχθεί αυτός ο κίνδυνος, γίνεται χρήση της συνάρτησης `mysqli_real_escape_string()` η οποία είναι υλοποιημένη στο σύνολο των εντολών της PHP. Ο ρόλος της συνάρτησης αυτής είναι η προετοιμασία της μεταβλητής ώστε να εισαχθεί στη βάση με ασφαλής μορφή.

```
$name_gr = mysqli_real_escape_string($name_gr,
$name_gr);
```

Αυτό που κάνει πραγματικά αυτή η εντολή είναι η διαφυγή του χαρακτήρα “ ‘ ” πριν η μεταβλητή χρησιμοποιηθεί από τη βάση δεδομένων.

5.5.2 XSS Attack

Ο τρόπος με τον οποίο γίνεται η επίθεση XSS Attack έχει να κάνει με την παρεμβολή κώδικα JavaScript μέσα στην HTML. Για την προστασία των εφαρμογών από τέτοιες επιθέσεις, η PHP παρέχει έτοιμες συναρτήσεις με περισσότερο γνωστή από αυτές την `htmlentities()`. Η συνάρτηση αυτή μετατρέπει τους HTML χαρακτήρες στις κωδικοποιημένες οντότητες που αντιστοιχούν σε αυτούς.

```

```

Όταν ο παραπάνω κώδικας εκτυπωθεί στην οθόνη, θα εμφανιστεί ένα παράθυρο `alert` με το μήνυμα “ XSS attack!”. Λύση σε αυτό δίνει η χρήση της συνάρτησης `htmlentities()`.

```
echo htmlentities('');
```

Έτσι με αυτόν τον τρόπο θα γίνει διαφυγή του κώδικα JavaScript. Επίσης, κατά τη λήψη μεταβλητών από τη URL κάθε μεταβλητή που δεν παίρνει αλφαριθμητική τιμή, παύλα “ - ” ή κάτω παύλα “ _ ” μετατρέπεται σε “ % ” ακολουθούμενο από δύο ψηφία. Εκτός της `htmlentities()`, μπορεί να χρησιμοποιηθεί και η συνάρτηση `rawurlencode()` η οποία μετατρέπει κάθε μεταβλητή με τον ίδιο τρόπο που θα μετατρέπονταν από τον server.

5.5.3 Sanitize

Πέρα από τις ενσωματωμένες εντολές της PHP, στην εφαρμογή χρησιμοποιείται επιπρόσθετος κώδικας για τον έλεγχο κάθε μεταβλητής που πρόκειται να χρησιμοποιηθεί από τη βάση. Αυτό γίνεται με τις συναρτήσεις `sanitize`. Έτσι, πριν οποιαδήποτε μεταβλητή συνεχίσει για χρήση στη βάση δεδομένων, φιλτράρεται μέσα από αυτού του είδους συναρτήσεις οι οποίες χρησιμοποιούνται για όλα τα input πεδία που παρέχει η HTML.

```
function sanitize_post($varname, $type, $default_value = '') {
    if(empty($_POST[$varname])) {
        return $default_value;
    }
    $var = trim($_POST[$varname]);
    switch ($type) {
        case 'num':
            if(is_numeric($var) && ctype_digit($var)) {
                return $var;
            }
            break;
        case 'str':
            if(is_string($var) && !is_numeric($var) &&
!preg_match('/[?;|_\'\"@#$$%^&*()0-9]/', $var)) {
                return $var;
            }
            break;
        case 'alnum':
            if(is_string($var) && !preg_match('/[;|_\'\"%^]/',
$var)) {
                return $var;
            }
            break;
        case 'arr':
            if(is_array($var)) {
                return $var;
            }
            break;
        case 'bool':
            if(is_bool($var)) {
                return $var;
            }
            else if($var) {
                return true;
            }
            break;
    }
    return $default_value;
}
```

Αυτό που κάνει η παραπάνω συνάρτηση, είναι δεχόμενη τα τρία ορίσματα να ελέγχει σύμφωνα με τον τύπο που δέχεται από το δεύτερο όρισμα αν η τιμή της μεταβλητής `$_POST[$varname]` είναι σύμφωνη με το πεδίο τιμών που μπορεί να παίρνει. Αν κάτι τέτοιο δεν ισχύει, η μεταβλητή παίρνει την προεπιλεγμένη τιμή που έχει οριστεί στο τρίτο όρισμα.

5.6 Κώδικας εμφάνισης σελίδων

Η εφαρμογή χρησιμοποιεί δυναμικό τρόπο δημιουργίας κάθε σελίδας που φαίνεται στον τελικό χρήστη. Αυτό γίνεται επειδή όλη η πληροφορία είναι αποθηκευμένη στη βάση δεδομένων και σύμφωνα με τα ορίσματα που υπάρχουν στη γραμμή διευθύνσεων URL λαμβάνονται οι ανάλογες μεταβλητές και ως εκ τούτου και διαφορετικό περιεχόμενο.

```
function do_site_content($connection, $mods_id, $l) {
    $s = '';
    $content_l = '';
    $mod = strtolower(sanitize_get('mod', 'str', ''));
    $ref = strtolower(sanitize_get('ref', 'str', ''));
    $subref = strtolower(sanitize_get('subref', 'str', ''));
    $query1 = "SELECT content_{$l} FROM mods WHERE id = '{$mods_id}'";
    $result1 = q($connection, $query1);
    $row1 = mysqli_fetch_array($result1);
    $content_l = $row1['content_{$l}'];
    if (!empty($ref)) {
        $query2 = "SELECT id, content_{$l} FROM refs WHERE ref =
'$ref' AND mods_id = '{$mods_id}'";
        $result2 = q($connection, $query2);
        $num_rows2 = mysqli_num_rows($result2);
```

....συνέχεια....

```
        if ($num_rows2 < 1) {
            redirect_to(ROOT.'mod=.'.$mod);
        } else {
            $row2 = mysqli_fetch_array($result2);
            $content_l = $row2['content_{$l}'];
            $refs_id = $row2['id'];
            if (!empty($subref)) {
                $query3 = "SELECT id, content_{$l} FROM subrefs
WHERE subref = '{$subref}' AND refs_id = '{$refs_id}'";
                $result3 = q($connection, $query3);
                $num_rows3 = mysqli_num_rows($result3);
                if ($num_rows3 > 0) {
                    $row3 = mysqli_fetch_array($result3);
                    $content_l = $row3['content_{$l}'];
                } else {

                    redirect_to(ROOT.'mod=.'.$mod.'&ref=.'.$ref);
                }
            }
        }
    }
    $s .= '<div id="content_container">';
    $s .= do_top_corners();
    $s .= $content_l;
    $s .= do_bottom_corners();
    $s .= '</div>';
    return $s;
}
```

Η συγκεκριμένη συνάρτηση λαμβάνει ως παράμετρο το `mods_id` και μέσα στον κώδικά της καλεί την `sanitize` ώστε να πάρει τις τιμές του `ref` και του `subref`. Αρχικοποιεί τη μεταβλητή `content_l` με το περιεχόμενο του τρέχοντος `mod` κι έπειτα βρίσκει αν αυτό περιέχει κάποιο `ref` και αυτό με τη σειρά του κάποιο `subref` ώστε να πάρει διαδοχικά τις τιμές του καθένα από αυτά. Σε περίπτωση που κάποια από αυτές τις μεταβλητές είναι άδεια, το περιεχόμενο της `content_l` παραμένει το ίδιο με το προηγούμενο και η συνάρτηση επιστρέφει την τιμή της μεταβλητής.

5.7 Επιλογή γλώσσας

Η εφαρμογή υποστηρίζει την ύπαρξη πολλαπλών γλωσσών για το περιεχόμενο. Ο κώδικας με τον οποίο γίνεται η επιλογή και η εναλλαγή γλωσσών είναι ο

```
function lang($l) {
    if (!empty($l)) {
        if ($l == 'gr') {
            $_SESSION['l'] = 'gr';
            require_once("languages/lang_gr.php");
        } elseif ($l == 'en') {
            $_SESSION['l'] = 'en';
            require_once("languages/lang_en.php");
        }
    } else {
        if (!empty($_SESSION['l'])) {
            if ($_SESSION['l'] == 'gr') {
                require_once("languages/lang_gr.php");
            } elseif ($_SESSION['l'] == 'en') {
                require_once("languages/lang_en.php");
            }
        } else {
            $_SESSION['l'] = 'gr';
            require_once("languages/lang_gr.php");
        }
    }
}
```

παρακάτω:

Η συνάρτηση αυτή αρχικά ελέγχει αν ο χρήστης έχει πατήσει το σύνδεσμο εναλλαγής γλώσσας. Αν αυτό ισχύει τότε αν έχει πατήσει τον ελληνικό σύνδεσμο φορτώνεται το κατάλληλο αρχείο και η μεταβλητή `$_SESSION['l']` παίρνει την αντίστοιχη τιμή. Το ίδιο γίνεται και με το σύνδεσμο της αγγλικής γλώσσας. Σε

περίπτωση που δεν έχει γίνει κλικ σε κάποιον από τους συνδέσμους, τότε ελέγχεται η μεταβλητή `$_SESSION['l']` για να βρεθεί η τιμή της ώστε να φορτωθεί το αντίστοιχο αρχείο. Σε περίπτωση όμως που είναι η πρώτη φορά εκτέλεσης της συγκεκριμένης συνάρτησης σημαίνει ότι τίποτα από τα παραπάνω δεν ισχύει αφού όλες οι μεταβλητές είναι άδειες. Για το λόγο αυτό οι μεταβλητές τίθενται σε προεπιλεγμένες τιμές οι οποίες στη συγκεκριμένη περίπτωση φορτώνουν την ελληνική γλώσσα.

5.8 Εύρεση τελευταίας μέτρησης

Για την εύρεση της τελευταίας μέτρησης από τους δύο περιβαλλοντικούς σταθμούς χρησιμοποιείται ο παρακάτω κώδικας:

```
function get_last_data($table_name, $period) {
    $data = '';
    $query1 = "SELECT TimeString AS max_time_string,
Time_ms FROM $table_name WHERE Time_ms = (
                SELECT
DISTINCT(MAX(Time_ms)) FROM $table_name WHERE TimeString
LIKE '$period' AND Validity = 1);";
    $result1 = sql_q($sql_connection, $query1);
    $row1 = mssql_fetch_array($result1);
    $max_time_string = $row1['max_time_string'];
    $query2 = "SELECT DISTINCT(VarValue) AS VarValue
FROM $table_name WHERE TimeString LIKE
'$max_time_string';";
    $result2 = sql_q($sql_connection, $query2);
    $row2 = mssql_fetch_array($result2);
    $data = $row2['VarValue'];
    $data = number_format($data, 3, '.', '');
    return $data;
}
```

Πρώτα επιλέγεται η τιμή με το ανώτερο `Time_ms` για τη συγκεκριμένη χρονική περίοδο. Έπειτα, βρίσκεται η τιμή της μέτρησης που αντιστοιχεί σε αυτό το `Time_ms`. Αυτό είναι υλοποιήσιμο λόγω του γεγονότος ότι το `Time_ms` έχει την ιδιότητα του εναλλακτικού κλειδιού αφού όλες οι τιμές του είναι διακριτές.

5.9 Δημιουργία γραφήματος

Για τη δημιουργία customized γραφήματος με σύγχρονο τρόπο γίνεται χρήση της βιβλιοθήκης GD της PHP και της έτοιμης βιβλιοθήκης jrggraph.

```
function create_graph($connection, $sql_connection, $table_name,
$array1, $graph_filename, $graph_period) {
    $s .= '';
    if (empty($array1)) {
        return $s;
    }
    $graph = new Graph(515, 370);
    $graph->SetScale("textlin");
    $theme_class=new UniversalTheme;
    $graph->SetTheme($theme_class);
    $graph->img->SetAntiAliasing(false);
    $graph->title->Set($table_name);
    $graph->SetBox(false);
    $graph->img->SetAntiAliasing();
    $graph->yaxis->HideZeroLabel();
    $graph->yaxis->HideLine(false);
    $graph->yaxis->HideTicks(false, false);
    $graph->xgrid->Show();
    $graph->xgrid->SetLineStyle("solid");
    $graph->xgrid->SetColor('#E3E3E3');
    $p1 = new LinePlot($array1);
    $graph->Add($p1);
    $p1->SetColor("#6495ED");
    $p1->SetLegend($table_name.' over Time ('. $graph_period.'');
    $graph->legend->SetFrameWeight(1);
    $gdImgHandler = $graph->Stroke(_IMG_HANDLER);
    $graph->img->Stream($graph_filename);
    return $s;
}
```

Ο πίνακας array1 περιέχει όλα τα δεδομένα τα οποία θα γίνουν γράφημα. Αφού δημιουργηθεί ο καμβάς του γραφήματος, δημιουργείται ένα αντικείμενο LinePlot όπου και ενσωματώνεται στο γράφημα. Έπειτα, αποθηκεύεται σε ένα αρχείο τύπου png και τέλος εμφανίζεται στην οθόνη.

5.10 Δημιουργία αρχείου δεδομένων

Για την εμφάνιση του αρχείου δεδομένων με τις περιβαλλοντολογικές μετρήσεις χρησιμοποιείται τα παρακάτω τμήματα κώδικα τα οποία κάνουν χρήση και της MySQL και του SQL Server:

```
function create_file($connection, $sql_connection, $custom,
$stable_name, $periodFrom = '', $periodSince = '', $graph_period) {
    $s = '';
    $date = date('d_m_Y');
    $client = $_SESSION['session_key'];
    $t = '*.txt';
    $files =
glob("stats/". $stable_name . "_" . $date . "_" . $client . "*.txt");
    foreach($files as $file) {
        if (file_exists($file)) {
            unlink($file);
        }
    }
}
```

Πρώτα βρίσκεται η ταυτότητα του χρήστη από το `$_SESSION['session_key']` και οποιοδήποτε αρχείο έχει δημιουργήσει ο συγκεκριμένος χρήστης διαγράφεται.

```
$filename = 'stats/'. $stable_name . '_' . $date . '_' . $client . '.txt';
$graph_filename =
'stats_images/'. $stable_name . '_' . $date . '_' . $client . '.png';
$db_filename = $stable_name . '_' . $date . '_' . $client;
$query2 = "SELECT id, name FROM files WHERE
TIMESTAMPDIFF(MINUTE, time_created, NOW()) > 10;";
$result2 = q($connection, $query2);
while ($row2 = mysqli_fetch_array($result2)) {
    $id = $row2['id'];
    $name = $row2['name'];
    $query3 = "DELETE FROM files WHERE id = '$id'";
    $result3 = q($connection, $query3);
    if (file_exists('stats/'. $name . '.txt')) {
        unlink('stats/'. $name . '.txt');
    }
    if (file_exists('stats_images/'. $name . '.png')) {
        unlink('stats_images/'. $name . '.png');
    }
}
```

Έπειτα, βρίσκονται όλα τα αρχεία και γραφήματα τα οποία έχουν δημιουργηθεί οποιαδήποτε χρονική στιγμή μεγαλύτερη των 10 λεπτών και κατόπιν διαγράφονται. Αυτό γίνεται λόγω του περιορισμένου χώρου που υπάρχει στον server καθώς αν δεν υπήρχε μέριμνα για κάτι τέτοιο ο δίσκος θα γέμιζε από τα αρχεία που θα δημιουργούνταν κατά τη διάρκεια της χρήσης της εφαρμογής.


```

$query4 = "INSERT INTO files (name) VALUES ('$db_filename');";
$result4 = q($connection, $query4);
$link = fopen($filename, "a");
//fwrite($link, $data."\r\n");
if ($custom == 'no') {
    $query1 = "SELECT VarName, VarValue, TimeString,
Validity, Time_ms FROM $table_name WHERE TimeString LIKE
'$periodFrom' AND Validity = 1";
} else {
    $query1 = "SELECT VarName, VarValue, TimeString,
Validity, Time_ms FROM $table_name WHERE TimeString > '$periodFrom'
AND TimeString < '$periodSince' AND Validity = 1";
}
$data =
"VarName."\t"."TimeString."\t"."VarValue."\t"."Validity."\t"."Time
_ms";
fwrite($link, $data."\r\n");
$values = array();
$result1 = sql_q($sql_connection, $query1);
while ($row1 = mssql_fetch_array($result1)) {
    $VarName = $row1['VarName'];
    $TimeString = $row1['TimeString'];
    $VarValue = $row1['VarValue'];
    $Validity = $row1['Validity'];
    $Time_ms = $row1['Time_ms'];
    $values[] = $VarValue;
    $data =
$VarName."\t".$TimeString."\t".$VarValue."\t".$Validity."\t".$Time_ms
;
    fwrite($link, $data."\r\n");
}
fwrite($link, "\r\n");
fwrite($link, "\r\n");
$d = file($filename);
fclose($link);
create_graph($connection, $sql_connection, $table_name,
$values, $graph_filename, $graph_period);
unset($values);
$s .= '<div><a class="newWindow"
href="http://localhost/gleridis/' . $filename . '">' . $table_name . '</a></d
iv>';
$s .= '<div></div>';
return $s;
}

```

Τέλος, τα ονόματα των αρχείων αποθηκεύονται στη βάση δεδομένων της MySQL ενώ τα δεδομένα των μετρήσεων ανακτώνται από τον SQL Server και γράφονται στο αρχείο και καλείται η συνάρτηση create_graph() για τη δημιουργία του γραφήματος. Η συνάρτηση επιστρέφει HTML με συνδέσμους προς τα αρχεία και τα γραφήματα.

5.11 Επίλογος

Στο κεφάλαιο αυτό παρουσιάστηκαν τα πιο σημαντικά τμήματα του κώδικα που χρησιμοποιήθηκαν στην εφαρμογή. Έμφαση επίσης δόθηκε στην ασφάλεια της βάσης δεδομένων για την προστασία από κακόβουλους χρήστες. Για περισσότερες πληροφορίες και λεπτομέρειες σχετικά με τον κώδικα της εφαρμογής, οι ενδιαφερόμενοι μπορούν να έχουν πρόσβαση στην ηλεκτρονική έκδοση της τεκμηρίωσης στη γραμματεία του τμήματος.

Με την ολοκλήρωση του κεφαλαίου αυτού, τελειώνει το βιβλιογραφικό μέρος αυτής της πτυχιακής εργασίας. Ελπίδα είναι ο αναγνώστης να βρήκε ενδιαφέρουσα την όλη εφαρμογή και να τον βοηθήσει ως αναφορά σε κάποιο μελλοντικό του έργο.

ΒΙΒΛΙΟΓΡΑΦΙΑ

Έντυπα

Αναγνωστάκης, Α. (2010), *Ανάπτυξη – Κατασκευή συστήματος SCADA για τον σταθμό μετρήσεων του Α.Τ.Ε.Ι.*, Πτυχιακή Εργασία, ΑΤΕΙ Θεσσαλονίκης.

Δαμιανίδης, Χ., Αναγνωστόπουλος, Ν. (2010), *Σταθμός μέτρησης ποιότητας νερού*, Πτυχιακή Εργασία, ΑΤΕΙ Θεσσαλονίκης.

Μάργαρης, Α. (2002), *Σημειώσεις για το μάθημα Βάσεων Δεδομένων*, ΑΤΕΙ Θεσσαλονίκης.

Boronczyk, T., Naramore, E., Gerner, J., Scouarnec, Y., Stolz, J., Glass, M. (2009), *Beginning PHP6, Apache, MySQL web development*, Wiley Publishing, Indianapolis, USA.

Ramakrishnan, R., Gehrke, J. (2002), *Database Management Systems*, 2nd Edition, McGraw-Hill, NY, USA.

Ιστοσελίδες

http://www.ekke.gr/estia/gr_pages/F_synerg/KPE_Makrinitzas/METEOROLOGIA.pdf [Τελευταία πρόσβαση: 20/01/2011]

http://el.wikipedia.org/wiki/Microsoft_SQL_Server [Τελευταία πρόσβαση: 20/01/2011]

<http://el.wikipedia.org/wiki/%CE%9C%CE%B5%CF%84%CE%B5%CF%89%CF%81%CE%BF%CE%BB%CE%BF%CE%B3%CE%AF%CE%B1> [Τελευταία πρόσβαση: 20/01/2011]

http://en.wikipedia.org/wiki/HTTP_cookie [Τελευταία πρόσβαση: 20/01/2011]

http://en.wikipedia.org/wiki/File:HTTP_cookie_exchange.svg [Τελευταία πρόσβαση: 20/01/2011]

http://www.hnms.gr/hnms/greek/meteorology/full_story_html?dr_url=/docs/misc/pr ognosis&dr_page_number=2 [Τελευταία πρόσβαση: 20/01/2011]

Πτυχιακή εργασία του φοιτητή Χρήστου Γλερίδη

http://www.hnms.gr/hnms/greek/meteorology/full_story_html?dr_url=/hnms/docrep/docs/misc/prognosis_II [Τελευταία πρόσβαση: 20/01/2011]

[http://www.moa.gov.cy/moa/ms/ms.nsf/all/94C2FFF978F16A14C22576C8003677E0/\\$file/Weather_Forecast_HOW_ITS_DONE_GR.pdf?openelement](http://www.moa.gov.cy/moa/ms/ms.nsf/all/94C2FFF978F16A14C22576C8003677E0/$file/Weather_Forecast_HOW_ITS_DONE_GR.pdf?openelement) [Τελευταία πρόσβαση: 20/01/2011]

<http://www.php.net/> [Τελευταία πρόσβαση: 20/01/2011]

<http://www.phpfreaks.com/> [Τελευταία πρόσβαση: 20/01/2011]

<http://students.ceid.upatras.gr/~zafeiris/wather.doc> [Τελευταία πρόσβαση: 20/01/2011]

<http://www.tuxradar.com/practicalphp/10/1/0> [Τελευταία πρόσβαση: 20/01/2011]

Παράρτημα I

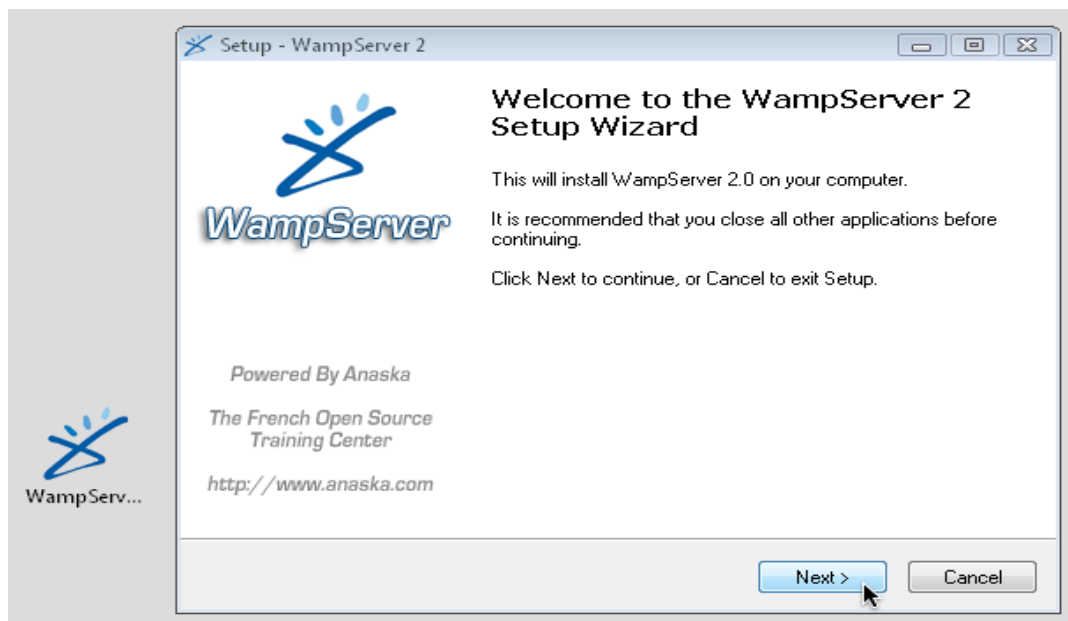
Εγχειρίδιο εγκατάστασης απαιτούμενων εργαλείων

Ο WampServer περιλαμβάνει τα προγράμματα Apache, MySQL, PHP και phpMyAdmin και εγκαθίσταται σε Windows περιβάλλοντα. Πρόκειται για ένα δωρεάν πρόγραμμα το οποίο περιλαμβάνει όλα τα αντίγραφα των πιο πρόσφατων εκδόσεων του Apache server, της script γλώσσας προγραμματισμού PHP, της MySQL καθώς και του εργαλείου χειρισμού βάσεων δεδομένων phpMyAdmin. Για την εγκατάστασή του ακολουθούνται τα παρακάτω βήματα:

1. Κατέβασμα της τελευταίας έκδοσης WampServer από την ιστοσελίδα <http://www.wampserver.com/en/> (η τρέχουσα έκδοση είναι η WampServer 2.0i [20/01/2011] και περιλαμβάνει τα Apache 2.2.11, MySQL 5.1.36, PHP 5.3.0).

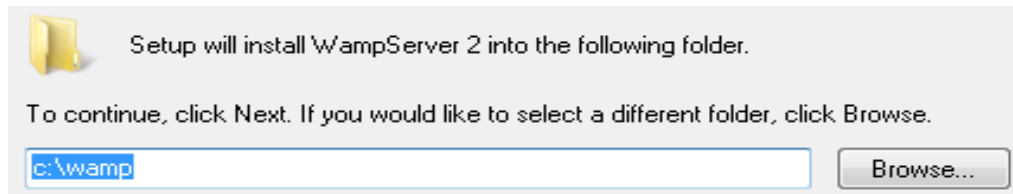
Προειδοποίηση: Σε περίπτωση που το WAMP5 είναι εγκατεστημένο στον υπολογιστή σας, σώσετε τα δεδομένα, κάνετε απεγκατάστασή του κι έπειτα εγκαταστήστε το WampServer 2.

Μόλις το αρχείο φορτωθεί στον υπολογιστή σας, κάντε διπλό κλικ στο αρχείο εγκατάστασης, όπως ακριβώς δείχνει η Εικόνα 1:



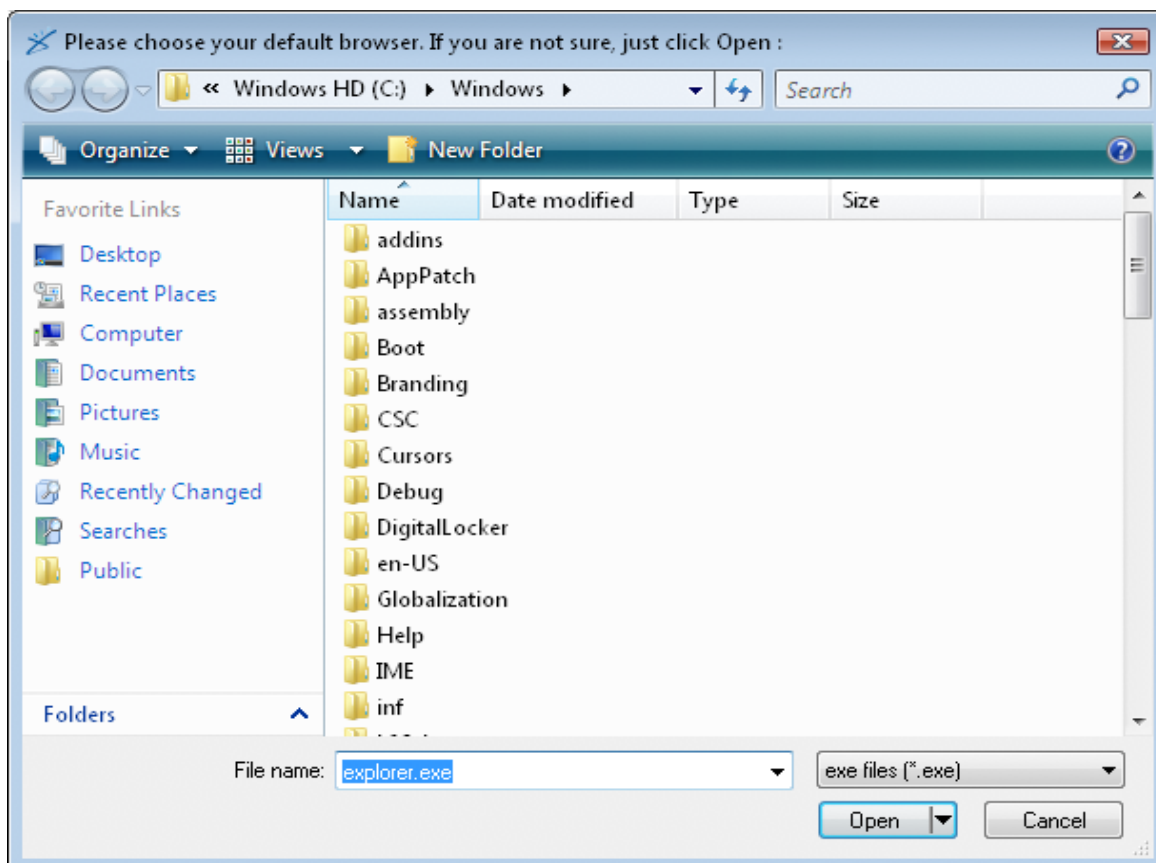
Εικόνα 1: Η αρχική οθόνη του WampServer 2

2. Το αρχείο εγκατάστασης θα ζητήσει τη θέση στην οποία θα εγκατασταθεί ο WampServer. Η προεπιλογή ορίζεται να είναι η θέση c:\wamp η οποία συνήθως προτείνεται. Μπορείτε όμως να καθορίσετε οποιαδήποτε θέση στον υπολογιστή σας.



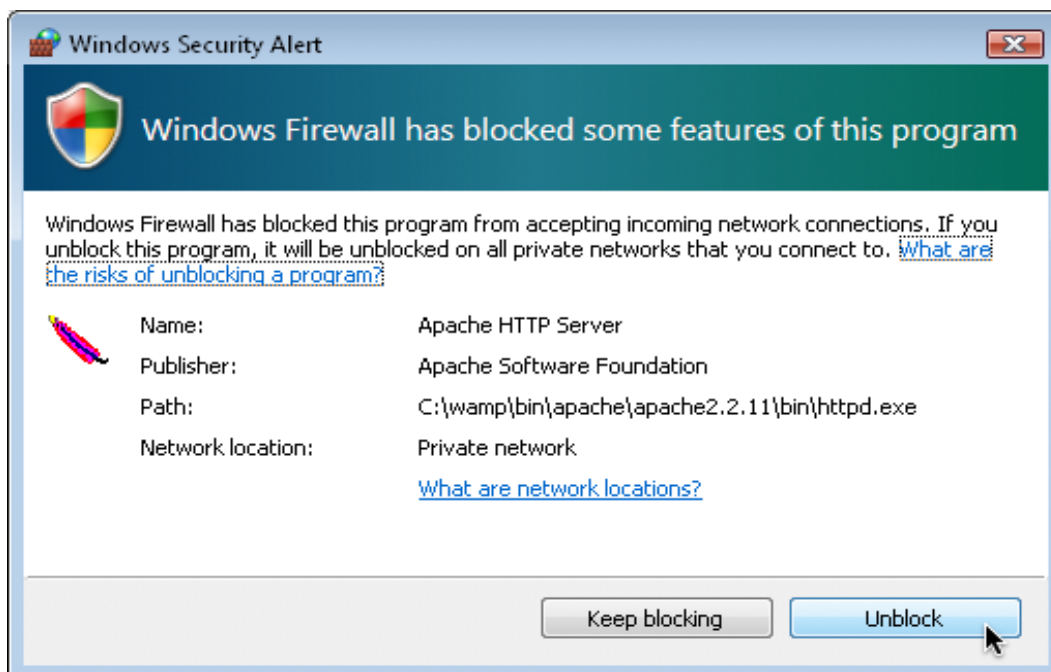
Εικόνα 1: Η προεπιλεγμένη θέση εγκατάστασης του WAMP

3. Μόλις η εγκατάσταση ολοκληρωθεί, ο WampServer θα ζητήσει να επιλεγεί ο προεπιλεγμένος browser. Αυτός θα είναι και ο browser που θα ανοίγει όταν ελέγχονται τα διάφορα project. Αν ο Firefox είναι εγκαταστημένος στο σύστημά σας, θα εμφανιστεί η ερώτηση αν επιθυμείτε να τον ορίσετε ως τον προεπιλεγμένο φυλλομετρητή όπου η PHP θα εκτελώντας τον κώδικα των διαφόρων script θα μεταφέρει στην οθόνη του υπολογιστή σας. Σε περίπτωση που η απάντηση είναι "No" κι έχετε κάποιον άλλον browser εγκαταστημένο τότε θα σας ζητηθεί να επιλέξετε ποιο θα είναι το εκτελέσιμο αρχείο του browser που επιθυμείτε να χρησιμοποιείτε αντί αυτού. Στην Εικόνα 1, επιλέγεται ο Internet Explorer (explorer.exe). Εναλλακτικά, μπορείτε να επιλέξετε τον Safari, Opera ή Google Chrome ή και κάποιο άλλος browser της επιλογής σας.



Εικόνα 1: Επιλογή προεπιλεγμένου browser

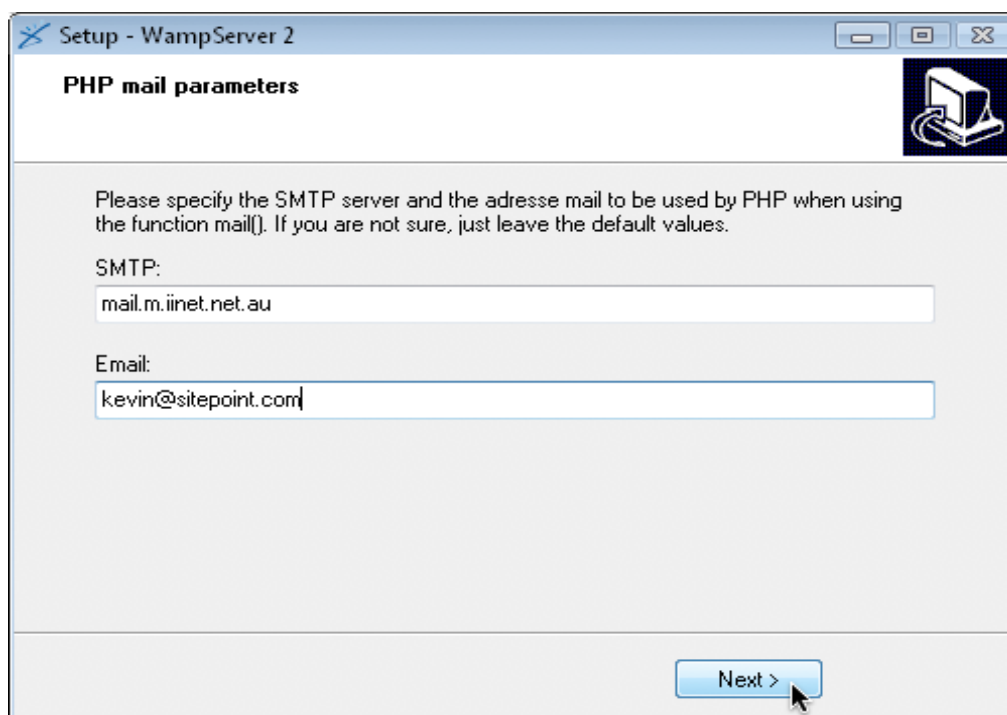
4. Μόλις ο WampServer εγκατασταθεί, θα εμφανιστεί το εικονίδιο του Apache HTTP Server. Τα Windows θα εμφανίσουν ένα μήνυμα ασφαλείας, όπως αυτό φαίνεται στην παρακάτω Εικόνα, αφού ο web server θα προσπαθήσει να αποκτήσει πρόσβαση στις αιτήσεις που κάνει ο browser.



Εικόνα 1: Μήνυμα ασφαλείας των Windows στην αίτηση του WampServer

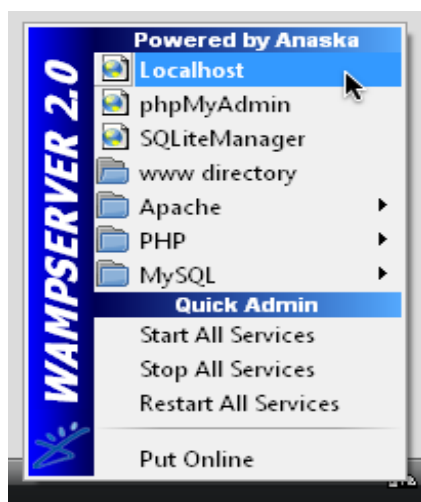
Αν θέλετε να είστε απολύτως σίγουροι ότι ο Apache θα απορρίπτει κάθε σύνδεση από το εξωτερικό περιβάλλον και ότι ο μόνος web browser που θα εκτελείται στον υπολογιστή σας θα μπορεί να προσπελαύνει σελίδες στον server εργασίας, τότε επιλέξτε “Keep blocking”. Ο WampServer έχει ενσωματωμένη την επιλογή να απορρίπτει τις συνδέσεις από το εξωτερικό περιβάλλον αν αυτό είναι επιθυμητό, όμως συστήνεται να επιλεγεί το “Unblock” ώστε να υπάρχει η ευελιξία της πρόσβασης στον server εργασίας και από το εξωτερικό περιβάλλον.

5. Στη συνέχεια, όπως φαίνεται και στην Εικόνα 1, το πρόγραμμα εγκατάστασης του WampServer, θα ζητήσει τον SMTP server και την διεύθυνση email που θα χρησιμοποιεί. Αυτό γίνεται επειδή ένα PHP script έχει τη δυνατότητα αποστολής μηνυμάτων ηλεκτρονικού ταχυδρομείου χρησιμοποιώντας το συγκεκριμένο πρωτόκολλο και από αυτές τις ρυθμίσεις εξαρτώνται ο server αποστολής email αλλά και το προεπιλεγμένο πεδίο “from” του email. Εισάγεται τη διεύθυνση email που επιθυμείτε και αν μπορείτε εισάγεται και τη διεύθυνση Service Provider’s SMTP. Πάντως, σε αυτό το σημείο μπορείτε να αφήσετε την προεπιλεγμένη τιμή ως έχει και να θέσετε την εκάστοτε διεύθυνση μέσα στο PHP script όταν κάτι τέτοιο απαιτείται.



Εικόνα 1: Εισάγετε την διεύθυνση Service Provider's SMTP

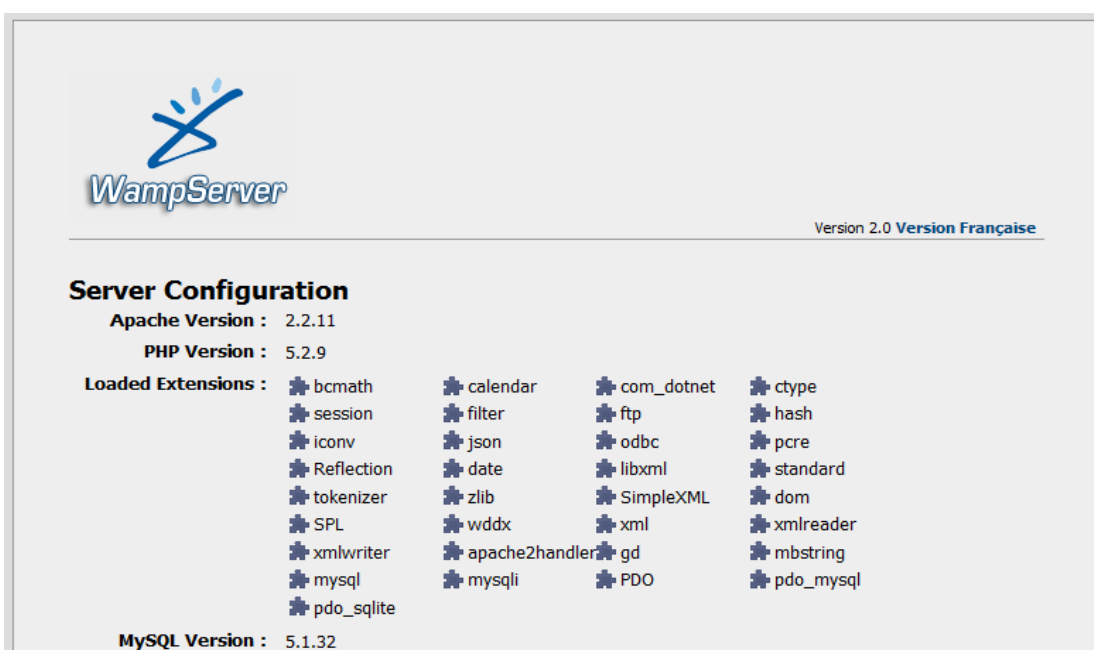
Μόλις η εγκατάσταση ολοκληρωθεί μπορείτε να τρέξετε τον WampServer. Θα εμφανιστεί ένα εικονίδιο στο System Tray των Windows στο δεξί μέρος της οθόνης. Κάντε κλικ σε αυτό και θα εμφανιστεί το μενού του WampServer το οποίο φαίνεται στην Εικόνα 1.



Εικόνα 1: Το μενού του WampServer

Από προεπιλογή, ο server θα είναι προσβάσιμος από browser που τρέχουν μόνο στον υπολογιστή που χρησιμοποιείται εσείς. Αν κάνετε όμως κλικ στην επιλογή του μενού “Put Online”, τότε ο server θα είναι προσβάσιμος από browser του εξωτερικού περιβάλλοντος.

Για να ελέγξετε αν ο WampServer λειτουργεί σωστά, κάντε κλικ στην επιλογή “Localhost”, η οποία βρίσκεται στην κορυφή του μενού του WampServer. Αυτό είναι και το προεπιλεγμένο όνομα του server στον υπολογιστή σας. Ο browser που θα ανοίξει πρέπει να εμφανίσει μία αρχική σελίδα όπως αυτής της Εικόνας 1.



Εικόνα 1: Αρχική σελίδα του WampServer

Μόλις τελειώσετε με τις εργασίες αυτές στον WampServer, μπορείτε να τον κλείσετε κάνοντας δεξί κλικ στο System Tray εικονίδιο και επιλέγοντας “Exit”.

Πλέον, έπειτα από την επιτυχημένη εκτέλεση των παραπάνω βημάτων μπορείτε να χρησιμοποιήσετε πλήρως τα εργαλεία που προσφέρει το WampServer.

Σημειώνεται ότι ανάλογη είναι και η εγκατάσταση των εργαλείων σε περιβάλλον Linux και Mac όπου παρόμοιες οδηγίες μπορείτε να βρείτε στις διευθύνσεις

Πτυχιακή εργασία του φοιτητή Χρήστου Γλερίδη

<http://www.linux.com/learn/tutorials/288158:easy-lamp-server-installation>

και

<http://www.mamp.info/en/index.html> αντίστοιχα.