



ΑΛΕΞΑΝΔΡΕΙΟ Τ.Ε.Ι. ΘΕΣΣΑΛΟΝΙΚΗΣ
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ



Web application for classroom/calendar Administration based on AJAX



Φοιτητής

Χαραλάμπους Μάριος

Αρ. Μητρώου: 07/3270

Επιβλέπων καθηγητής

Σιδηρόπουλος Αντώνης

Θεσσαλονίκη 2012

Περίληψη

Στην παρούσα πτυχιακή εργασία ζητήθηκε να δημιουργηθεί ένα διαδικτυακό ημερολόγιο το οποίο θα έπρεπε να κάνει χρήση κάποιων συγκεκριμένων τεχνολογιών.

Στα κεφάλαια που ακολουθούν θα υπάρξει μια σύντομη περιγραφή των τεχνολογιών που χρησιμοποιήθηκαν με μερικά παραδείγματα για την καλύτερη κατανόηση τους, περιγραφή της βάσης δεδομένων που έχει δημιουργηθεί για το ημερολόγιο, και τι στοιχεία θα καταχωρούνται στον κάθε πίνακα.

Στην συνέχεια ακολουθεί επεξήγηση του user interface σχετικά με τους ρόλους που υποστηρίζει το σύστημα και αναφορά στα θέματα ασφάλειας που υπάρχουν γενικά στο διαδίκτυο και τι μέτρα που λήφθηκαν

Τέλος υπάρχει μια γενική περιγραφή κάποιων ιδιαίτερων τμημάτων του κώδικα

Abstract

The management of all the halls of the school each semester sets a problem for teachers. With this application we want to be able to register their corresponding program using the halls of the department and to have a weekly show every lesson, teacher and hall. Also, teachers will have the ability to add, remove or modify commitments to a specific room for hours and/or days.

The system will have the ability to produce the appropriate documents (pdf program in every lesson, teacher, room, export to ics Yahoo, Google, MS Calendar, Desktop calendar and others), as well as the automatic creation of changes to the announcement.

The system will also enable the administrator to make changes any way.

Besides the above, this thesis aims to learn the programming language PHP and mySql database and also to create an integrated online application for better understanding of technologies HTML, Javascript, jQuery, AJAX and Json.

In the following chapters there will be an introduction and a short description of the technologies used.

Then will follow the explanation of the application in five steps

- I. Description and reference on the technologies that this system uses
- II. Description of the database,
- III. Understanding the user interface regarding the three available roles,
- IV. Reference on the system security and the steps that were taken.
- V. General description of some specific code fragments

Ευχαριστίες

Στο σημείο αυτό θα ήθελα να εκφράσω τις θερμές μου ευχαριστίες και την ευγνωμοσύνη μου στον κ Σιδηρόπουλο Αντώνη , τόσο για την εμπιστοσύνη που μου επέδειξε δίνοντας μου την πτυχιακή αυτή, η οποία ανταποκρίνεται στο αντικείμενο με το οποίο ήθελα να ασχοληθώ και να διευρύνω τις γνώσεις άμεσα, όσο και για την βοήθεια και τη καθοδήγηση στην πτυχιακή μου εργασία μέχρι τέλους, παρόλο τις δυσκολίες στην επικοινωνία και συνεννόηση που είχαμε λόγω της μεγάλης απόστασης και με μόλις δύο συναντήσεις καθ'όλη την διάρκεια.

Επίσης επιθυμώ να ευχαριστήσω τους γονείς μου Στέλιο και Καλλισθένη και τη φίλη μου Κλάρα Ευλογημένου για την βοήθεια, την κατανόηση, την υπομονή και την αμέριστη υποστήριξη και τέλος όλους όσους συνέβαλαν γενικότερα με την υποστήριξη τους στην ολοκλήρωση της πτυχιακής μου εργασίας.

Περιεχόμενα

Περίληψη	3
Abstract	4
Ευχαριστίες	5
Κεφάλαιο 1: Περιγραφή του προβλήματος.....	8
Κεφάλαιο 2. Απαιτήσεις συστήματος	9
2.1 Εισαγωγή.....	9
2.2 Προγράμματα που χρησιμοποιήθηκαν.	9
2.2.1 Eclipse [1] [2] [3].....	9
2.2.2 XAMPP [4] [5]	10
2.3 Περιγραφή των τεχνολογιών.	10
2.3.1 HTML (Hypertext Markup Language).....	11
2.3.2 DOM [33] [34] [35] [36].....	11
2.3.2 CSS [6] [7] [8]	12
2.3.3 JAVASCRIPT [9] [10] [11]	14
2.3.4 jQuery [12] [13] [14].....	14
2.3.5 jQuery – Datatables plug-in.....	16
2.3.6 AJAX [32] [40] [41].....	17
2.3.7 AJAX – jQuery [15] [16] [17] [18] [19]	19
2.3.8 JSON [20] [21].....	20
2.3.9 PHP	22
2.3.10 MySql.....	23
2.3.11 Διαδικτυακές υπηρεσίες	24
Κεφάλαιο 3 Σχεδιασμός	24
3.1 Εισαγωγή.....	24
3.2 ER Diagram.....	25
3.3 Ανάλυση βάσης δεδομένων.....	25
3.3.1 Semesters	25
3.3.2 Halls	26
3.3.3 Users.....	26
3.3.4 Holidays	26
3.3.5 Courses	27
3.3.6 Semester_courses	27
3.3.7 login_attempts	28

Πτυχιακή Εργασία του Χαραλάμπους Μάριου

3.3.8 halls_reservations	28
3.3.9 announcements.....	29
Κεφάλαιο 4 Αρχιτεκτονική και Αλγόριθμοι [22]	29
4.1 Εισαγωγή	29
4.2 Αρχιτεκτονική server-client 2 επιπέδων [23] [24].	29
4.3 Αρχιτεκτονική AJAX	30
4.4 Παράδειγμα 1 – Γέμισμα drop down στοιχείου	31
4.5 Παράδειγμα 2 – Καταχώρηση εγγραφής.....	33
4.6 Παράδειγμα 3 – Ενημέρωση εγγραφής.....	34
4.7 Παράδειγμα 4 – Διαγραφή εγγραφής.	37
4.8 Διατήρηση αρχείων ιστορικού.....	39
Κεφάλαιο 5 Ασφάλεια. [25] [27][28][29][30][31][37][38]	40
5.1 Brute force και επιθέσεις λεξικού.....	40
5.2 Denial of service (DoS)	41
5.3 Sql injection	41
5.4 Blind SQL injection.....	42
5.5 Rainbow tables	42
5.6 Μέτρα που λήφθηκαν.....	43
Κεφάλαιο 6 Τρεις ρόλοι και περιπτώσεις χρήσης τους [26].....	45
6.1 Απλοί χρήστες.	45
6.2 Καθηγητές	47
6.3 Διαχειριστές	52
Κεφάλαιο 7 Παράρτημα.....	60
7.1 Δημιουργία αντικειμένου JSON στον server.....	60
7.2 Μέθοδοι επαλήθευσης ημερομηνίας, url και email.	61
7.3 Διαδικασία δημιουργίας ενός Datatable.....	61
7.4 Διαδικασία προετοιμασίας δεδομένων το ημερολόγιο.	64
7.5 Εμφάνιση δεδομένων στο ημερολόγιο.	67
Βιβλιογραφία	68

Κεφάλαιο 1: Περιγραφή του προβλήματος

Η διαχείριση όλων των αιθουσών του τμήματος αποτελεί κάθε εξάμηνο σχετικό πρόβλημα για τους διδάσκοντες. Με την παρούσα εφαρμογή θέλουμε να υπάρχει η δυνατότητα να καταχωρείται το αντίστοιχο πρόγραμμα χρήσης των αιθουσών του τμήματος και να υπάρχει μια εβδομαδιαία προβολή ανά μάθημα, διδάσκοντα και αίθουσα. Επίσης οι διδάσκοντες θα έχουν την δυνατότητα να προσθέτουν , αφαιρούν ή να τροποποιούν δεσμεύσεις για μια αίθουσα για συγκεκριμένες ώρες ή/και μέρες.

Το σύστημα θα έχει την δυνατότητα εξαγωγής των κατάλληλων εγγράφων και δεδομένων σε αρχείο τύπου ics (τα οποία είναι συμβατά με Yahoo calendar, Google calendar, MS Calendar, Desktop calendar και άλλα) ανά μάθημα, διδάσκοντα, αίθουσα, καθώς επίσης και της αυτόματης δημιουργίας ανακοίνωσης αλλαγών του προγράμματος.

Το σύστημα επίσης θα δίνει τη δυνατότητα στον διαχειριστή να μπορεί να κάνει οποιαδήποτε αλλαγή επιθυμεί.

Εκτός από τα πιο πάνω, η παρούσα πτυχιακή εργασία αποσκοπεί στην εκμάθηση της γλώσσας προγραμματισμού PHP και της MySQL βάσης δεδομένων και επιπλέον στην δημιουργία μιας online ολοκληρωμένης εφαρμογής για την καλύτερη κατανόηση των τεχνολογιών HTML, Javascript, jQuery, AJAX και Json.

Στα παρακάτω κεφάλαια θα ακολουθήσει η εισαγωγή και μια μικρή περιγραφή των τεχνολογιών που χρησιμοποιήθηκαν.

Έπειτα θα ακολουθήσει η επεξήγηση της εφαρμογής σε πέντε στάδια:

- I. Περιγραφή των τεχνολογιών που χρησιμοποιήθηκαν
- II. Περιγραφή της βάσης δεδομένων,
- III. Επεξήγηση του user interface σχετικά με τους ρόλους που υποστηρίζει,
- IV. Αναφορά στα θέματα ασφάλειας και μέτρα που λήφθηκαν
- V. Γενική περιγραφή κάποιων ιδιαιτέρων τμημάτων του κώδικα

Κεφάλαιο 2. Απαιτήσεις συστήματος

2.1 Εισαγωγή.

Στο κεφάλαιο αυτό θα γίνει μια σύντομη παρουσίαση των προγραμμάτων που χρησιμοποιήθηκαν για τη σχεδίαση και ανάπτυξη της εφαρμογής, όπως και μια μικρή εισαγωγή και επεξήγηση των τεχνολογιών που χρησιμοποιήθηκαν.

Επίσης θα γίνει μια μικρή αναφορά στον ορισμό «Διαδικτυακές υπηρεσίες» οι οποίες στις μέρες μας χρησιμοποιούνται όλο και περισσότερο από όλους.

2.2 Προγράμματα που χρησιμοποιήθηκαν.

Τα προγράμματα που χρησιμοποιήθηκαν για τη παρούσα πτυχιακή εργασία είναι:

- Eclipse.
- XAMPP

2.2.1 Eclipse [1][2][3]

Eclipse είναι ένα πολύ-γλωσσικό περιβάλλον ανάπτυξης λογισμικού που αποτελείται από ένα ολοκληρωμένο περιβάλλον ανάπτυξης (IDE) και ένα σύστημα επέκτασης (plug-in). Είναι γραμμένο σε Java ως επί το πλείστον. Μπορεί να χρησιμοποιηθεί για την ανάπτυξη εφαρμογών σε Java και, μέσω των διαφόρων plug-ins, άλλες γλώσσες προγραμματισμού όπως Ada, C, C + +, COBOL, Perl, PHP, Python, E, Ruby (συμπεριλαμβανομένων των Ruby on Rails framework), Scala, Clojure κλπ. Μπορεί επίσης να χρησιμοποιηθεί για την ανάπτυξη πακέτων λογισμικού για το Mathematica. Το Eclipse περιλαμβάνει περιβάλλοντα ανάπτυξης Java development tools (JDT) για Java, Eclipse CDT για C / C + +, και το **Eclipse PDT** για την PHP, το οποίο και έχει χρησιμοποιηθεί στο έργο αυτό.

Το Eclipse δεν έχει ένα πρόγραμμα εγκατάστασης, μπορεί απλά να κατεβεί το πρόγραμμα, να αποσυμπιεστεί σε ένα φάκελο και να τρέξει το .exe. Μπορεί να εγκατασταθούν πολλαπλά αντίγραφα του Eclipse και να εκτελεστούν χωρίς να υπάρχει η ανησυχία για τις αλληλεπιδράσεις με το λειτουργικό σύστημα.

Εγκατάσταση του PDT (PHP Development Tools).

All-in-One: Είναι ο ευκολότερος τρόπος για να εγκατασταθεί το PDT, ειδικά αν δεν υπάρχει καμία προηγούμενη εγκατάσταση του Eclipse. Απλά χρειάζεται να κατεβεί το All-in-One πακέτο, να αποσυμπιεστεί στην επιθυμητή θέση, και να εκτελεστεί το εκτελέσιμο αρχείο eclipse που βρίσκεται σε αυτό. Η εγκατάσταση αυτή περιλαμβάνει όλα όσα χρειάζεστε για να έχετε ως μέρος του πακέτου, συμπεριλαμβανομένης της πλατφόρμας Eclipse, WTP (Web Εργαλεία έργου) και περισσότερο.

2.2.2 XAMPP [4] [5]

XAMPP είναι ένας cross-platform web server, που περιλαμβάνει κατά κύριο λόγο το Διακομιστή (server) HTTP Apache, MySQL βάση δεδομένων, και διερμηνείς (interpreters) για scripts γραμμένα σε γλώσσες προγραμματισμού PHP και Perl.

Το όνομα XAMPP είναι ένα ακρωνύμιο για:

- X (σημαίνει οποιοδήποτε λειτουργικό σύστημα)
- Apache (web server)
- MySQL (Βάση δεδομένων)
- PHP (scripting γλώσσα)
- Perl (scripting γλώσσα).

Με την εγκατάσταση του XAMPP έχει αυτόματα εγκατασταθεί στον υπολογιστή μας τα πάρα πάνω εργαλεία οπότεν μπορεί να εκτελεστεί οποιοδήποτε έτοιμο πρόγραμμα σε PHP με τη χρήση ενός browser το οποίο πρόγραμμα μπορεί να κάνει ή να μην κάνει χρήση βάσης δεδομένων.

Νόμιμα, οι σχεδιαστές του XAMPP προτείνουν να γίνεται χρήση του ως εργαλείο ανάπτυξης ιστοσελίδων, επιτρέποντας στους σχεδιαστές και προγραμματιστές να ελέγξουν τη δουλειά τους στους δικούς τους υπολογιστές χωρίς πρόσβαση στο Διαδίκτυο. Ο XAMPP παρέχει επίσης υποστήριξη για τη δημιουργία και χειρισμό βάσεων δεδομένων MySQL με το phpMyAdmin.

Το phpMyAdmin είναι ένα δωρεάν εργαλείο λογισμικού, γραμμένο σε PHP, που προορίζεται για τη διαχείριση της MySQL μέσω του World Wide Web. Το phpMyAdmin υποστηρίζει ένα ευρύ φάσμα δραστηριοτήτων με την MySQL, με τις πιο συχνά χρησιμοποιούμενες λειτουργίες που υποστηρίζονται από το περιβάλλον εργασίας χρήστη, όπως διαχείριση βάσεων δεδομένων, πίνακες, πεδία, σχέσεις, ευρετήρια, χρήστες, άδειες, κλπ, ενώ εξακολουθεί να υπάρχει η δυνατότητα να εκτελεστεί άμεσα κάποια SQL statements.

2.3 Περιγραφή των τεχνολογιών.

Οι τεχνολογίες που χρησιμοποιήθηκαν και θα περιγραφούν στην παρούσα πτυχιακή εργασία είναι:

- HTML
- DOM
- CSS
- Javascript
- jQuery
- jQuery – Datatables plug-in
- AJAX
- AJAX – jQuery
- JSON
- PHP
- MySql
- Διαδικτυακές υπηρεσίες.

2.3.1 HTML (Hypertext Markup Language)

Είναι η γλώσσα στην οποία είναι γραμμένες οι σελίδες. Επιτρέπει στις εικόνες να συνδυάζονται με το κείμενο και προσφέρει ένα ευρύ φάσμα δυνατοτήτων μορφοποίησης. Ένα από τα πιο σημαντικά χαρακτηριστικά της HTML είναι ο υπερ-σύνδεσμος (**hypertext**), που είναι ένα μέρος της ιστοσελίδας που συνδέεται με μια άλλη ιστοσελίδα. Κάνοντας κλικ σε μια υπερ-σύνδεση ανακατευθύνει τον περιηγητή σε άλλη σελίδα.

Η HTML βασίζεται στις ετικέτες σήμανσης (markup tags), οι οποίες συχνά αποκαλούνται HTML ετικέτες (HTML tags).

Οι ετικέτες αυτές είναι λέξεις-κλειδιά που περιβάλλονται από γωνιακές αγκύλες όπως πχ. <html>.

Οι ετικέτες έρχονται σε ζεύγη αρχής . <a> και τέλους όπως επίσης ονομάζονται ετικέτες ανοίγματος και κλεισίματος.

Είναι ένα αρχείο που περιέχει διάφορες HTML ετικέτες και σκέτο κείμενο και το οποίο έχει κατάληξη .html, ιστοσελίδα (web page).

HTML ιστορικά

Για να μπορούν οι browser να ερμηνεύουν σωστά την HTML πρέπει να πληρούνται κάποιες προδιαγραφές. Οι πρώτες προδιαγραφές ήταν η HTML 1.0 (1991) και HTML 2.0 (1995). Ακολούθησαν οι HTML 3.2 (1997) και η HTML 4.0 (1997) στους οποίους είχαν προστεθεί ακόμα περισσότερα χαρακτηριστικά, όπως η υποστήριξη για γλώσσες σεναρίων, έτσι ώστε να επιτρέπεται το δυναμικό περιεχόμενο. Οι ραγδαίες αλλαγές στον παγκόσμιο ιστό, και οι ανάγκες για standards και πλήρη συμβατότητα, οδήγησε το World Wide Web Consortium (W3C) και το Web Hypertext Application Technology Working Group (WHATWG) να συνεργαστούν και να δημιουργήσουν την νέα έκδοση την HTML5. Η εν λόγω έκδοση επιτρέπει την ελαχιστοποίηση των plugins, καθώς είναι καλύτερη λειτουργία για τον εντοπισμό λαθών. Διαθέτει περισσότερο markup για να αντικατασταθεί το scripting και πλήρη συμβατότητα ανεξαρτήτως συσκευής. Τα πιο σημαντικά χαρακτηριστικά της είναι ο καμβάς χρωμάτων για ζωγραφική, τα στοιχεία video και audio για αναπαραγωγή πολυμέσων, τα νέα στοιχεία περιεχομένου όπως τα footer, header, nav και section και τα νέα στοιχεία δημιουργίας φόρμας όπως τα calendar, date, time, email, url και search.

2.3.2 DOM [33] [34] [35] [36]

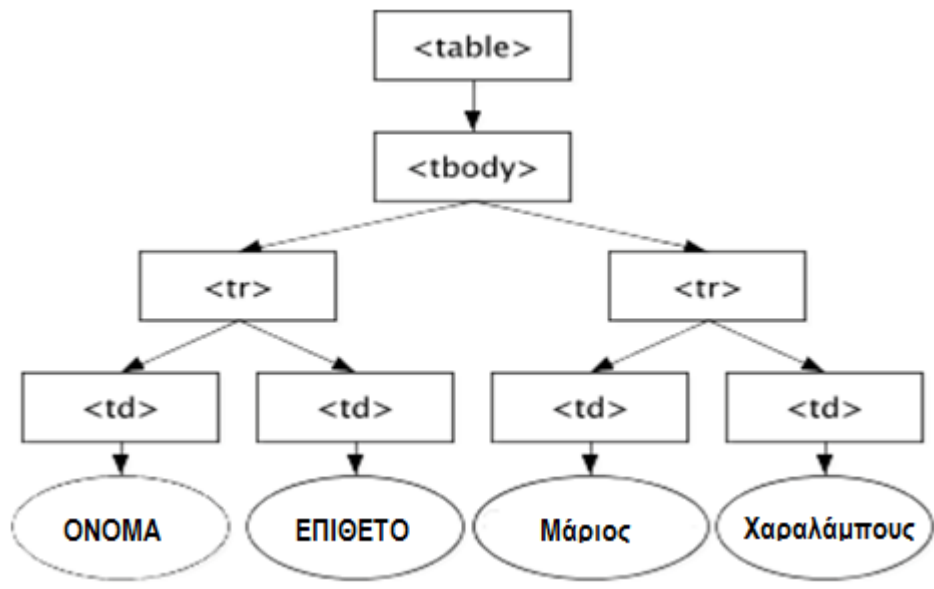
Το Μοντέλο Αντικειμενικού Εγγράφου χρησιμοποιείται για την δυναμική έκθεση των δεδομένων καθώς και για την αλληλεπίδραση με αυτά. Συγκεκριμένα το DOM είναι μια σύμβαση για τον συμβολισμό και την αλληλεπίδραση των αντικειμένων μέσα σε ένα έγγραφο HTML, XHTML και XML. Είναι ανεξάρτητο του λειτουργικού συστήματος και

της γλώσσας προγραμματισμού που χρησιμοποιείται. Ένας φυλλομετρητής δεν είναι υποχρεωμένος να χρησιμοποιεί το μοντέλο αυτό για να απεικονίσει ένα έγγραφο HTML. Ωστόσο απαιτείται από τη γλώσσα Javascript όταν πρόκειται να επιθεωρήσει (inspect) ή επεξεργαστεί περαιτέρω το έγγραφο, τα αποτελέσματα αυτά να ενσωματώνονται στον ιστότοπο που παρουσιάζεται στο φυλλομετρητή.

Το DOM αποδομεί και παρουσιάζει ένα HTML έγγραφο σε δενδρική μορφή. Κατά τη διάρκεια αυτής της διαδικασίας απομνημονεύει αρχικά όλο το έγγραφο, για το λόγο αυτό τα προγράμματα που χρησιμοποιούν DOM επιβάλλεται να έχουν μεγάλη μνήμη.

Παράδειγμα απεικόνισης ενός πίνακα σε δέντρο DOM.

ΟΝΟΜΑ	ΕΠΙΘΕΤΟ
Μάριος	Χαραλάμπους



2.3.2 CSS [6] [7] [8]

Το πρότυπο CSS είναι μια γλώσσα υπολογιστή που ανήκει στην κατηγορία των γλωσσών φύλλων στυλ. Είναι ένα σύνολο από εντολές τις οποίες ο φυλλομετρητής (browser) ακολουθεί ώστε να αλλάξει την προεπιλεγμένη παρουσίαση των HTML στοιχείων ανάλογα με τις τιμές που έχουν οριστεί. Μέσω του προτύπου CSS επιτυγχάνεται η εύκολη και μαζική μορφοποίηση, δίνοντας τη δυνατότητα στο σχεδιαστή να ξεχωρίζει το περιεχόμενο των σελίδων από το σχεδιασμό, έχοντας ως αποτέλεσμα οι αλλαγές καθώς και οι επανασχεδιασμοί να γίνονται πολύ πιο εύκολα σε μια ιστοσελίδα.

Η εισαγωγή ενός CSS προτύπου σε μια HTML σελίδα μπορεί να γίνει με δύο τρόπους:

Ενσωματώνοντας τον CSS δηλώσεις μέσα στα `<head>` `</head>` tags ο οποίος κώδικας θα πρέπει να βρίσκεται ανάμεσα στα `<Style type="TEXT/CSS">` `</Style>` tags.

Δημιουργώντας ένα αρχείο στο οποίο θα υπάρχουν όλες οι δηλώσεις το αρχείο θα εισαχθεί στην HTML σελίδα μας μέσα στα `<head>``</head>` tags με αυτή τη δήλωση `<link rel="stylesheet" type="text/css" href="something.css" />`. Το αρχείο αυτό όπως φαίνεται, πρέπει να έχει την κατάληξη «.css»

Σημαντικότεροι CSS selectors

Ο επιλογέας (selector) συγκεκριμενοποιεί το στοιχείο το οποίο πρόκειται να μορφοποιηθεί. Κάθε δήλωση (declaration) αποτελείται από δύο στοιχεία, την ιδιότητα (property) και την τιμή (value). Η δήλωση γίνεται ανάμεσα σε ένα ζεύγος άγκιστρων { }. Υπάρχει η δυνατότητα να εφαρμοστούν πολλές δηλώσεις σε ένα συγκεκριμένο selector, η κάθε μία από τις οποίες μπορεί να μορφοποιεί ξεχωριστά σημεία κάθε στοιχείου ή ακόμα και όλα εξαρχής. Κάθε ιδιότητα χωρίζεται από την τιμή της με άνω κάτω τελεία (:). Η δήλωση ολοκληρώνετε με ερωτηματικό (;).

Παράδειγμα: `selector{ property : value; }, body{ color : black; }`

Ο universal selector είναι ο λιγότερο συγκεκριμένος selector αφού επιλέγει όλα τα στοιχεία. Εκφράζεται με το σημείο του αστερίσκου (*) και λειτουργεί ως wild card για την επιλογή κάθε στοιχείου στο έγγραφο.

Ο element selector επιλέγει όλα τα στιγμιότυπα ενός στοιχείου τα οποία καθορίζονται από το όνομα της ετικέτας τους. Σηματοδοτείται με το είδος του κάθε στοιχείου. Πχ. για την επιλογή όλων των φορμών `<form>` είναι (form).

Ένας class selector στοχεύει στα στοιχεία που ανήκουν σε μία συγκεκριμένη κλάση. Σηματοδοτείται με την τελεία (.).

Παράδειγμα: `.class{ property : value; }, .form-fields{ color : blue; }`

Ο id selector είναι πολύ συγκεκριμένος αφού επιλέγει μόνο το στοιχείο που φέρει το αντίστοιχο id και είναι μοναδικό. Σηματοδοτείται με το κάγκελο (#).

Τα ψευδοστοιχεία δημιουργούν γενικεύσεις σχετικά με το έγγραφο πέραν του βαθμού που καθορίζονται από την εκάστοτε γλώσσα. Ονομαστικά τα ισχυρότερα εργαλεία είναι τα ακόλουθα `:link { }` `:visited { }` `:active { }` `:hover { }` `:focus { }` `:first-line` `:first-letter` `:before` `:after`

Οι child selectors επιλέγουν τα παιδιά πρώτου επιπέδου αντί όλους τους απογόνους. Σηματοδοτούνται με το συμβόλου (>).

Παράδειγμα: `father > child{ property : value; }, form > .form-fields{ color : blue; }`

2.3.3 JAVASCRIPT [9] [10] [11]

Η Javascript είναι μια διερμηνευτική (interpreted) γλώσσα προγραμματισμού που τρέχει στην πλευρά του φυλλομετρητή (client side), το οποίο την κάνει πιο γρήγορη από κάποια άλλη γλώσσα που τρέχει στον διακομιστή (server side). Φυσικά αυτό έχει ένα μειονέκτημα ότι ο πηγαίος κώδικας είναι προσβάσιμος από όλους τους χρήστες της ιστοσελίδας.

Σχεδιάστηκε για να δώσει στους προγραμματιστές ένα εργαλείο με απλή σύνταξη. Η Javascript, όπως και όλες οι γλώσσες προγραμματισμού, έχει μεταβλητές, αλφαριθμητικά, πίνακες, αντικείμενα συναρτήσεις και όλες τις συνηθισμένες δομές ελέγχου. Διαθέτει λειτουργίες ειδικά για ιστοσελίδες, όπως δυνατότητα διαχείρισης παραθύρων και πλαισίων, αποστολή-λήψη και δημιουργία-κατάργηση cookies, χειρισμός φορμών και άλλων στοιχείων που βρίσκονται στη σελίδα.

Έχει επίσης πολλές ιδιότητες αντικειμενοστραφούς γλώσσας και πολλές ομοιότητες με τις άλλες γλώσσες αφού κτίστηκε πάνω στο πρότυπο των γλωσσών C, C++ και Java. Έχει όμως μία πολύ σημαντική διαφορά, διαχειρίζεται τους τύπους δεδομένων πολύ πιο χαλαρά, για παράδειγμα μια μεταβλητή στην αρχή μπορεί να είναι τύπου *int* και στη συνέχεια να γίνει τύπου *string*. Αξίζει να σημειωθεί στο παρόν στάδιο πως η Java και η Javascript είναι δυο εντελώς διαφορετικές γλώσσες προγραμματισμού.

Με την JavaScript μπορεί να επιτευχθεί η μειωμένη αποστολή δεδομένων στον διακομιστή (server) όπως και το φιλτράρισμα των δεδομένων που θα αποσταλούν, λόγω του ότι δίνεται η δυνατότητα να ελεγχθούν τα δεδομένα ή ακόμη και να αλλάξουν μέσα από την φόρμα πριν την αποστολή τους στον διακομιστή. Επίσης υπάρχει η δυνατότητα να αλλάξουν οι τιμές και ιδιότητες στοιχείων της σελίδα δυναμικά πολύ εύκολα.

Τέλος, η JavaScript υποστηρίζεται από όλους τους browsers και προσφέρει ένα καλύτερο και ελαφρότερο γραφικό περιβάλλον αφού τα σενάρια (scripts) είναι μικρά σε μέγεθος και αποθηκεύονται στη μνήμη του browser μόλις αυτά κατέβουν.

2.3.4 jQuery [12] [13] [14]

Η jQuery είναι η δημοφιλέστερη cross-browser JavaScript βιβλιοθήκη η οποία δίνει έμφαση στην αλληλεπίδραση μεταξύ JavaScript και HTML. Απλοποιεί πολύ τη διαδικασία συγγραφής σεναρίων. Επιπλέον, είναι λογισμικό ανοιχτού κώδικα.

Η βιβλιοθήκη jQuery είναι ειδικά σχεδιασμένη για επιλογή HTML στοιχείων, για εφαρμογή κάποιας ενέργειας σε αυτά, αλλά και για χειρισμό CSS. Στόχος της είναι η εύκολη πλοήγηση ανάμεσα στους κόμβων ενός εγγράφου, η επιλογή στοιχείων του DOM, η δημιουργία Animations, η διαχείριση γεγονότων (events), και να αναπτύσσει διαδικτυακές εφαρμογές AJAX. Παρέχει την δυνατότητα αλλαγής εμφάνισης μιας ιστοσελίδας χωρίς να υπάρχει θέμα ασυμβατότητας στους διάφορους φυλλομετρητές (browsers). Επίσης, δίνει τη δυνατότητα στους προγραμματιστές να επεκτείνουν τη βιβλιοθήκη, δημιουργώντας χαμηλού επιπέδου αλληλεπιδράσεις, animations,

προχωρημένα εφέ και υψηλού επιπέδου widgets. Συμβάλλει έτσι, δραστικά στην δημιουργία ισχυρών και δυναμικών ιστοσελίδων.

Σύνταξη jQuery

Η βασική σύνταξη είναι: jQuery(selector).action() όπου το «jQuery» μπορεί να αντικατασταθεί από το «\$» για συντομία.

Μερικοί από τους κύριους επιλογείς (selectors):

- Επιλογέας κλάσης (“.class”). Επιλέγει όλα τα αντικείμενα με αυτήν την κλάση.
- Επιλογέας παιδιού (“parent > child”). Επιλέγει όλα τα παιδιά αυτού του γονέα.
- Επιλογέας βάση id (“#id”). Επιλέγει το στοιχείο με αυτό το id.

Μερικές από τις πιο συχνά χρησιμοποιούμενες μεθόδους (actions)

- .append(). Εισάγει το περιεχόμενο της παρένθεσης στο τέλος του επιλεγμένου στοιχείου.
- .addClass(). Προσθέτει κλάση(εις) στα επιλεγμένα στοιχεία.
- .mouseover(), .mouseout(), .mouseenter(), .mouseleave(), .mouseup(), . Προσθέτει αυτά τα events στα επιλεγμένα στοιχεία.
- .val(). Επιστρέφει την τιμή του επιλεγμένου στοιχείου.
- .html(). Επιστρέφει το Html περιεχόμενο των επιλεγμένων στοιχείων.

Όπως διαφαίνεται από τα παραδείγματα, ο κώδικας που χρειάζεται για πάρα πολλές λειτουργίες έχει ελαχιστοποιηθεί καθώς επίσης είναι πιο κατανοητός.

Παράδειγμα

Παρατίθεται κώδικας σε javascript κάνοντας και μη χρήση jQuery βιβλιοθήκης.

```
1 <!-- Απλή JavaScript -->
2 <script type="text/javascript">
3     function formSubmitted(){
4         var elem = document.getElementById('form').elements;
5         for(var i = 0; i < elem.length; i++){
6             alert(elem[i].value);
7         }
8     }
9 </script>
10
11 <!-- Κάνοντας χρήση jQuery -->
12 <script type="text/javascript">
13     $('#form').submit(function() {
14         $('#form').each(function() {
15             alert($(this).val());
16         })
17     })
18 </script>
```

Δεν υπάρχει μεγάλη διαφορά σε θέμα γραμμών κώδικα αλλά είναι εμφανές πως είναι πιο κατανοητός και πιο απλός στη χρήση του.

Εισαγωγή της jQuery βιβλιοθήκης.

Πολλές μεγάλες επιχειρήσεις φιλοξενούν αντίγραφα της βιβλιοθήκης αυτής στα δίκτυα τους που είναι διαθέσιμα για δημόσια χρήση. Μερικοί από τους συνδέσμους είναι:

- Google Ajax API CDN (Also supports SSL via HTTPS)
 - <http://ajax.googleapis.com/ajax/libs/jquery/1.7.2/jquery.min.js>
 - <http://code.google.com/apis/ajaxlibs/documentation/index.html#jquery> [Google Ajax CDN Documentation]
- Microsoft CDN (Also supports SSL via HTTPS)
 - <http://ajax.aspnetcdn.com/ajax/jQuery/jquery-1.7.2.min.js>
 - [Ajax CDN Announcement](#), [Microsoft Ajax CDN Documentation](#)
- jQuery CDN (via [Media Temple](#))
 - <http://code.jquery.com/jquery-1.7.2.min.js> Minified version
 - <http://code.jquery.com/jquery-1.7.2.js> Development (Full) version

Ο πιο εύκολος τρόπος για να εισαχθεί η βιβλιοθήκη αυτή στην ιστοσελίδα μας είναι με την εισαγωγή `<script type="text/javascript" src="source"></script>` όπου στο «source» θα μπει το link που θα δείχνει σε έναν από τους διακομιστές που φιλοξενούν τον πηγαίο κώδικα της βιβλιοθήκης.

2.3.5 jQuery – Datatables plug-in

Το DataTables είναι ένα plug-in για την jQuery βιβλιοθήκη της Javascript. Πρόκειται για ένα ιδιαίτερα ευέλικτο εργαλείο, που βασίζεται πάνω στα θεμέλια της προοδευτικής ενίσχυσης, η οποία έχει προσθέσει προηγμένες ελέγχους αλληλεπίδραση σε κάθε πίνακα HTML.

Έχει την δυνατότητα να προσφέρει:

- Σελιδοποίηση
- Φιλτράρισμα δεδομένων «on the fly»
- Αυτόματη αλλαγή πλάτους κελιών
- Απόκρυψη κάποιων στηλών που είναι επιθυμητό να μην φαίνονται (πχ. ids)
- Αυτόματη φόρτωση δεδομένων με AJAX
- Και πολλά άλλα μερικά από τα οποία θα αναλυθούν σε επόμενο κεφάλαιο.

Κάτι πολύ σημαντικό όμως είναι το γεγονός ότι μπορεί να δημιουργηθεί ένα DataTable από έναν στατικό HTML table, ο οποίος υπάρχει στη σελίδα, αλλά μπορεί επίσης να

δημιουργηθεί δίνοντας του όρισμα ένα JavaScript array, JSON array, PHP-C#-JSP array ως απάντηση από ένα AJAX call, ή ακόμη από ένα XML αρχείο.

Επίσης, έχουν αναπτυχθεί και άλλες προεκτάσεις αυτού του plug-in, οι οποίες δίνουν ακόμη περισσότερη ευελιξία και ευκολία στην δημιουργία web εφαρμογών, όπως plug-in για προσθήκη μεταφράσεων, ταξινόμηση βάση 20+ διαφορετικών τρόπων, σελιδοποίηση, δυναμική προσθήκη, διαγραφή και ενημέρωση των στοιχείων των κελιών με χρήση AJAX κ.α.

2.3.6 AJAX [32] [40] [41]

Για χρόνια, οι σχεδιαστές ιστοσελίδων παραπονούνταν για την αδυναμία να δημιουργήσουν εξελιγμένες διεπαφές που να ανταποκρίνονται, κάτι σαν αυτές που βρίσκουμε στις desktop εφαρμογές. Όλα άρχισαν να αλλάζουν το 2005, όταν ο Jesse James Garrett, επινόησε τον όρο Ajax σε ένα άρθρο "**Ajax**: Μια νέα προσέγγιση για εφαρμογές στον Παγκόσμιο Ιστό", οι οποίες βασίζονται σε τεχνικές που χρησιμοποιούνται στις ιστοσελίδες της **Google**.

Ajax ένα ακρωνύμιο για Asynchronous JavaScript και **XML**, είναι ένα σύνολο αλληλένδετων τεχνικών ανάπτυξης ιστοσελίδων που χρησιμοποιείται στην πλευρά του πελάτη και δημιουργεί web εφαρμογές που δουλεύουν ασύγχρονα.

Στον παραδοσιακό προγραμματισμό με JavaScript, αν θέλει κάποιος να πάρει πληροφορίες από μια βάση δεδομένων ή ένα αρχείο από τον server, ή να στείλει πληροφορίες για το χρήστη σε ένα server, θα πρέπει να κάνει μια HTML φόρμα και με χρήση GET ή POST να στείλει τα δεδομένα στο διακομιστή (server). Ο χρήστης θα χρειαστεί να κάνετε κλικ στο κουμπί "Υποβολή" για να στείλει / λάβει τις πληροφορίες, να περιμένει το διακομιστή να απαντήσει, και στη συνέχεια μια νέα σελίδα θα φορτωθεί με τα αποτελέσματα.

Επειδή ο διακομιστής επιστρέφει μια νέα σελίδα κάθε φορά που ο χρήστης υποβάλλει αίτημα, οι παραδοσιακές web εφαρμογές τρέχουν αργά και τείνουν να είναι λιγότερο φιλικές προς το χρήστη. Με την AJAX, η JavaScript επικοινωνεί απευθείας με το διακομιστή, μέσω του αντικειμένου XMLHttpRequest της JavaScript.

Με αίτηση HTTP, μια ιστοσελίδα μπορεί να υποβάλει μία αίτηση και να πάρει μία απάντηση από έναν web server, χωρίς την επαναφόρτωση της σελίδας. Ο χρήστης θα παραμείνει στην ίδια σελίδα, και δεν θα παρατηρήσει πως σενάρια (scripts) που τρέχουν στο παρασκήνιο αιτούνται σελίδες, ή στέλνουν δεδομένα σε ένα διακομιστή.

Παρά το όνομα της, η χρήση της **XML** δεν είναι απαραίτητη, αντ' αυτού, συχνά χρησιμοποιείται κωδικοποίηση **JSON**, η οποία έχει χρησιμοποιηθεί στην παρούσα πτυχιακή εργασία. Επίσης, οι αιτήσεις δεν χρειάζεται να γίνονται ασύγχρονα.

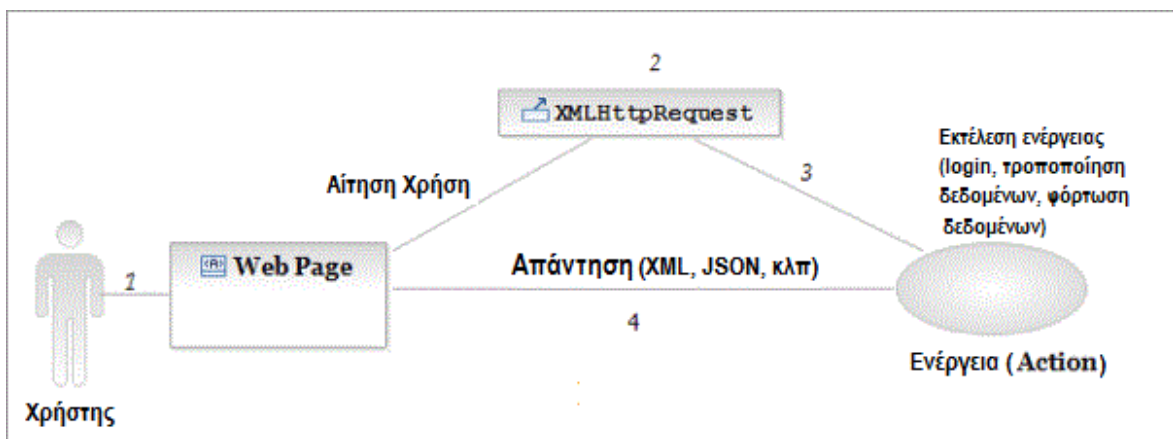
Το αντικείμενο XMLHttpRequest

Το **XMLHttpRequest** είναι μία διεπαφή ανάπτυξης εφαρμογών (API) του μοντέλου **DOM**. Μπορεί να χρησιμοποιηθεί σε μία γλώσσα προγραμματισμού **client-side** όπως η **JavaScript** για να στείλει ένα αίτημα **HTTP** ή **HTTPS** σε ένα Διακομιστή διαδικτύου και να φορτώσει τα δεδομένα απάντησης του διακομιστή πίσω στη γλώσσα. Από τη στιγμή που τα δεδομένα βρεθούν μέσα στη γλώσσα, είναι διαθέσιμα ως ένα **XML** έγγραφο, εφόσον η απάντηση είχε έγκυρη σήμανση και ως «απλό κείμενο». Τα δεδομένα απάντησης μπορούν να χρησιμοποιηθούν για να ενημερωθεί το περιεχόμενο της σελίδας χωρίς να χρειάζεται επαναφόρτωση του browser.

Το απλό κείμενο ως απάντηση, όπως χρησιμοποιήθηκε και στην παρούσα πτυχιακή εργασία, μπορεί από τον browser να διαμορφωθεί ως **JSON (JavaScript Object Notation)** και να αξιολογηθεί από την JavaScript για να δημιουργηθεί ένα αντικείμενο δεδομένων για χρήση στο τρέχον μοντέλο **DOM**.

Το αντικείμενο XMLHttpRequest παίζει ίσως τον πιο σημαντικό ρόλο στην διαδικτυακή τεχνική ανάπτυξης AJAX.

Σχήμα AJAX



Το πρόβλημα με την AJAX

Παρά την επαναστατική επίπτωση της στο διαδίκτυο, η AJAX μπορεί να είναι δύσκολο να χρησιμοποιηθεί ακόμα και για τους πιο έμπειρους web developers.

Οι άνθρωποι συνήθως δημιουργούν δικές τους προσαρμοσμένες μεθόδους για να χειριστούν Ajax κλήσεις, έτσι ώστε η λειτουργικότητα των μεθόδων αυτών να μπορεί να επαναχρησιμοποιηθεί σε μια διαδικτυακή εφαρμογή. Αυτό μπορεί να καταλείξει πολύ κουραστικό όταν μια διαδικτυακή εφαρμογή κάνει χρήση διαφορετικών τύπων του Ajax κλήσεων, όπως για παράδειγμα την ικανότητα να χειρίζεται τόσο σύγχρονων και ασύγχρονων κλήσεων ή την ικανότητα να χειρίζεται διάφορες μορφές δεδομένων που επιστρέφονται από τον server (response format) όπως string, HTML, CSV, XML, JSON, κλπ

```

1 <!-- Ajax - JavaScript -->
2 <script type="text/javascript">
3 function ajaxRequest(theURL, sendString, callbackFunction){
4     var thisRequestObject;
5     thisRequestObject = initiateRequest();
6     thisRequestObject.onreadystatechange = processRequest;
7     function initiateRequest(){
8         if (window.XMLHttpRequest)
9             return new XMLHttpRequest();
10        elseif (window.ActiveXObject)
11            return new ActiveXObject("Microsoft.XMLHTTP");
12    }
13    function processRequest(){
14        if (thisRequestObject.readyState == 4){
15            if (thisRequestObject.status == 200){
16                if (callbackFunction)
17                    callbackFunction(thisRequestObject, sendString);
18            }
19            else
20                alert("There was an error: (" + thisRequestObject.status + ") " +
21                    thisRequestObject.statusText);
22        }
23    }
24    this.sendPostData = function(){
25        if (theURL)
26        {
27            thisRequestObject.open("POST", theURL, true);
28            thisRequestObject.setRequestHeader("Content-Type",
29                "application/x-www-form-urlencoded");
30            thisRequestObject.send(sendString);
31        }
32    }
33 }
34 </script>

```

Παραδοσιακά μία AJAX κλήση έχει την μορφή:

Η ιδέα του Ajax όπως έχουμε περιγράψει και πιο πριν είναι η αποστολή ενός αιτήματος σε κάποιο web server η αναμονή μιας απάντησης, και έπειτα η ενημέρωση της HTML σελίδας. Θα ήταν βολική η δημιουργία μιας μεθόδου, όπως η ajaxRequest(), που θα λαμβάνει δύο παραμέτρους - το αιτούμενο URL και μια μέθοδο ανάλογα με την απάντηση που θα μπορούσε να επαναχρησιμοποιηθεί σε όλο το σύστημά. Η ιδέα αυτή όμως σύντομα αποδεικνύεται ότι δεν είναι και η πιο απλή, αφού όπως λέχθηκε και πιο πάνω, απαιτείται ο χειρισμός διαφορετικών ειδών κλήσεων, διαφορετικών ειδών απαντήσεων κλπ.

2.3.7 AJAX – jQuery [15] [16] [17] [18] [19]

Σε αυτό το σημείο είναι αφάνταστα βολική η χρήση της τεχνολογίας - βιβλιοθήκης που περιγράφηκε πιο πάνω, της jQuery, καθώς με 10 γραμμές κώδικα υλοποιούνται πράγματα με την ίδια και καλύτερη λειτουργικότητα και με πιο ευανάγνωστο κώδικα, όπου κανονικά θα χρειάζονταν 3πλάσιο ίσως και περισσότερο κώδικα.

```

32 </script>
33 |
34 <!-- Ajax - jQuery -->
35 <script type="text/javascript">
36     $.Ajax({
37         type: "POST",
38         url: "some.php",
39         data: "name=Marios&location=Limassol",
40         succedd: function(msg){
41             alert("Data saved: " + msg);
42         },
43         error: (function(err) { alert("Error: " + err); }),
44         complete: (function() { alert("Completed"); })
45     })
46 </script>

```

Η jQuery έχει βάλει όλες του τις ιδιότητες της AJAX σε ένα απλό API. Υπάρχει η δυνατότητα του ελέγχου για τα πάντα, συμπεριλαμβανομένης της διεύθυνσης, μνήμης, της λειτουργίας σε περίπτωση επιτυχίας ή/και αποτυχίας, τον τύπο δεδομένων, ακόμα και συγχρονισμού. Όλα αυτά από μια τακτοποιημένη δήλωση.

2.3.8 JSON [20] [21]

Το JSON είναι ακρώνυμο του JavaScript Object Notation και είναι ένα format μορφοποιήσεις δεδομένων σε javascript όπως έχει περιγραφεί μερικώς λίγο πιο πριν. Το format του είναι πολύ απλό και τυποποιημένο. Έχει απλή ιεραρχική μορφή, ανεξάρτητη από την γλώσσα ή την πλατφόρμα στην οποία χρησιμοποιείται.

Είναι ένα πρότυπο κειμένου το οποίο είναι τελείως ανεξάρτητο από γλώσσες προγραμματισμού, αλλά χρησιμοποιεί πρακτικές (conventions) οι οποίες είναι γνωστές στους προγραμματιστές της οικογένειας προγραμματισμού C, συμπεριλαμβανομένων των C, C++, C#, Java, JavaScript, Perl, Python, και πολλών άλλων. Αυτές οι ιδιότητες κάνουν το JSON μια ιδανική γλώσσα προγραμματισμού ανταλλαγής δεδομένων.

Το JSON προσφέρεται ως εναλλακτική λύση στην XML. Φυσικά δεν μπορεί να αντικαταστήσει την XML 100% διότι δεν υποστηρίζει schema validation, δεν μπορεί από μόνη της να ενημερώσει σχετικά με την κωδικοποίησή της και δεν έχει την έννοια των attributes, αλλά όπου αυτά τα ελαττώματα μπορούν να αγνοηθούν, η αντικατάσταση θα είναι πολύ εύκολη. Όπως και η XML έτσι και το JSON έχει self-documentated format που περιγράφει την δομή των δεδομένων και δεν ασχολείται με την παρουσίασή τους. Επίσης, παραλείπονται πολλές επαναλήψεις κάποιων πληροφοριών που υπάρχουν στην XML.

Παράδειγμα

JSON

```

18 [
19   {id: 1, code: "101", name: "Εισαγωγή στην Πληροφορική", semester: "1"},
20   {id: 2, code: "102", name: "Ψηφιακά Συστήματα", semester: "1" }
21 ]

```

XML

```
3 <courses>
4   <course>
5     <id>1</id>
6     <code>101</code>
7     <name>Εισαγωγή στην Πληροφορική</name>
8     <semester>1</semester>
9   </course>
10  <course>
11    <id>2</id>
12    <code>102</code>
13    <name>Ψηφιακά Συστήματα</name>
14    <semester>1</semester>
15  </course>
16 </courses>
```

Τα μειονεκτήματα της JSON σε σχέση με την XML είναι:

- Δεν υποστηρίζει schema validation
- Δεν μπορεί από μόνη της να ενημερώσει σχετικά με την κωδικοποίησή της
- Δεν έχει την έννοια των attributes.

Με την PHP, για την οποία θα υπάρξει αναφορά αμέσως μετά, δίνεται η δυνατότητα

1. Να μετατρέπει PHP objects και arrays σε JSON με την μέθοδο `json_encode()`.
2. Να δημιουργεί PHP objects και arrays από την JSON με την μέθοδο `json_decode()`.

Επίσης υπάρχει η ίδια δυνατότητα με την JavaScript να γίνει αυτό χρησιμοποιώντας της μεθόδους

1. Την `eval` και την `JSON.parse()` για μετατροπή JSON αντικειμένου σε κείμενο με προτίμηση στην `JSON.parse()` λόγω του ότι παρέχει μεγαλύτερη ασφάλεια. Έχει την ικανότητα να παρέχει μεγαλύτερη ασφάλεια αφού αναγνωρίζει μόνο κείμενο και απορρίπτει τυχόν scripts.
 - a. `var myObject = eval('(' + myJSONtext + ')');`
 - b. `var myObject = JSON.parse(myJSONtext, reviver);`
2. Και την μέθοδο `JSON.stringify` για την αντίστροφη ενέργεια, την μετατροπή απλού κειμένου σε αντικείμενα JSON.
 - a. `var myJSONText = JSON.stringify(myObject, replacer);`

2.3.9 PHP

PHP (ακρωνύμιο του PHP: Hypertext Preprocessor) είναι μια ευρέως χρησιμοποιούμενη, ανοιχτού κώδικα, γενικού σκοπού scripting γλώσσα προγραμματισμού, η οποία είναι ειδικά κατάλληλη για ανάπτυξη ιστοσελίδων και μπορεί να ενσωματωθεί στην HTML.

Μεγάλο μέρος της σύνταξης του είναι δανεισμένο από την C, Java και την Perl με μια σειρά από μοναδικά PHP-ειδικά χαρακτηριστικά.

Ο στόχος της PHP είναι να επιτρέπει στους web developers να δημιουργούν γρήγορα δυναμικά παραγόμενες σελίδες.

Αυτό που διαχωρίζει την PHP από κάτι σαν client-side Javascript είναι ότι, όταν κάποιος επισκέπτεται την ιστοσελίδα, ο web server επεξεργάζεται τον PHP κώδικα. Βλέπει ποια μέρη θα πρέπει να δείξει στους επισκέπτες (το περιεχόμενο και τις εικόνες) και κρύβει όλα τα υπόλοιπα (τη λειτουργία των αρχείων, τους μαθηματικούς υπολογισμούς, κλπ.). Τότε μεταφράζεται ο PHP κώδικας σε μορφή HTML. Μετά την μετάφραση σε HTML, στέλνει την ιστοσελίδα στον browser του επισκέπτη.

Η PHP διαθέτει ενσωματωμένες εντολές υποστήριξης για ένα μεγάλο αριθμό βάσεων δεδομένων (MySQL Microsoft SQL Server, PostgreSQL και Oracle). Μπορεί να μεταφραστεί και να τρέξει σε οποιοδήποτε λειτουργικό σύστημα (Microsoft Windows, Linux, SD, Solaris, Macintosh OS X, και UNIX servers). Συνεργάζεται χωρίς προβλήματα με πολλούς web servers (Apache, IIS) και μπορεί να επικοινωνήσει με αρκετά πρωτόκολλα (HTTP , POP3, SNMP και LDAP) ώστε να παρέχει τη δυνατότητα ανάγνωσης και εγγραφής σε αρχεία και αποστολής email.

Επίσης, παρόλο που η PHP δεν άρχισε ως αντικειμενοστρεφής (object-oriented) γλώσσα προγραμματισμού, μετά από χρόνια προσπάθειας, προστέθηκαν πολλά αντικειμενοστρεφή χαρακτηριστικά που παρουσιάζονταν στις άλλες γλώσσες προγραμματισμού, όπως:

- Ενθυλάκωση
- Κληρονομικότητα – πολλαπλή κληρονομικότητα
- Πολυμορφισμός
- Κλάσεις (Classes)
- Διεπαφές
- Namespaces
- Υπερφόρτωση μεθόδων.

Η PHP επίσης υποστηρίζει όπως και οι άλλες γλώσσες προγραμματισμού λειτουργίες όπως

- Διαχείριση λαθών.
- Εργασία με αρχεία και το λειτουργικό σύστημα.
- Κανονικές εκφράσεις.
- Διαχείριση συνεδριάσεων (session handling).
- Διαδικτυακές υπηρεσίες ... και άλλα.

Ο PHP κώδικας είναι κλεισμένος σε ειδικά tags αρχής `<?php` και τέλους `?>` που επιτρέπουν τη μετάφραση εντός και εκτός PHP λειτουργίας. Αντί για πολλές εντολές για την εξαγωγή HTML κώδικα (όπως στην C ή Perl), οι PHP σελίδες περιέχουν HTML με ενσωματωμένο κώδικα που κάνει «κάτι».

Παράδειγμα

```
<body>
  <b>
    <?php echo "Hello World from PHP script!"; ?>
  </b>
</body>
```

(στην προκειμένη περίπτωση, η έξοδος «**Hello Word from PHP script!**»).

Για να αρχίσει κάποιος να εργάζεται με την PHP θα πρέπει πρώτα να έχει πρόσβαση σε ένα από τα ακόλουθα:

- Ένα λογαριασμό φιλοξενίας ιστοσελίδων (web hosting account) που υποστηρίζει τη χρήση της PHP και του παρέχει πρόσβαση σε βάσεις δεδομένων MySQL.
- Ή να έχει την PHP και MySQL εγκατεστημένη στον υπολογιστή του και να δουλεύει τοπικά την ιστοσελίδα του.

Παρά το γεγονός ότι για την MySQL δεν είναι απολύτως απαραίτητο να χρησιμοποιείται PHP, οι MySQL και PHP δουλεύουν τέλεια μαζί, γι' αυτό και είναι ο συνδυασμός ο οποίος συναντάται περισσότερο σε τέτοιας φύσης συστήματα και ιστοσελίδες και που τον επιλέξαμε για την παρούσα πτυχιακή.

2.3.10 MySql

Η MySQL είναι ένα σύστημα διαχείρισης σχεσιακής βάσης ανοικτού κώδικα (relational database management system - RDBMS) που χρησιμοποιεί την Structured Query Language (SQL), την πιο γνωστή γλώσσα για την προσθήκη, την πρόσβαση και την επεξεργασία δεδομένων σε μία Βάση Δεδομένων.

Λόγω του ότι είναι ανοικτού κώδικα (open source), οποιοσδήποτε μπορεί να κατεβάσει την MySQL και να την διαμορφώσει βάσει των αναγκών του, σύμφωνα πάντα με την γενική άδεια που υπάρχει. Η MySQL είναι γνωστή κυρίως για την ταχύτητα, την αξιοπιστία, και την ευελιξία που παρέχει.

Η MySQL αυτή τη στιγμή μπορεί να λειτουργήσει σε περιβάλλον Linux, Unix, και Windows και χρησιμοποιείται από τα πιο μεγάλα CMS (Joomla, Wordpress, Drupal κλπ) και όπως και από τεράστια συστήματα που χρησιμοποιούνται καθημερινά (Google, Wikipedia, Facebook, Twitter.).

2.3.11 Διαδικτυακές υπηρεσίες

Ο σύγχρονες ιστοσελίδες σπάνια λειτουργούν μεμονωμένα, αντ' αυτού συχνά βασίζονται στα δεδομένα, τις αποθηκευτικές ικανότητες, ακόμη και στην υπολογιστική δύναμη άλλων υπηρεσιών του διαδικτύου, προκειμένου να δημιουργηθούν νέες και μοναδικές υπηρεσίες.

Ιστοσελίδες όπως η λειτουργία Jog Walk (www.walkjogrun.net) και Wozor (www.wozor.com) είναι παραδείγματα αυτής της πρακτικής, που στηρίζονται κυρίως στο API της Google Maps και άλλων τρίτων πηγών δεδομένων για να παράγουν συναρπαστικά online εργαλεία. Εκατομμύρια άλλες ιστοσελίδες εξαρτώνται και αυτές από τρίτους.

Μερικές φορές αυτή η εξάρτηση δεν είναι πάντα τόσο προφανής (εμφανής?), όταν για παράδειγμα κάποια sites χρησιμοποιούν CloudFront, υπηρεσία της Amazon που φιλοξενεί αντίγραφα ιστοσελίδων (website images) και άλλα στατικά αρχεία. Επίσης, άλλοι χρησιμοποιούν αυτές τις υπηρεσίες τρίτων, για τον υπολογισμό εξόδων αποστολής (www.usps.com / webtools) και την εκπλήρωση παραγγελιών (Amazon FWS στο <http://aws.amazon.com/fws>).

Αυτές οι λύσεις, τρίτων κατασκευαστών, ονομάζονται *διαδικτυακές υπηρεσίες*.

Χαρακτηριστικά Διαδικτυακών υπηρεσιών

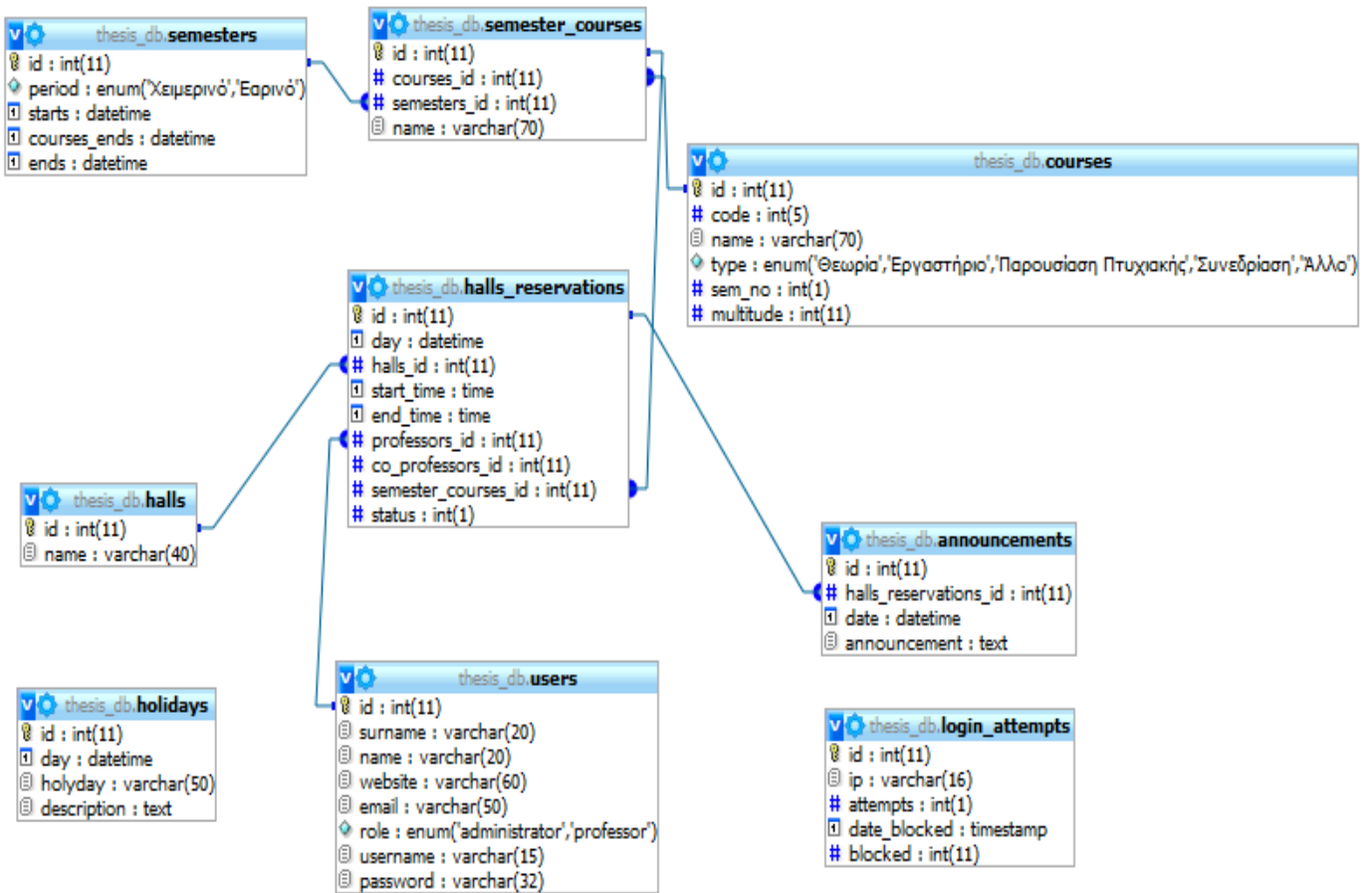
- Είναι κομμάτια εφαρμογής.
- Επικοινωνούν χρησιμοποιώντας ανοικτά πρωτόκολλα
- Είναι αυτόνομες και αυτο-περιγραφικές
- Μπορούν να ανακαλυφθούν με UDDI
- Μπορούν να χρησιμοποιηθούν από άλλες εφαρμογές.
- Το XML είναι η βάση για τις διαδικτυακές υπηρεσίες.
-

Κεφάλαιο 3 Σχεδιασμός

3.1 Εισαγωγή

Σε αυτό το κεφάλαιο παρουσιάζεται ένα διάγραμμα σχέσης οντοτήτων (ER – Entity relationship) το οποίο μπορεί να το εξαχθεί από το εργαλείο *phrmyadmin* και στη συνέχεια αναλύονται περιληπτικά οι πίνακες της βάσης και ποια στοιχεία θα κρατάει ο καθένας.

3.2 ER Diagram



3.3 Ανάλυση βάσης δεδομένων

Αρχικά θα παρατεθεί η περιγραφή της βάσης αναλύοντας τους σχετικά απλούς στην ανάλυση τους πίνακες και αφήνοντας για το τέλος κάποιους που κρατάνε όλο το ημερολόγιο και ίσως να είναι λίγο πιο περίπλοκοι.

3.3.1 Semesters

Ο πίνακας αυτός κρατάει τα εξάμηνα και τις ημερομηνίες τους.

Τα πεδία που έχει είναι:

- **Id:** τύπου integer και auto increment, δηλαδή με κάθε εισαγωγή αυξάνεται από μόνο του χωρίς να χρειάζεται να δίνεται τιμή κάθε φορά από τον χρήστη.
- **period:** τύπου enum που έχει δύο συγκεκριμένες τιμές τις οποίες μπορεί να δεχθεί (χειμερινό, εαρινό).
- **starts:** τύπου datetime που θα κρατά την ημερομηνία την οποία ξεκινά το εξάμηνο και οι παρακολουθήσεις μαθημάτων.
- **courses_ends:** τύπου datetime που θα κρατά την ημερομηνία την οποία καθορίζει το πότε τελειώνει η διεξαγωγή μαθημάτων και κατ'επέκταση η αρχή της εξεταστικής περιόδου.

- **ends:** τύπου `datetime` που θα κρατά την ημερομηνία την οποία καθορίζει πότε τελειώνει η εξεταστική περίοδος και ουσιαστικά το τέλος του εξαμήνου.

3.3.2 Halls

Στον πίνακα αυτό θα καταχωρηθούν οι διάφορες αίθουσες τις οποίες θα υπάρχει η δυνατότητα δέσμευσης τους.

Τα πεδία που έχει είναι:

- **Id:** τύπου `integer` και `auto increment`.
- **Name:** τύπου `varchar(40)` αλφαριθμητικό μεγίστου μεγέθους 40 που θα κρατά το όνομα της αίθουσας (101, 311, αμφιθέατρο,...).

3.3.3 Users

Σε αυτό το πίνακα θα υπάρχουν τα στοιχεία των χρηστών του συστήματος.

Τα πεδία που έχει είναι:

- **Id:** τύπου `integer` και `auto increment`.
- **Surname:** τύπου `varchar (20)` αλφαριθμητικό μεγίστου μεγέθους 20 που θα κρατά το επίθετο του χρήστη.
- **Name:** τύπου `varchar (20)` αλφαριθμητικό μεγίστου μεγέθους 20 που θα κρατά το όνομα του χρήστη.
- **Website:** τύπου `varchar (60)` αλφαριθμητικό μεγίστου μεγέθους 60 που θα κρατά URL για το website του χρήστη.
- **Email:** τύπου `varchar (50)` αλφαριθμητικό μεγίστου μεγέθους 50 που θα κρατά το email του χρήστη.
- **Role:** τύπου `enum` που θα παίρνει μία από τις δύο τιμές που καθορίζονται (`administrator`, `professor`). Αυτό το πεδίο θα χρησιμοποιείται για να δίνει κάποια δικαιώματα στους `professors` (καθηγητές) και όλα στους `administrators` (διαχειριστές).
- **Username:** τύπου `varchar (15)` αλφαριθμητικό μεγίστου μεγέθους 15 που θα κρατά το αναγνωριστικό (`username`) του χρήστη.
- **Password:** τύπου `varchar (32)` αλφαριθμητικό μεγίστου μεγέθους 32 που θα κρατά το κωδικό του χρήστη.

3.3.4 Holidays

Αυτός ο πίνακας θα κρατάει τις μέρες τις οποίες μπορεί να δηλώσει ο διαχειριστής, στις οποίες μέρες μπορούν να ή δεν μπορούν να διεξαχθούν μαθήματα, όπως σαββατοκύριακα και αργίες. Όμως επειδή η πείρα δείχνει ότι κάποιες πρέπει να διεξαχθούν και σε κάποιες

αργίες ή Σάββατα, υπάρχει το κατάλληλο πεδίο το οποίο μπορεί του δοθεί τιμή 1 ή 0 ανάλογα με τις ανάγκες μας.

Τα πεδία που έχει είναι:

- **Id:** τύπου **integer** και auto increment.
- **Day:** τύπου **datetime** που θα κρατά την ημερομηνία η οποία θα δηλωθεί σαν αργία.
- **Holiday:** τύπου **varchar (50)** που θα κρατά μονολεκτικά το λόγο που η συγκεκριμένη μέρα είναι αργία.
- **Description:** τύπου **text** όπου θα μπορεί να καταχωρηθεί μια περιγραφή για τον λόγο που είναι δηλωμένη σαν αργία ή για το λόγο που στη συγκεκριμένη δεν θα διεξαχθούν ή θα διεξαχθούν μαθήματα.
- **Canbereserved:** τύπου **varchar (5)** που θα παίρνει τιμές **true/false** και θα καθορίζει εάν μπορούν ή όχι να γίνουν δεσμεύσεις την συγκεκριμένη μέρα.

3.3.5 Courses

Ο πίνακας αυτός κρατάει τα μαθήματα και γενικά τους «λόγους» που μπορούν να γίνονται οι δεσμεύσεις των αιθουσών, όπως θεωρίες, εργαστήρια, παρουσιάσεις κλπ.

Τα πεδία που έχει είναι:

- **Id:** τύπου **integer** και auto increment, δηλαδή με κάθε εισαγωγή αυξάνεται από μόνο του χωρίς να χρειάζεται να δίνει ο φοιτητής/ επισκέπτης τιμή κάθε φορά
- **Code:** τύπου **integer** που θα κρατάει τον κωδικό αριθμό
- **Type:** τύπου **enum** και ουσιαστικά θα κρατά τα είδη των δεσμεύσεων (θεωρία, εργαστήριο, παρουσίαση, συνεδρίαση, άλλο)
- **Name:** τύπου **varchar (70)** που σημαίνει αλφαριθμητικό μεγίστου μεγέθους 70 χαρακτήρων που θα είναι το όνομα για το μάθημα, «Συνεδρίαση» για τις συνεδριάσεις, «Παρουσίαση πτυχιακής» για τις παρουσιάσεις κλπ.
- **Sem_no:** τύπου **integer(1)** ακέραιος μεγέθους 1 που θα είναι ο αριθμός του εξαμήνου διεξαγωγής του μαθήματος/παρουσίασης κλπ.
- **Multitude:** τύπου **integer** το οποίο θα είναι για όλα εξορισμού (by default) 1 εκτός από τα εργαστήρια, όπου θα κρατάει το πλήθος των εργαστηρίων βάσει του οποίου θα δημιουργεί το σύστημα τα κατάλληλα «Εργαστήριο T1», «Εργαστήριο T2» που θα αποθηκεύονται στον **semester_courses**.

3.3.6 Semester courses

Σε αυτό τον πίνακα θα αποθηκεύεται κάθε διακριτό εργαστήριο, θεωρία κλπ που θα έχουν οι φοιτητές το εκάστοτε τρέχον εξάμηνο. Πιο συγκεκριμένα, μετά από την εισαγωγή στον **course** και από την επεξεργασία των δεδομένων που εισάχθηκαν, θα δημιουργούνται οι κατάλληλες εισαγωγές για αυτό το πίνακα.

Τα πεδία που έχει είναι:

- **Id:** τύπου **integer** και auto increment, δηλαδή με κάθε εισαγωγή αυξάνεται από μόνο του χωρίς να χρειάζεται να δίνει ο φοιτητής/ επισκέπτης τιμή κάθε φορά.
- **Courses_id:** τύπου **integer** και θα κρατά την τιμή της εγγραφής **course** στην οποία αναφέρεται ο φοιτητής/ επισκέπτης.
- **Semester_id:** τύπου **integer** και θα κρατά την τιμή του **semester** στο οποίο αναφέρεται ο φοιτητής/ επισκέπτης.
- **Name:** τύπου **varchar (70)** που σημαίνει αλφαριθμητικό μεγίστου μεγέθους 70 χαρακτήρων, που θα είναι το όνομα πχ. «Εισαγωγή στην πληροφορική Τ4».

3.3.7 login attempts

Σε αυτό το πίνακα θα καταχωρείται κάθε καινούρια IP διεύθυνση που επισκέπτεται το ημερολόγιο του φοιτητή/ επισκέπτη, μαζί με ένα counter, ο οποίος θα αυξάνεται μετά από κάθε άκυρη προσπάθεια login έτσι ώστε, μετά από ένα προκαθορισμένο αριθμό προσπαθειών, να μπλοκάρεται με αποτέλεσμα να μην μπορεί για κάποιο χρονικό διάστημα ο χρήστης να προσπαθήσει ξανα.

Τα πεδία που έχει είναι:

- **Id:** τύπου **integer** και auto increment, δηλαδή με κάθε εισαγωγή αυξάνεται από μόνο του χωρίς να χρειάζεται να δίνει ο φοιτητής/επισκέπτης τιμή κάθε φορά
- **IP:** τύπου **varchar(16)** που θα κρατάει την IP που θέλει ο φοιτητής/ επισκέπτης να ελέγξει
- **Date_blocked:** τύπου **datetime** που θα κρατά την ημέρα και ώρα της πρώτης προσπάθειας login.
- **Attempts:** τύπου **integer** και ορίζει το πλήθος αποτυχημένων προσπαθειών
- **Blocked:** τύπου **integer** και ορίζει με τιμή «1» ότι ο επισκέπτης έχει ήδη μπλοκαριστεί και με «0» ότι ακόμη δεν έχει μπλοκαριστεί.

3.3.8 halls reservations

Αυτός ο πίνακας είναι και ο πιο κρίσιμος, καθότι εδώ αποθηκεύονται όλα τα δεδομένα για τις τελικές δεσμεύσεις που έχουν γίνει καθώς και από ποιον καθηγητή, μάθημα, αίθουσα, ώρα, βοηθό καθηγητή...

Τα πεδία που έχει είναι:

- **Id:** τύπου **integer** και auto increment, δηλαδή με κάθε εισαγωγή αυξάνεται από μόνο του χωρίς να χρειάζεται να δίνει ο φοιτητής/επισκέπτης τιμή κάθε φορά.
- **Day:** τύπου **date** που θα κρατά την ημέρα της δέσμευσης.
- **Halls_id:** τύπου **integer** και καθορίζει το **id** της αίθουσας για την οποία αναφέρεται η συγκεκριμένη δέσμευση.
- **Start_time – End_time:** τύπου **time** που καθορίζουν τις ώρες αρχής και τέλους.
- **Professors_id:** τύπου **integer** που καθορίζει το **id** του καθηγητή ο οποίος κάνει την κράτηση.

- **Co_professors_id:** τύπου **integer** που καθορίζει το **id** του βοηθού καθηγητή για την κράτηση.
- **Semester_courses_id:** τύπου **integer** που καθορίζει το **id** του λόγου της κράτησης από τον πίνακα που αναλύθηκε πιο πάνω.

3.3.9 announcements

Σε αυτό το τελευταίο πίνακα μπαίνουν οι ανακοινώσεις και/ή κάποια περιγραφή που θα βάζει ο χρήστης για την κράτηση αίθουσας που κάνει.

Τα πεδία που έχει είναι:

- **Id:** τύπου **integer** και auto increment, δηλαδή με κάθε εισαγωγή αυξάνεται από μόνο του χωρίς να χρειάζεται να δίνει ο φοιτητής/επισκέπτης τιμή κάθε φορά.
- **Halls_reservation_id:** τύπου **integer** και είναι το **id** της εγγραφής από το πίνακα πιο πάνω που έχει τα στοιχεία της κράτησης.
- **Date:** τύπου **datetime** το οποίο θα κρατά την ακριβή ημέρα και ώρα που έγινε η εγγραφή ή η τελευταία τροποποίηση της.
- **Announcement:** τύπου **text** το οποίο θα είναι το κείμενο-περιεχόμενο της ανακοίνωσης.

Κεφάλαιο 4 Αρχιτεκτονική και Αλγόριθμοι [22]

4.1 Εισαγωγή

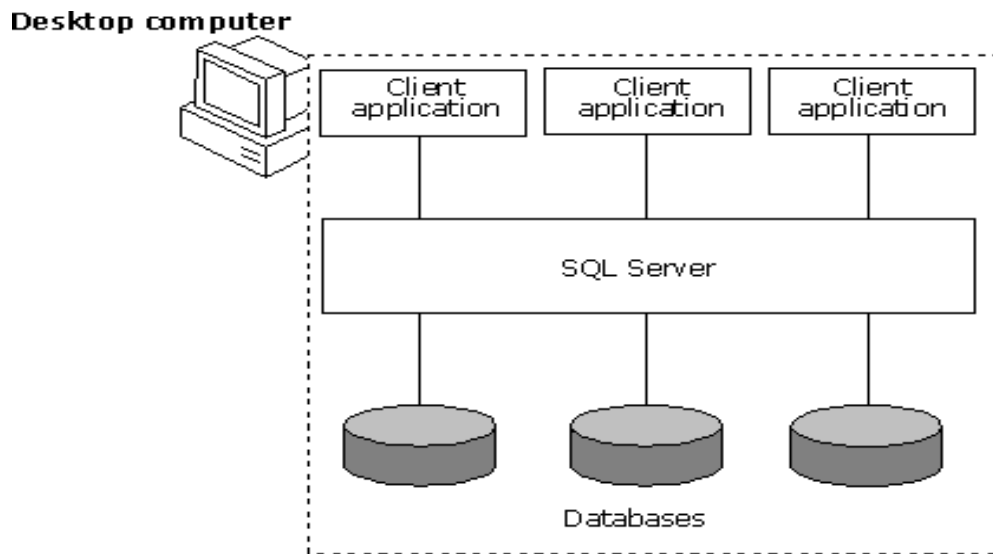
Σε αυτό το κεφάλαιο θα παρουσιαστούν κάποια χαρακτηριστικά και αρχιτεκτονικές που χαρακτηρίζουν τις αρχιτεκτονικές των περισσότερων web applications και επιπλέον αυτών που κάνουν χρήση της τεχνολογίας AJAX. Θα υπάρξει μια μικρή περιγραφή του τρόπου με τον οποίο δουλεύει το σύστημα και έπειτα θα ακολουθήσουν κάποιες τεχνικές λεπτομέρειες που αφορούν το θέμα της ασφάλειας και κάποιες επιπλέον τεχνικές που χρησιμοποιήθηκαν σε αυτή την πτυχιακή εργασία. Επίσης, θα εξετασθούν και θα αναλυθούν τέσσερα κύρια παραδείγματα με επεξήγηση αλγόριθμου.

4.2 Αρχιτεκτονική server-client 2 επιπέδων [23] [24].

Η εφαρμογή χωρίζεται σε δύο μέρη. Την τοποθεσία διαχείρισης και την τοποθεσία παρουσίασης των δεδομένων. Οι δύο αυτές τοποθεσίες έχουν επίσης φυσικό διαχωρισμό όσον αφορά την οργάνωση τους σε καταλόγους. Σε αυτές τις τοποθεσίες βρίσκεται όλο το client-side μέρος της εφαρμογής. Υπάρχει φυσικά και ένα τρίτο μέρος όπου είναι καθαρά το server-side κομμάτι του κώδικα, με το οποίο η εφαρμογή επικοινωνεί καθαρά με την τεχνολογία AJAX, αφού προηγηθεί η ενέργεια της σύνδεσης του χρήστη για να έχουν πλέον καθοριστεί οι ενέργειες και οι σελίδες που μπορεί να έχει πρόσβαση.

Η αρχιτεκτονική πελάτη-διακομιστή υλοποιείται από δύο κόμβους-συσκευές: α) του πελάτη ο οποίος μπορεί να είναι ένας χρήστης ή μια διεργασία στο διαδίκτυο ο οποίος

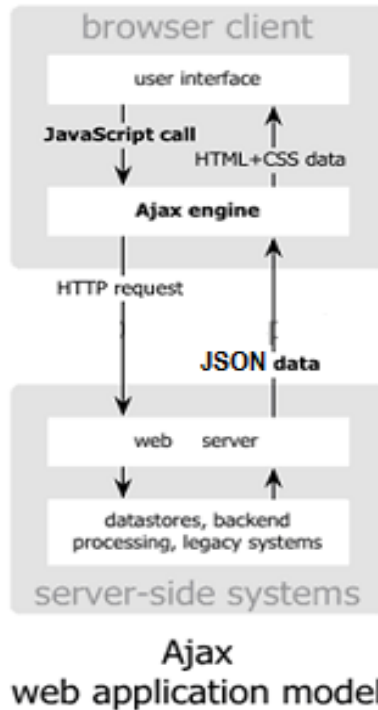
μέσω του περιηγητή αποστέλλει και λαμβάνει δεδομένα από το διακομιστή και β) του διακομιστή ο οποίος ενσωματώνει τις υπηρεσίες εξυπηρέτησης ιστού, βάσης δεδομένων, εκτυπωτών, κλπ..



4.3 Αρχιτεκτονική AJAX

Στην παρακάτω εικόνα βλέπουμε πώς δουλεύουν οι εφαρμογές που χρησιμοποιούν την τεχνολογία AJAX και ακολούθως θα παρουσιαστεί ένα παράδειγμα για να γίνει πιο καλά αντιληπτός ο τρόπος με τον οποίο λειτουργεί το σύστημα αυτό.

Όπως φαίνεται, στην πλευρά του χρήστη (browser) δίνεται εντολή για μια λειτουργία μέσω javascript για να έρθουν δεδομένα για το γέμισμα του ημερολογίου ή για την αποθήκευση κάποιων καινούριων δεδομένων ή για οποιονδήποτε λόγο. Εδώ γίνεται μια AJAX κλήση που ξεκινά από τον browser και απευθύνεται σε ένα php script που βρίσκεται στον server. Το php script τρέχει, επικοινωνεί με την βάση, παίρνει τα δεδομένα, τα κωδικοποιεί στο κατάλληλο format και τα επιστρέφει στην μέθοδο από την οποία κλήθηκε, για να συνεχίσει αποκωδικοποιώντας τα δεδομένα ούτως ώστε να μπορεί να τα χειριστεί η javascript.



4.4 Παράδειγμα 1 – Γέμισμα drop down στοιχείου

Σε αυτό το παράδειγμα είναι επιθυμητό κατά την φόρτωση μιας σελίδας, να συμπληρώνονται κάποια drop down πεδία δυναμικά οι τιμές των οποίων δεν είναι δηλωμένες στατικά, αλλά προέρχονται από ένα αίτημα στον διακομιστή.

Δημιουργία του HTML στοιχείου θέτοντας ιδιότητες **class** και **id** το οποίο **class** θα χρησιμοποιηθεί για το styling του στοιχείου και το **id** για να μπορεί ο χρήστης μέσω javascript-jQuery να έχει πρόσβαση σε αυτό και να του προσδώσει τις τιμές.

Ο αντίστοιχος κώδικας για την σχεδίαση του στοιχείου είναι

```
<select name="professors" class="professors_db" id="professors" >  
  <option>Select Professors</option>  
</select>
```

Έπειτα παρατίθεται ο javascript-jQuery κώδικας ο οποίος θα κάνει το Ajax call και θα πάρει ως απάντηση από τον server ένα JSON αντικείμενο, από το οποίο θα εξάγει τα δεδομένα, και θα καλέσει μια άλλη μέθοδο, που η οποία θα τα μετατρέψει σε επιλογές(<option>) και θα τις τοποθετήσει στο drop down.

Ο κώδικας για αυτές τις δύο μεθόδους είναι

Πτυχιακή Εργασία του Χαραλάμπους Μάριου

```
/**
 * Returns all Professors in a drop down list
 * so users can select
 */
$(document).ready( function () {
    $.ajax({
        type : "POST",
        url : 'getByProfessor.php',
        data : 'getProfessors=yes',
        success:(function(data) {
            var json_data_object = eval("(" + data + ")");
            var options = '<option>Select Professor</option>';
            traverseProfessors(json_data_object, options);
        }),
        error: (function() { alert("error"); }),
        complete: (function() { /*alert("Completed");*/ })
    });
});

/**
 * Traverse the JSON obj creates all options
 * for the Professors drop down
 */
function traverseProfessors(obj, options) {
    var type = typeof obj ;
    if (type == "object") {
        for (var key in obj) {
            options+="
```

Με αυτό επιτυγχάνεται η συμπλήρωση? Όλων των στοιχείων που χρειάζονται σε μια HTML σελίδα με τα πιο ενημερωμένα δεδομένα, αυτόματα κάθε φορά, χωρίς να χρειάζεται καν να ξαναφορτωθεί η σελίδα, μένοντας έτσι σε μια μόνο σελίδα.

Στην εικόνα που ακολουθεί βλέπουμε το αποτέλεσμα

Course Calendar

<< Previous Week Select Week

Select Hall OR Select Professor OR Select Course

Monday	Tuesday	Wednesday	Thursday	Friday

Select Professor
Σιδηρόπουλος Αντώνης
Ηλιούδης Χρίστος
Ιγνάτιος Δελιγιάννης
Χατζημίσιος Περικλής
Αντωνίου Σταύρος
Ψαράς Νικόλας
Χαραλάμπους Μάριος
Ρήγας Βασίλης
mmvirtual Marios

Με τον ίδιο τρόπο συμπληρώνονται όλα τα στοιχεία των διάφορων φορμών (τα οποία θα αναφερθούν στο επόμενο κεφάλαιο) που υπάρχουν στο σύστημα με τα πιο ενημερωμένα δεδομένα αφού αυτά συλλέγονται την ίδια τη στιγμή της επίσκεψης στην ιστοσελίδα.

Βάσει αυτού του τρόπου και της αρχιτεκτονικής δουλεύει όλο το σύστημα, σαν να υπάρχει κάπου εκεί πίσω μια υπηρεσία, η οποία, στέλνοντας της τα κατάλληλα ερωτήματα απαντά

με τα κατάλληλα δεδομένα τα οποία ο χρήστης μπορεί όχι μόνο να τα επεξεργαστεί με αυτό τον τρόπο αλλά ακόμη θα μπορούσε να τα καταχωρήσει σε δική του βάση, να δημιουργήσει άλλα στατικά αρχεία και ό,τι άλλο θα θέλει.

4.5 Παράδειγμα 2 – Καταχώρηση εγγραφής.

Σε αυτό το παράδειγμα θα εξετασθεί η περίπτωση κατά την οποία ένας καθηγητής ή ο διαχειριστής του συστήματος θέλει να κάνει μια καινούρια δέσμευση αίθουσας για μια συγκεκριμένη ημέρα. Δεν θα αναφερθεί σε αυτό το σημείο στο πώς μπορεί κάποιος να μεταφερθεί στη σελίδα και να επιλέξει την ημερομηνία που επιθυμεί, αφού αυτό το κομμάτι αποτελεί μέρος του επόμενου κεφαλαίου.

Όταν ο χρήστης λοιπόν εισάγει το στοιχεία στην κατάλληλη φόρμα και πατήσει το κουμπί “save”, το client side μέρος της εφαρμογής καλεί μια μέθοδο που πρώτα θα ελέγξει εάν όλα τα απαραίτητα πεδία είναι συμπληρωμένα και σε περίπτωση που υπάρχει κάποιο κενό, αλλάζει το χρώμα του background του κάθε μη συμπληρωμένου στοιχείου σε κόκκινο. Σε αντίθετη περίπτωση, ελέγχει τα δεδομένα εάν είναι του τύπου που πρέπει να είναι, δηλαδή, όπου χρειάζεται ημερομηνία να είναι ημερομηνία με το κατάλληλο format, όπου χρειάζεται ακέραιος να είναι ακέραιος και όπου αλφαριθμητικό να είναι αλφαριθμητικό. Αφού όλα επιβεβαιωθούν πως είναι σωστά, τότε, μπορεί να προχωρήσει το AJAX κομμάτι της εφαρμογής στην αποστολή του POST με τα δεδομένα στο κατάλληλο κάθε φορά script, το οποίο σε αυτή τη περίπτωση θα απαντήσει με ένα κατάλληλο μήνυμα σωστής εισαγωγής ή λάθους, το οποίο ήταν η αιτία να μην μπορέσει να γίνει η καταχώρηση.

Ο javascript κώδικας που τρέχει για το validation των τιμών των στοιχείων

```

*/
function validateAddForm(){
    //Validate the dropdowns
    if($("#select#id_add_fillForm_professor option:selected").val()=='Select Professor'){
        $("#select#id_add_fillForm_professor").css('background-color', 'red');
    } else {
        $("#select#id_add_fillForm_professor").css('background-color', 'white');
    }
    //Validate the 3 dropdowns
    if($("#select#id_add_fillForm_co_professor option:selected").val()=='Select Professor'){
        $("#select#id_add_fillForm_co_professor").css('background-color', 'red');
    } else {
        $("#select#id_add_fillForm_co_professor").css('background-color', 'white');
    }
    if($("#select#id_add_fillForm_course option:selected").val()=='Select Course'){
        $("#select#id_add_fillForm_course").css('background-color', 'red');
    } else {
        $("#select#id_add_fillForm_course").css('background-color', 'white');
    }
    if($("#select#id_add_fillForm_hall option:selected").val()=='Select Hall'){
        $("#select#id_add_fillForm_hall").css('background-color', 'red');
    } else {
        $("#select#id_add_fillForm_hall").css('background-color', 'white');
    }
    if($("#select#id_add_fillForm_professor option:selected").val()=='Select Professor' ||
        $("#select#id_add_fillForm_course option:selected").val()=='Select Course' ||
        $("#select#id_add_fillForm_hall option:selected").val()=='Select Hall'){
        return false;
    } else {
        return true;
    }
}

```

Ο javascript κώδικας που τρέχει για την αποστολή των δεδομένων:

```

* Fires when the img-button add id clicked and tries to add
* new reservation.
* Returns the proper msg.
*/
function addReservation(){
    if(!validateAddForm()){
        return false;
    }
    // serialize(): serializes all form data ready to be POSTed.
    var dataToAdd = $('#add_fillFromForm').serialize();
    //Show msg to inform that the system will try to add many entries.
    if($("#input[@name=addAllRecords]:checked").val()=="yes"){
        //Alert to confirm it also
        if(!confirm("The system will try to insert entries until the " +
            "end of the semester. Are you sure?")){
            alert("This action aborted by the user.");
            return false;
        }
    }
    //Add to the serialized data the action
    dataToAdd = dataToAdd+"&action="+"add";
    $.ajax({
        type : "POST",
        url : 'calendarCrud.php',
        data : dataToAdd,
        success:( function(data){ alert(data);}),
        error: (function(err) { alert("error: "+err); }),
        complete: (function() {
            rebuildCalendar(); // If added, rebuild the calendar
            // And remove the fillFormDiv
            $('#fillFormDiv').hide('explode', {}, 'slow');
        })
    });
}

```

4.6 Παράδειγμα 3 – Ενημέρωση εγγραφής.

Σε αυτό το παράδειγμα θα εξετασθεί η περίπτωση κατά την οποία ένας καθηγητής ή ο διαχειριστής του συστήματος θέλει να κάνει μια ενημέρωση μιας δέσμευση αίθουσας. Και πάλι όπως με την εισαγωγή δέσμευσης, με τη συμπλήρωση της κατάλληλης φόρμας τρέχει μια μέθοδος, η οποία ελέγχει και επιβεβαιώνει τα εισαχθέντα στοιχεία καταπόσο δεν υπάρχουν κενά και ότι είναι του τύπου που πρέπει. Αφού τελειώσει, τρέχει ο AJAX κώδικας που θα προκαλέσει ένα POST αίτημα, το οποίο θα μεταφέρει τα δεδομένα σε ένα άλλο php script. Αυτό το script θα ελέγξει και πάλι τον τύπο των δεδομένων με περισσότερη ασφάλεια, και εφόσον εκτελείται στο server και αφού όλα είναι εντάξει, θα προχωρήσει με την ενημέρωση.

Για αυτό και στο επόμενο παράδειγμα θα παρουσιαστεί ο php κώδικας (ο οποίος θα επεξηγείται αμέσως πιο κάτω) αφού ο javascript – AJAX είναι σχεδόν ίδιος με του προηγούμενου παραδείγματος.

Ο php κώδικας για την επαλήθευση και ενημέρωση των δεδομένων

Πτυχιακή Εργασία του Χαραλάμπους Μάριου

```
165 function update($link, $log){
166     // Get the posted data
167     $id_fillForm_professor = (isset($_POST['id_fillForm_professor'])) ? trim($_POST['id_fillForm_professor']) : n
168     $id_fillForm_co_professor = (isset($_POST['id_fillForm_co_professor'])) ? trim($_POST['id_fillForm_co_profess
169     $id_fillForm_course = (isset($_POST['id_fillForm_course'])) ? trim($_POST['id_fillForm_course']) : null;
170     $id_fillForm_hall = (isset($_POST['id_fillForm_hall'])) ? trim($_POST['id_fillForm_hall']) : null;
171     $id_fillForm_date = (isset($_POST['id_fillForm_date'])) ? trim($_POST['id_fillForm_date']) : null;
172     $id_fillForm_start_time = (isset($_POST['id_fillForm_start_time'])) ? trim($_POST['id_fillForm_start_time'])
173     $id_fillForm_end_time = (isset($_POST['id_fillForm_end_time'])) ? trim($_POST['id_fillForm_end_time']) : null
174     $updateAllRecords = (isset($_POST['updateAllRecords'])) ? trim($_POST['updateAllRecords']) : null;
175     $id_fillForm_hall_reservation = (isset($_POST['id_fillForm_hall_reservation'])) ? trim($_POST['id_fillForm_ha
176     $hallReservationId = (isset($_POST['hallReservationId'])) ? trim($_POST['hallReservationId']) : null;
177     $fillForm_announcement = (isset($_POST['fillForm_announcement'])) ? trim($_POST['fillForm_announcement']) : null
178     //Validate the date
179     if(!validateDates($id_fillForm_date)){
180         echo "Not valid date.";
181         exit;
182     }
183     // Check if someone is connected properly
184     if(isset($_SESSION['user_id'])){
185         // Build sql to check if its holiday
186         $sql_isHoliday = "SELECT holidays.holiday
187                         FROM holidays
188                         WHERE holidays.day = '._insert($id_fillForm_date).'"
189                         AND holidays.canbereserved='false' ";
190         $result_isHoliday = mysql_query($sql_isHoliday, $link);

190     $result_isHoliday = mysql_query($sql_isHoliday, $link);
191     if($result_isHoliday){
192         $r = mysql_fetch_row($result_isHoliday);
193         if(isset($r) && !empty($r)){
194             echo "There won't be any courses on this day (._insert($id_fillForm_date).)";
195             echo "\nReason: ".$r[0]; exit;
196         } else { //do nothing. "You can add.";
197         }
198     } else {
199         $log->lwrite("update(). Something went wrong while updating. Error: ". mysql_error());
200         echo "Something went wrong. Please try again."; exit;
201     }
202     //Check if there is no other reservation with this data.
203     $updateR = sqlInsertUpdateCheck($id_fillForm_date, $id_fillForm_hall, $id_fillForm_co_professor,
204     $id_fillForm_professor, $id_fillForm_course, $id_fillForm_start_time, $id_fillForm_end_time, $link);
205     if(is_array($updateR) && $updateR[0]>1){
206         echo "More than one rows exists on this time/date/hall/professor/course."; exit;
207     } elseif($updateR==-1) {
208         // "You can make the update.";
209     } elseif($updateR[1]!=$id_fillForm_hall_reservation){
210         echo "<br /> One row exists but not with the same id.. "; exit;
211     }
212     // Build sql
213     $mysqli = new mysqli(DB_HOST, DB_USERNAME, DB_PASSWORD, DB_DATABASE);
214     $sql_update = "UPDATE halls_reservations
215                 SET day='._insert($id_fillForm_date).', halls_id='._insert($id_fillForm_hall).',
216                     start_time='._insert($id_fillForm_start_time).',
217                     end_time='._insert($id_fillForm_end_time).',
218                     semester_courses_id='._insert($id_fillForm_course).',
219                     professors_id='._insert($id_fillForm_professor).',
220                     co_professors_id='._insert($id_fillForm_co_professor).'"
221     WHERE id='._insert($id_fillForm_hall_reservation).' ";
222 }
```

Πτυχιακή Εργασία του Χαραλάμπους Μάριου

```
223 // if its professor, confirm that its his reservation
224 if($_SESSION['user_role']=='professor'){
225     $sql_update .= " AND professors_id="._insert($_SESSION['user_id'])." ";
226 } elseif($_SESSION['user_role']=='administrator'){
227
228     // If the user is logged in as administrator then update only by the id of the record.
229 } else {
230     echo "<h2> You have to <a href='./login/index.php'> login </a> first. </h2>"; exit;
231 }
232 //build the sql
233 $sql_update .= "; UPDATE announcements
234     SET announcement="._insert($fillForm_announcement)."
235     WHERE halls_reservations_id="._insert($id_fillForm_hall_reservation)."; ";
236 // Execute the query
237 $result_update = mysqli_multi_query($mysqli, $sql_update);
238 if(!$result_update){
239     echo 'Could not update data. Please try again. ';
240     $log->lwrite("Could not update.");
241 } else {
242     $sql_confirm = " SELECT id FROM halls_reservations
243         WHERE day="._insert($id_fillForm_date)."
244         AND halls_id="._insert($id_fillForm_hall)."
245         AND start_time="._insert($id_fillForm_start_time)." AND end_time="._insert
246 $resultConfirm = mysql_query($sql_confirm, $link);
247 if(!$resultConfirm){
248     echo "Something went wrong while updating";
249     $log->lwrite("update(). Something went wrong while updating.");
250 } else {
251     $log->lwrite("update(). Updated successfully.");
252     echo "Updated successfully";
253 }
254 }
255 } else {
256     $log->lwrite("update(). Anuthorized user try to update.");
257     echo "You are not connected so you can't perform this action."; }
258 }//end
```

```
/**
 * Checks if the param its a date..
 * @param date $date
 */
if (!function_exists('validateDates')) {
    function validateDates($date) {
        // Makes sure things like 1998-13-32 won't get past and validate.
        if ( preg_match('/^[0-9]{4}-(0*[1-9]|1[0-2])-(0[1-9]|[1-2][0-9]|3[0-1])$/ ', $date) ) {
            list($year , $month , $day) = explode('-', $date);
            return checkdate($month , $day , $year);
        } else {
            echo "Date's format invalid.<br /> ";
        }
    }
}
```

```
323 function sqlInsert_UpdateCheck($date, $hall, $co_professor, $professor, $course, $start_time, $end_time, $link){
324     $nst_p = date("H:i:s", strtotime($start_time)+(1*60));
325     $net_s = date("H:i:s", strtotime($end_time)-(1*60));
326     $sql_check = "SELECT count(id) as `count`, id as `id`
327         FROM halls_reservations
328         WHERE halls_reservations.day = "._insert($date)."
329         AND (
330             halls_reservations.halls_id="._insert($hall)." OR
331             halls_reservations.professors_id="._insert($professor)." OR
332             halls_reservations.professors_id="._insert($co_professor)." OR
333             halls_reservations.semester_courses_id="._insert($course)." OR
334             halls_reservations.co_professors_id="._insert($professor)." OR
335             halls_reservations.co_professors_id="._insert($co_professor)." )
336         AND (
337             "._insert($start_time) ." < halls_reservations.start_time
338             AND "._insert($end_time) ." > halls_reservations.end_time OR
339             "._insert($nst_p) ." between halls_reservations.start_time AND halls_reservations.end_time OR
340             "._insert($net_s) ." between halls_reservations.start_time AND halls_reservations.end_time
341         )";
342     $result_check = mysql_query($sql_check, $link);
343     if($result_check){
344         $r = mysql_fetch_row($result_check);
345         if($r[0]>=1){ return $r; } else { return -1; }
346     } else {
347         echo "Something went wrong. Please ty again.";
348         echo 'MySQL Error: ' . mysql_error();
349     }
350 }
351 }
```

Στις γραμμές 167-177 μπαίνουν οι τιμές που έχουν σταλεί με το αίτημα POST από τον browser στο server, σε τοπικές μεταβλητές. Αμέσως μετά, ελέγχεται η ημερομηνία καταπόσο είναι όντως ημερομηνία και σε σωστό format με μια δική μου μέθοδο (προτελευταία εικόνα), η οποία κάνει τον έλεγχο με κανονικές εκφράσεις και την build in μέθοδο της php την checkdate. Εάν μέχρι εδώ όλα είναι σωστά τότε απλά ο χρήστης συνεχίζει. Σε αντίθετη περίπτωση, επιστρέφεται το κατάλληλο μήνυμα και τερματίζει.

Στις γραμμές 186-201 ελέγχεται η ημερομηνία ούτως ώστε να εξασφαλίζεται να μην είναι αργία. Εάν είναι αργία, επιστρέφεται και πάλι το κατάλληλο μήνυμα και τερματίζει.

Στις γραμμές 202-211 ελέγχεται κατά πόσο υπάρχει άλλη δέσμευση αίθουσας βάση τριών διαφορετικών περιπτώσεων α) βάσει καθηγητή τη συγκεκριμένη ώρα και/ή μέρα, β) βάσει αίθουσας τη συγκεκριμένη ώρα και/ή μέρα και γ) βάσει μαθήματος τη συγκεκριμένη ώρα και/ή μέρα (μέθοδος τελευταίας εικόνας). Εάν και εδώ δεν υπάρξει κανένα πρόβλημα, τότε στις γραμμές 212 – 258 γίνονται τα εξής:

Έλεγχος του id του συνδεδεμένου χρήστη εάν είναι ο ίδιος με του καθηγητή που έκανε την καταχώρηση ή εάν είναι ο διαχειριστής. Κάποια από τα στοιχεία το σύστημα τα εντοπίζει στο \$_SESSION πίνακα που συμπληρώνεται με την σύνδεση του χρήστη. Τότε το σύστημα δοκιμάζει να ενημερώσει την ανακοίνωση της συγκεκριμένης εγγραφής και τέλος γίνεται η προσπάθεια να ενημερωθεί η ίδια η εγγραφή.

Σε κάθε περίπτωση λάθους ή ομαλής εκτέλεσης των σεναρίων, το σύστημα συμπληρώνει ένα log file με μερικά σχόλια για το τι έχει εκτελεστεί.

4.7 Παράδειγμα 4 – Διαγραφή εγγραφής.

Το τρίτο παράδειγμα που θα εξηγηθεί είναι πολύ πιο εύκολο και σύντομο και αφορά την διαγραφή μιας καταχώρησης.

Όταν ο καθηγητής που έκανε τη δέσμευση ή ο διαχειριστής του συστήματος θελήσουν, μπορούν πολύ απλά να διαγράψουν την δέσμευση πατώντας το κουμπί «DELETE». Αμέσως ενεργοποιείται το event και καλείται μια μέθοδος όπου ελέγχει το **id** του καθηγητή, ο οποίος έκανε την κράτηση και το **id** του συνδεδεμένου χρήστη και εάν ταιριάζουν, τότε το σύστημα του επιτρέπει να προχωρήσει. Αυτός ο έλεγχος γίνεται και στην ενέργεια της τροποποίησης μιας δέσμευσης. Εάν δεν ταιριάζει, τότε ελέγχει το **role** του χρήστη και εάν είναι **administrator** (διαχειριστής) τότε και πάλι του επιτρέπει να προχωρήσει στην διαγραφή.

Μετά τον έλεγχο, ακολουθεί ένα παράθυρο που ζητά να επιβεβαιώσουμε την ενέργεια (javascript confirmation alert) και όταν γίνει η επιβεβαίωση, τότε πραγματοποιείται ένα **POST** αίτημα προς το κατάλληλο php script που θα κάνει τους κατάλληλους ελέγχους και την διαγραφή.

Ο javascript κώδικας για τον έλεγχο και αποστολή των δεδομένων

```
152  /**
153  * Fires when the user clicks the delete button.
154  * An ajax call will try to delete the entry IF its the owner
155  * or if its the administrator
156  */
157  $('#deleteButton').click(function(){
158  // Check if the user can
159  if(currentUserCan($('#id_fillForm_professor').val())){
160  if (confirm("Are you sure you want to delete this reservation?")) {
161  $.ajax({
162  type : "POST",
163  url : 'calendarCrud.php',
164  data : {action: 'delete' , hallReservationId :deleteRecordWithId},
165  success:( function(data){
166  alert(data);
167  }),
168  error: (function(err) { alert("error: "+err); }),
169  complete: (function() {
170  // If deleted, rebuild the calendar
171  rebuildCalendar();
172  })
173  });
174  }
175  } else {
176  alert("You dont own this reservation so you cannot change or delete it.");
177  }
178  // And remove the fillFormDiv
179  $('#fillFormDiv').hide('explode',{}, 'slow');
180  });
647  /**
648  * Checks if the logged in user can perform actions on reservation
649  * @returns {Boolean}
650  */
651  function currentUserCan(profId) {
652  if(pid==profId){
653  return true;
654  } else if(role=='administrator') {
655  return true;
656  } else {
657  return false;
658  }
659  }
```

Ο php κώδικας για τον έλεγχο και διαγραφή της εγγραφής

Πτυχιακή Εργασία του Χαραλάμπους Μάριου

```
102 * Check if the user can perform this action.
103 * If true, delete.
104 * mysqli = Needs because we want to execute multiple queries on one time.
105 */
106 function delete($rid, $link, $log){
107     // Check if someone is connected properly
108     if(isset($_SESSION['user_id'])){
109         // $mysqli = new mysqli(DB_HOST, DB_USERNAME, "", DB_DATABASE);
110         $mysqli = new mysqli(DB_HOST, DB_USERNAME, DB_PASSWORD, DB_DATABASE);
111         // Delete the announcement first and then the reservation.
112         $sql_delete = "DELETE FROM announcements WHERE halls_reservations_id='"._insert($rid)."'";
113                     DELETE FROM halls_reservations
114                     WHERE id='"._insert($rid)."' ";
115         // If its professor, confirm that its his resevation
116         if($_SESSION['user_role']=='professor'){
117             $sql_delete .= " AND professors_id='"._insert($_SESSION['user_id'])."'";
118         } elseif($_SESSION['user_role']=='administrator'){
119             // If the user is logged in as administrator then delete only by the id of the record.
120         } else {
121             // every other user_role redirect them to login page
122             header( 'Location: ./login/index.php' );
123         }
124         $result_delete = mysqli_multi_query($mysqli, $sql_delete);
125         if(!$result_delete) {
126             echo 'Could not delete data. Please try again.. ';
127             $log->lwrite('MySQL Error while deteting: ' . mysqli_error());
128             exit;
129         }

130         // Confirm the delete action.
131         $sql_confirm = "SELECT id
132                     FROM halls_reservations
133                     WHERE id='"._insert($rid)."' AND professors_id='"._insert($_SESSION['user_id'])."'";
134         $result_confirm = mysqli_query($sql_confirm, $link);
135         if(!$result_confirm) {
136             echo 'Could not query database. Please try again. ';
137             $log->lwrite('MySQL Error while quering: ' . mysqli_error());
138         }
139         if(isset($result_confirm)){
140             echo "Deleted successfully.";
141             $log->lwrite("Deleted successfully.");
142         } else {
143             echo "Soomething went wrong. Please try again. ";
144         }
145     } else {
146         echo "You are not connected so you can't perform this action.";
147         $log->lwrite("Unauthorized user tried to delete.");
148     }
149     //close connections
150     mysqli_close($link);
151     $mysqli->close();
152 }
```

Για τις javascript μεθόδους δεν θα αναφερθώ περισσότερο από ότι στην εισαγωγή του τμήματος αυτού, αφού έχουμε περιγράψει και πιο πριν παρόμοιο κώδικα.

Δεν θα αναφερθώ επίσης περαιτέρω και για την php delete μέθοδο, παρά μόνο να πω ότι στις γραμμές 112-114 δημιουργείται ένα sql query στο οποίο θα προστεθεί 116-118 ένας μικρός έλεγχος για το καταπόσο είναι κάποιος συνδεδεμένος. Εάν είναι καθηγητής, τότε στο where μέρος του query προστίθεται ακόμη ένας περιορισμός, αυτός του id του καθηγητή που είναι συνδεδεμένος, οπότεν έτσι το σύστημα θα αποτρέψει κάποιον άλλο καθηγητή που κατά λάθος ή σκόπιμα θα δοκιμάσει και καταφέρει να φτάσει μια ενέργεια διαγραφής μέχρι σε εκείνο το σημείο.

4.8 Διατήρηση αρχείων ιστορικού

Η οργάνωση και συντήρηση ενός αρχείου ιστορικού (log file) στοχεύει στη διατήρηση και παρακολούθηση γεγονότων του συστήματος. Τα αρχεία αυτά αποτελούν

τους κυριότερους χώρους έρευνας σε περίπτωση που για κάποιο λόγο το σύστημα έχει κάνει λάθος ενέργειες που από αυτό θα μπορούμε να πάρουμε πληροφορίες πότε και ποιος έκανε την ενέργεια ή απλά εάν θέλουμε να μάθουμε πότε έγινε και τι.

Τρόπος δημιουργίας και χρήσης

Στο παρών σύστημα έχω δημιουργήσει μια κλάση η οποία έχει 4 απλές μεθόδους, μέθοδος που καθορίζει το μονοπάτι (path) για το αρχείο, μια μέθοδο δημιουργίας ή ανοίγματος του αρχείου σε περίπτωση που υπάρχει, μια μέθοδος για το κλείσιμο του αρχείου και μια μέθοδος εγγραφής στο αρχείο.

Σε κάθε PHP script που υπάρχει στον διακομιστή, υπάρχει η εισαγωγή της κλάσης, δημιουργία και καθορισμός του μονοπατιού και του ονόματος του αρχείου στο οποίο όνομα προσθέτει το σύστημα αυτόματα την ημερομηνία για τον καλύτερο διαχωρισμό των αρχείων.

Κάθε μέθοδος-ενέργεια του script έχει ως παράμετρο και ένα αντικείμενο τύπου log το οποίο χρησιμοποιεί για να γράψει το ιστορικό με μήνυμα το όνομα του script, το όνομα της μεθόδου που τρέχει, το email του χρήστη που είναι συνδεδεμένος και το μήνυμα που θέλει να αποθηκεύσει.

Κεφάλαιο 5 Ασφάλεια. [25] [27][28][29][30][31][37][38]

Η ασφάλεια είναι ένας σημαντικό στοιχείο μιας εφαρμογής που όμως συχνά δε συνυπολογίζεται στο βαθμό που θα έπρεπε κατά την ανάπτυξη και τον κύκλο ζωής της. Αυτό έχει σαν αποτέλεσμα την προσπάθεια επιδιόρθωσης κενών ασφάλειας μετά την κυκλοφορία της και συχνά, αφού έχουν εμφανιστεί τα πρώτα προβλήματα. Ως κομμάτι της παρούσας πτυχιακής εργασίας, έγινε προσπάθεια να προβλεφθούν τα περισσότερα για γι'αυτό θα ακολουθήσει η περιγραφή κάποιων κύριων επιθέσεων που αντιμετωπίζουν οι διαδικτυακές εφαρμογές και θα αναφερθούν οι τρόποι και τα μέτρα που λήφθηκαν για την αντιμετώπιση τους.

5.1 Brute force και επιθέσεις λεξικού

Μία κοινή απειλή που αντιμετωπίζουν οι web developers είναι μια απόπειρα εντοπισμού του κωδικού πρόσβασης που είναι γνωστή ως μια επίθεση brute-force (ωμής βίας). Η brute-force επίθεση είναι μια προσπάθεια ανεύρεσης ενός κωδικού πρόσβασης με τη συστηματική προσπάθεια κάθε πιθανού συνδυασμού γραμμάτων, αριθμών και συμβόλων, μέχρι την ανεύρεση ενός σωστού συνδυασμού που λειτουργεί. Ένα web site που απαιτεί έλεγχο ταυτότητας χρήστη, είναι ένας καλός στόχος για μια τέτοια επίθεση.

Ένας εισβολέας πάντα μπορεί να ανακαλύψει έναν κωδικό με μία brute-force επίθεση, αλλά το μειονέκτημα είναι ότι μπορεί να πάρει χρόνια για να το βρει. Ανάλογα με το μήκος και την πολυπλοκότητα του κωδικού, θα μπορούσε να πάρει τρισεκατομμύρια πιθανούς συνδυασμούς. Για να επιταχυνθούν τα πράγματα λίγο, μια brute-force επίθεση θα μπορούσε να ξεκινήσει με λέξεις από ένα λεξικό ή ελαφρώς τροποποιημένη λέξεις από το λεξικό, επειδή οι περισσότεροι άνθρωποι χρησιμοποιούν αυτούς παρά ένα απολύτως

τυχαίο κωδικό πρόσβασης. Αυτές οι επιθέσεις είναι επιθέσεις που ονομάζονται λεξικό ή υβριδικές brute-force επιθέσεις. Οι brute-force επιθέσεις θέτουν τους λογαριασμούς των χρηστών σε κίνδυνο και φορτώνουν το site με περιττή κυκλοφορία.

Οι Hackers (εισβολείς) ξεκινούν μια brute-force επίθεση με τη χρήση ευρέως διαθέσιμων εργαλείων που χρησιμοποιούν λίστες λέξεων και ενός συνόλου έξυπνων κανόνων για να μαντέψουν έξυπνα και αυτόματα κωδικούς πρόσβασης χρηστών. Παρά το γεγονός ότι τέτοιου είδους επιθέσεις είναι εύκολες να ανιχνευθούν, δεν είναι τόσο εύκολο να αποφευχθούν.

Για παράδειγμα, πολλά HTTP brute-force εργαλεία μπορεί να μεταβιβάζουν τα αιτήματα μέσω μιας λίστας ανοικτών διακομιστών μεσολάβησης (open proxy servers). Δεδομένου ότι κάθε αίτηση φαίνεται να προέρχεται από μια διαφορετική διεύθυνση IP, δεν είναι δυνατόν να αποφευχθούν οι επιθέσεις αυτές απλά μπλοκάροντας τις διεύθυνση αυτές IP. Για να περιπλέξει περισσότερο τα πράγματα, ορισμένα εργαλεία δοκιμάζουν ένα διαφορετικό όνομα χρήστη και κωδικό πρόσβασης σε κάθε προσπάθεια, έτσι δεν μπορεί απλά να κλειδωθεί ένας ενιαίος λογαριασμός για τις αποτυχημένες προσπάθειες πρόσβασης.

5.2 Denial of service (DoS)

Η Denial of Service επίθεση εστιάζει στο να γίνει ένας πόρος μη διαθέσιμος (ιστοσελίδα, αίτηση, server), για τον σκοπό τον οποίο έχει σχεδιαστεί. Οι δράστες των επιθέσεων DoS κτυπούν στόχους υπηρεσίες που φιλοξενούνται σε υψηλού προφίλ διακομιστές (high-profile webservers), όπως τράπεζες, πύλες πληρωμής με πιστωτικές κάρτες, ακόμα και τους κύριους nameservers.

Μια κοινή μέθοδος επίθεσης περιλαμβάνει τον κατακλυσμό της μηχανής-στόχου με εξωτερικές αιτήσεις, έτσι ώστε να μην μπορεί να ανταποκριθεί στην κυκλοφορία, ή να ανταποκρίνεται τόσο αργά ώστε να καταστεί όντως μη-διαθέσιμος. Οι επιθέσεις αυτές συνήθως οδηγούν σε υπερφόρτωση του διακομιστή. Σε γενικές γραμμές, οι επιθέσεις DoS υλοποιούνται είτε αναγκάζοντας τον υπολογιστή προορισμού να επανεκκινήσει ή να καταναλώνει τους πόρους του ώστε να μπορεί πλέον να παράσχει επαρκώς τις υπηρεσίες του.

Αν μια υπηρεσία λαμβάνει ένα πολύ μεγάλο αριθμό αιτήσεων, μπορεί να σταματήσει την παροχή υπηρεσιών στους χρήστες. Κατά τον ίδιο τρόπο, μια υπηρεσία μπορεί να καταστεί εκτός λειτουργίας αν μια προγραμματιστική ευπάθεια αποκαλυφθεί, ή αποκαλυφθεί ο τρόπος με τον οποίο η υπηρεσία χειρίζεται πόρους που χρησιμοποιήθηκαν από αυτήν.

Μερικές φορές, ο εισβολέας μπορεί να προκαλέσει την εκτέλεση τμήματος κώδικα κατά την εκτέλεση μιας επίθεσης DoS, προκειμένου να αποκτήσει πρόσβαση σε κρίσιμες πληροφορίες ή να εκτελέσει σύνολο εντολών στον διακομιστή.

5.3 Sql injection

SQL injection είναι μια τεχνική που χρησιμοποιείται συχνά για την επίθεση σε βάσεις δεδομένων μέσω ενός δικτυακού τύπου. Αυτό γίνεται με τη εισαγωγή SQL

τημάτων κώδικα σε ένα πεδίο εισαγωγής κειμένου (text-field ή text-area) της ιστοσελίδας σε μια προσπάθεια να στείλει και να εκτελέσει μια εντολή SQL στην βάση δεδομένων (π.χ. να επιστρέψει το περιεχόμενο της βάσης δεδομένων στον εισβολέα).

Η SQL injection είναι μια τεχνική εισαγωγής κώδικα που εκμεταλλεύεται μια ευπάθεια ασφαλείας σε ένα website. Η ευπάθεια υπάρχει όταν ένα πεδίο εισαγωγής στοιχείων χρήστη φιλτράρεται λανθασμένα για την αποφυγή απαγορευμένων χαρακτήρων ενσωματωμένα σε ερωτήματα SQL και εκτελείται χωρίς να πρέπει.

Οι εντολές SQL περνάνε έτσι από τη διαδικτυακή φόρμα στη βάση δεδομένων της εφαρμογής (σαν ένα SQL ερώτημα) με σκοπό να αλλάξει το περιεχόμενο της βάσης δεδομένων ή να επιστρέψει στον εισβολέα τις πληροφορίες που υπάρχουν στη βάση όπως στοιχεία πιστωτικών καρτών ή κωδικούς πρόσβαση.

5.4 Blind SQL injection.

Blind SQL Injection είναι η ίδια με την τεχνική που μόλις περιγράψαμε αλλά αυτή χρησιμοποιείται όταν μια διαδικτυακή εφαρμογή είναι ευάλωτη σε SQL injection επίθεση, αλλά τα αποτελέσματα της δεν είναι ορατά από τον εισβολέα. Η σελίδα με την ευπάθεια μπορεί να μην είναι αυτή που εμφανίζει τα δεδομένα, αλλά θα εμφανιστούν διαφορετικά ανάλογα με τα αποτελέσματα μιας λογικής δήλωση SQL που εκτελείται στη συγκεκριμένη σελίδα.

Αυτός ο τύπος επίθεσης μπορεί να αποβεί χρονοβόρος, διότι μια νέα δήλωση θα πρέπει να κατασκευαστεί για κάθε bit που θα ανακτηθεί ως αποτέλεσμα προηγούμενου ερωτήματος.

Πχ. Με το παρακάτω ερώτημα θα μάθουμε εάν υπάρχει χρήστης με username=ooo

```
SELECT 1/0 FROM users WHERE username='ooo';
```

Υπάρχουν φυσικά διάφορα εργαλεία που μπορούν να αυτοματοποιήσουν τις επιθέσεις αυτές όταν η θέση και οι πληροφορίες για το στόχο έχουν καθοριστεί.

5.5 Rainbow tables

Για να καταλάβουμε πώς λειτουργούν οι rainbow tables , θα πρέπει πρώτα να κατανοήσουμε πώς αποθηκεύονται οι κωδικοί στους υπολογιστές, είτε πρόκειται για την επιφάνεια εργασίας, είτε ένα απομακρυσμένο web server.

Οι κωδικοί πρόσβασης δεν αποθηκεύονται ποτέ σε απλή μορφή κειμένου. Τουλάχιστον δεν θα έπρεπε να είναι αποθηκευμένοι έτσι, εκτός αν χιτίζετε το πιο ανασφαλές σύστημα του κόσμου χρησιμοποιώντας τους πιο αφελείς προγραμματιστές. Αντ 'αυτού, οι κωδικοί πρόσβασης πρέπει να αποθηκεύονται ως το αποτέλεσμα μιας συνάρτησης κατακερματισμού (hash function). Οι συναρτήσεις αυτές είναι μονόδρομης λειτουργίας. Ακόμα κι αν ένας εισβολέας αποκτήσει πρόσβαση στην κατακερματισμένη έκδοση του κωδικού, δεν είναι δυνατόν να βρει τον αρχικό κωδικό από την κατακερματισμένη τιμή και μόνο.

Μπορεί όμως να επιτεθεί στον κατακερματισμένο κωδικό χρησιμοποιώντας rainbow tables. Οι rainbow tables είναι τεράστιες, προ-κατακερματισμένες τιμές που υπολογίστηκαν για κάθε πιθανό συνδυασμό χαρακτήρων. Ένας υπολογιστής που επιτίθεται θα μπορούσε να υπολογίσει όλους αυτούς τους κωδικούς on the fly, αλλά εκμεταλλευόμενος έναν τεράστιο πίνακα με προ-υπολογισμένες τιμές κατακερματισμένων κωδικών, επιτρέπει στην επίθεση να γίνει πολλές φορές πιο γρήγορα, αν υποθεθεί ότι η μηχανή επίθεσης διαθέτει αρκετή μνήμη RAM για να αποθηκεύσει ολόκληρο τον πίνακα (ή τουλάχιστον το μεγαλύτερο μέρος της) στη μνήμη. Είναι ένα κλασικό χρόνο μνήμη tradeoff, ακριβώς το είδος της εξαπάτησης συντόμευσης που θα περίμενε κανείς από? ένα μαύρο καπέλο εισβολέα να αναλάβει.

5.6 Μέτρα που λήφθηκαν

1) Επειδή και στην προκείμενη διαδικτυακή εφαρμογή υπάρχει η ανάγκη εισόδου κάποιων χρηστών πριν την χρήση της, έπρεπε να ληφθούν κάποια μέτρα για τις περιπτώσεις επίθεσης.

Κατά την προσπάθεια του χρήστη να εισέλθει στο σύστημα βάζοντας το όνομα χρήστη (username) και κωδικό πρόσβασης (password) μπορεί να εισάγει λάθος στοιχεία, είτε κατά λάθος είτε εσκεμμένα εάν πρόκειται για κάποιο εισβολέα. Και για τις δύο περιπτώσεις όμως, το σύστημα κρατάει την IP διεύθυνση και σε κάθε επόμενη λάθος προσπάθεια αυξάνει έναν μετρητή ο οποίος μόλις φτάσει τις 4, μπλοκάρει της IP αυτή από το να μπορεί να δοκιμάσει ξανά να εισέλθει για κάποιο χρονικό διάστημα, το οποίο είναι προκαθορισμένο στον κώδικα. Αφού περάσει το χρονικό διάστημα αυτό, ο χρήστης-IP έχει και πάλι την δυνατότητα να δοκιμάσει να εισέλθει.

Αυτή η τεχνική αποτρέπει τις brute-force και τις επιθέσεις λεξικού, διότι εάν για κάθε τέσσερις λάθος δοκιμές θα έπρεπε να περιμένει ένα λεπτό, τότε αυτό θα έκανε τον χρόνο που χρειάζεται για την εύρεση κωδικών τεράστια και ασύμφορη.

2) Αν υποθέσουμε ότι το μηχάνημα επίθεσης έχει την δυνατότητα να αλλάζει IP μετά από τρεις λάθος προσπάθειες εύρεσης κωδικού, σημαίνει ότι το μέτρο που λήφθηκε και περιγράφηκε ως προηγούμενο μέτρο δεν θα λειτουργήσει, άρα θα έχει την δυνατότητα να κάνει ανενόχλητο 4*πληθος των IP δοκιμές, που μάλλον είναι αρκετές για την εύρεση ενός απλού κωδικού.

Για αυτό το λόγο το σύστημα χρησιμοποιεί δύο απλές τεχνικές.

- Αναγκάζει τον χρήστη να χρησιμοποιεί κωδικούς με ελάχιστο μήκος οκτώ χαρακτήρων και τον προτρέπει να χρησιμοποιεί γράμματα, αριθμούς και ειδικά σύμβολα. Δηλαδή, εάν το αλφάβητο εισόδου έχει μήκος 96 (π.χ. πεζά και κεφαλαία γράμματα της λατινικής αλφαβήτου : 52, αριθμητικοί χαρακτήρες : 10, σύμβολα και σημεία στίξης από το πληκτρολόγιο : 34), τότε για κωδικούς μήκους 8 $\Rightarrow 96^8 = 7,213,895,789,838,336$ πιθανοί κωδικοί οι οποίοι θα πρέπει να δοκιμαστούν.

- Αν αυτό και πάλι δεν είναι αρκετό η δεύτερη τεχνική που χρησιμοποιήθηκε είναι η χρήση αλατιού (salt) που παράχθηκε από τελείως τυχαίους χαρακτήρες το οποίο ενώνεται μαζί με τον κωδικό και έχει ως ένα αποτέλεσμα hashed κωδικού.

Δηλαδή `$hashed_pass = md5($pass.SALT.md5($pass));`

3) Όπως έχει ήδη διαφανεί στην sql injection τεχνική, ο χρήστης μπορεί να στείλει κάποια κομμάτια κώδικα μέσω των πεδίων εισόδου δεδομένων με ίσως καταστροφικά αποτέλεσμα για το σύστημα μας. Για την αποφυγή του προβλήματος αυτού έχουν ληφθεί δύο μέτρα.

- Σε σχεδόν όλα τα πεδία τα οποία έχει ο χρήστης να συμπληρώσει έχουν τεθεί επαληθεύσεις των εισαχθέντων δεδομένων. Δηλαδή, στα πεδία που πρέπει να εισαχθεί ημερομηνία, έχει χρησιμοποιηθεί ένα plug-in το οποίο εμφανίζει ημερολόγιο, όπως θα αναλυθεί στο επόμενο κεφάλαιο, από το οποίο ο χρήστης μπορεί να επιλέξει. Πέρα από την πλευρά του πελάτη (client side), κανονικές εκφράσεις έχουν χρησιμοποιηθεί και στην πλευρά του διακομιστή (server side) έτσι ώστε εάν κάποιος κακόβουλος χρήστης καταφέρει με κάποιο τρόπο να στείλει λάθος δεδομένα, αυτά θα εντοπιστούν και θα απορριφθούν.

The screenshot displays a web interface with two main components. On the left, there is a calendar for April 2012, with the date 19th highlighted. On the right, there is a form titled 'Add a new Semester'. The form contains several input fields: 'Name' (filled with 'asdasd'), 'Surname' (filled with 'asdasd'), 'Website' (filled with 'asdasd' and a red error message 'Please enter a valid URL'), 'Email' (filled with 'asdasd' and a red error message 'Please enter a valid email address'), 'Role' (a dropdown menu set to 'Select One'), 'Username' (empty with a red error message 'This field is required'), and 'Password' (empty with a red error message 'This field is required').

- Πριν την ενσωμάτωση των αποσταλμένων στον διακομιστή στοιχείων στα κατάλληλα SQL ερωτήματα, τα στοιχεία αυτά περνάνε από μια προσαρμοσμένη συνάρτηση (custom function) η οποία χρησιμοποιεί δύο ήδη υπάρχουσες PHP μεθόδους για το φιλτράρισμα του αλφαριθμητικού.

```
/**
 *
 * Function to help us prevent bad user input data.
 * Returns cleaned string.
 * @param string $str
 */
function _insert($str){
    return mysql_real_escape_string($str);
}
```

Κεφάλαιο 6 Τρεις ρόλοι και περιπτώσεις χρήσης τους [26]

Η διαδικτυακή αυτή εφαρμογή έχει την δυνατότητα να χωρίζει τους χρήστες σε τρεις διαφορετικούς ρόλους, που ο καθένας θα έχει την δυνατότητα σε ορισμένες ενέργειες, όπως οι απλοί χρήστες, οι καθηγητές που θα έχουν λίγες περισσότερες, και ο(οι) διαχειριστής(ές) που θα έχουν πλήρη έλεγχο της εφαρμογής και των δεδομένων.

Πρώτα θα παρουσιαστεί ο ρόλος με τις λιγότερες δυνατότητες, λόγω του ότι οι δυνατότητες αυτές δεν θα ξανά περιγραφούν για τους επόμενους ρόλους.

6.1 Απλοί χρήστες.

Οι χρήστες-επισκέπτες της εφαρμογής δε χρειάζονται να συνδεθούν κάπου και επιπλέον έχουν τις δυνατότητες προβολής όλων των δεσμεύσεων που έχουν γίνει με τρεις διαφορετικούς τρόπους α) Βάσει του καθηγητή β) Βάσει της αίθουσας και γ) Βάσει το μάθημα. Ο χρήστης απλά επιλέγει μια από τις πιο πάνω επιλογές και μια ημέρα από το ημερολόγιο το οποίο αυτομάτως επιλέγει όλη την βδομάδα (Δευτέρα - Κυριακή) και, αφού γίνουν οι απαραίτητες ενέργειες από το σύστημα, επιστρέφει τα αποτελέσματα συμπληρώνοντας το εβδομαδιαίο ημερολόγιο με τα δεδομένα. Μετά τη συμπλήρωση του ημερολογίου, υπάρχει η δυνατότητα επιλογής μιας συγκεκριμένης δέσμευσης για την προβολή περισσότερων στοιχείων, όπως σχετική ανακοίνωση που μπορεί να υπάρχει και ο βοηθός καθηγητής μαθήματος. Επίσης υπάρχει και η δυνατότητα για προβολή της προηγούμενης και της επόμενης βδομάδας.

Επιλογή καθηγητή και εβδομάδας.

The screenshot shows a web interface for selecting a week and hall. At the top, there are buttons for '<< Previous Week' and 'Next Week >>'. Below them is a 'Select Week' dropdown menu showing '2012-04-09 / 2012-04-15'. To the left, there is a 'Select Hall' dropdown menu with 'Σιδηρόπουλος Αντώνης' selected. To the right, there is a 'Hide weekend' button. The main part of the interface is a calendar grid for April 2012. The grid has columns for days of the week (Su, Mo, Tu, We, Th, Fr, Sa) and rows for time slots (08:00-10:00, 10:00-12:00, 12:00-14:00). The 9th of April is highlighted in orange, indicating the selected date. The calendar grid is partially obscured by a dropdown menu showing the days of the week and the dates of the month.

Επιλογή δέσμευσης για εκτενέστερη προβολή

Πτυχιακή Εργασία του Χαραλάμπους Μάριου

<< Previous Week Select Week 2012-04-16 / 2012-04-22 Next Week >>

Select Hall OR Σιδηρόπουλος Αντώνης OR Select Course

	Monday	Tuesday	Wednesday	Thursday
08:00 10:00				
10:00 12:00	Διαδικτυακές Υπηρεσίες - Θ / 10:00:00 - 11:00:00 / Σιδηρόπουλος Αντώνης / 208			Αναπτυξη Διαδικτυακών Εφαρμογών - T1 / 10:00:00 - 12:30:00 / Σιδηρόπουλος Αντώνης / 208
12:00 14:00				Αναπτυξη Διαδικτυακών Εφαρμογών - T1 / 10:00:00 - 12:30:00 / Σιδηρόπουλος Αντώνης / 208
14:00 16:00			Αναπτυξη Διαδικτυακών Εφαρμογών - T1 / 15:00:00 - 16:30:00 / Σιδηρόπουλος Αντώνης / 211	

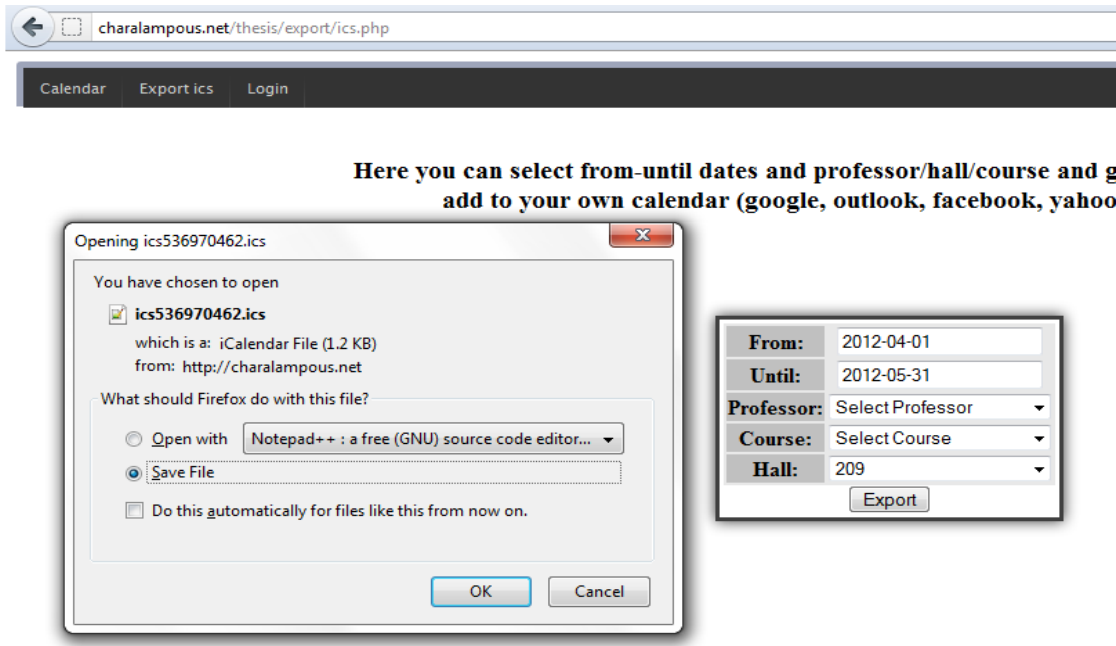
Detail view

Professor: Σιδηρόπουλος Αντώνης
Co professor: Χατζημήσιος Περικλής
Course: 09:00:00 / Σιδηρόπουλος Αντώνης / 211
Hall: 208
Date: 2012-04-19
Start time: 10:00:00
End time: 12:30:00

Announcement:
 This is the first online add reservation..
 And this is the first online edit reservation.

Η δεύτερη δυνατότητα που έχουν οι απλοί χρήστες, την οποία έχουν επίσης και οι απλοί χρήστες, είναι διαθέσιμη από το **menu** στο πάνω μέρος **“Export ics”**. Είναι η εξαγωγή και αποθήκευση στον υπολογιστή τους αρχείου με κατάληξη **.ics**, το οποίο περιέχει τις δεσμεύσεις για τις μέρες και το μάθημα / καθηγητή / αίθουσα που έχουν επιλέξει. Αυτό το αρχείο με τα δεδομένα είναι δυνατό να το εισάγουν μετά σε μια πληθώρα ημερολογίων όπως Google calendar, Yahoo calendar, Apple ical, Facebook, Microsoft , Nokia, Sun Java Calendar Server και πάρα πολλές άλλες εφαρμογές παγκόσμιας χρήσης. Ο τρόπος χρήσης είναι παρόμοιος, δηλαδή επιλέγεται δύο ημερομηνίες από-εώς (όπου το διάστημα εδώ μπορεί να είναι όσο θέλει ο χρήστης και όσο υπάρχουν δεσμεύσεις) και μία επιλογή από μαθήματα / καθηγητές / αίθουσες και ακολούθως το κουμπί **export**.

Δυνατότητα εξαγωγής δεσμεύσεων



6.2 Καθηγητές

Πέραν από τις λειτουργίες που έχουν περιγραφεί στο προηγούμενο μέρος αυτού του κεφαλαίου, ο ρόλος αυτός έχει ακόμα μερικές επιπλέον δυνατότητες.

Ένας χρήστης για να αποκτήσει αυτά τα δικαιώματα πρέπει να μεταφερθεί στην κατάλληλη σελίδα επιλέγοντας από το **menu** την επιλογή «**login**», όπου θα πρέπει να εισάγει τα δεδομένα (username και password). Στην περίπτωση που οποιοσδήποτε χρήστης εισάγει τέσσερις φορές λάθος στοιχεία, το σύστημα αυτομάτως καταχωρεί την IP του σε ένα πίνακα και χρησιμοποιεί μια σταθερά δηλωμένη στον κώδικα από τον διαχειριστή ως χρόνο (λεπτά) τον οποίο θα απαγορεύει στον χρήστη να δοκιμάσει ξανά να εισάγει στοιχεία για είσοδο του στο σύστημα..

Όταν ο χρήστης έχει ξεχάσει τον κωδικό εισόδου, αφού πληκτρολογήσει λάθος κωδικό θα του δοθεί ένας σύνδεσμος όπου, εισάγοντας την διεύθυνση ηλεκτρονικού ταχυδρομείου που υπάρχει καταχωρημένη στο σύστημα για αυτόν, θα του σταλεί ένας καινούργιος κωδικός εννέα χαρακτήρων, ο οποίος θα δημιουργηθεί τυχαία από ένα σύνολο 100 χαρακτήρων συμπεριλαμβανομένων γραμμάτων κεφαλαίων και πεζών, αριθμών και συμβόλων.

Ο χρήστης θα χρησιμοποιήσει έστω μια φορά τον κωδικό αυτό που θα του σταλεί για να συνδεθεί και ακολούθως θα μπορεί να εισάγει καινούριο.

[Σύνδεσμος για Forgot Your Password](#)

[Φόρα για νέο κωδικό](#)

<p style="text-align: center;">Login form</p> <div style="border: 1px solid gray; padding: 5px; margin-bottom: 10px;"> <p>Username: <input type="text" value="marios"/></p> <p>Password: <input type="password" value="••••••••"/></p> <p style="text-align: center;"><input type="button" value="Login"/></p> </div> <p style="text-align: center;">Whrong username or password.</p> <p style="text-align: center;">Forgot your password?</p>	<p style="text-align: center;">Restore password form</p> <div style="border: 1px solid gray; padding: 5px; margin-bottom: 10px;"> <p>Email: <input type="text"/></p> <p style="text-align: center;"><input type="button" value="Send new Password"/></p> </div>
<p style="text-align: center;"><i>Μήνυμα επιτυχούς αποστολής κωδικού</i></p>	<p style="text-align: center;"><i>Μήνυμα λάθους Email</i></p>
<p style="text-align: center;">Restore password form</p> <div style="border: 1px solid gray; padding: 5px; margin-bottom: 10px;"> <p>Email: <input type="text"/></p> <p style="text-align: center;"><input type="button" value="Send new Password"/></p> </div> <p style="background-color: #90EE90; padding: 5px; text-align: center;">The new password is activated. Please check your email. And also in SPAM.</p> <p style="text-align: center;">Login with your new credentials.</p>	<p style="text-align: center;">Restore password form</p> <div style="border: 1px solid gray; padding: 5px; margin-bottom: 10px;"> <p>Email: <input type="text"/></p> <p style="text-align: center;"><input type="button" value="Send new Password"/></p> </div> <p style="background-color: #FF0000; color: white; padding: 5px; text-align: center;">There is no such email in our database.</p>

(ποια σημαντικότερη?) Η σημαντικότερη είναι αυτή της εισαγωγής δέσμευσης μια αίθουσας για κάποιο μάθημα, μία μέρα-ώρα αλλά και η εισαγωγή δέσμευσης για όλες τις ίδιες μέρες-ώρες για το υπόλοιπο του εξαμήνου, για αυτό το μάθημα.

Με την είσοδο του χρήστη στο σύστημα, αναγνωρίζεται η ιδιότητα του ως καθηγητής και προσθέτονται σε όλα τα κελιά του πίνακα τα κατάλληλα links, όπου πατώντας πάνω σε αυτά θα εμφανιστεί η κατάλληλη φόρμα για την εισαγωγή των δεδομένων.

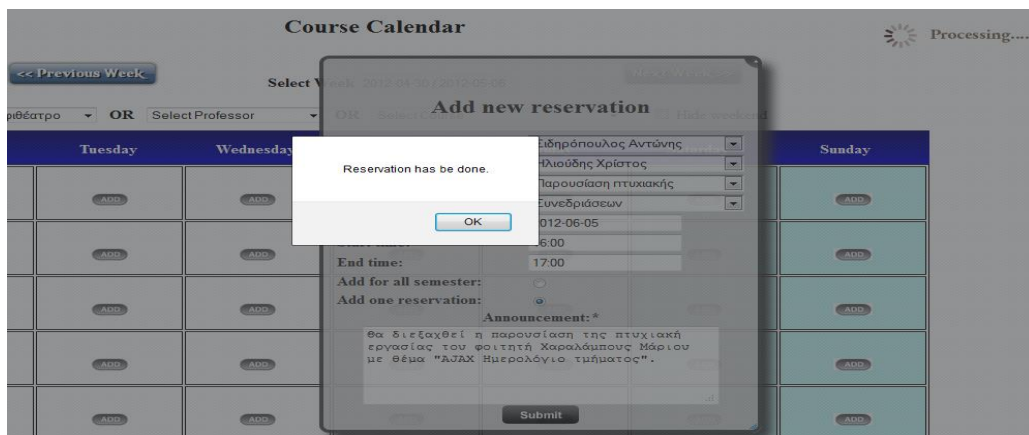
Προβολή ημερολογίου και φόρμας εισαγωγής δεδομένων

The screenshot displays a reservation system interface. At the top, there are navigation buttons: '<< Previous Week', 'Select Week 2012-04-30 / 2012-05-06', and 'Next Week >>'. Below these are dropdown menus for 'Αμφιθέατρο' and 'OR Select Professor', and another 'OR Select Course' dropdown. A 'Hide weekend' checkbox is also present. The main part of the interface is a calendar grid with columns for 'Monday', 'Tuesday', 'Wednesday', 'Friday', and 'Saturday'. The rows represent time slots from 08:00 to 20:00. Each cell in the grid contains an 'ADD' button. Overlaid on the calendar is a modal form titled 'Add new reservation'. This form includes fields for 'Professor:', '2nd Professor:', 'Course:', 'Hall:', 'Date:', 'Start time:', and 'End time:'. There are also radio buttons for 'Add for all semester:' and 'Add one reservation:'. An 'Announcement: *' field is at the bottom of the form, followed by a 'Submit' button.

Πτυχιακή Εργασία του Χαραλάμπους Μάριου

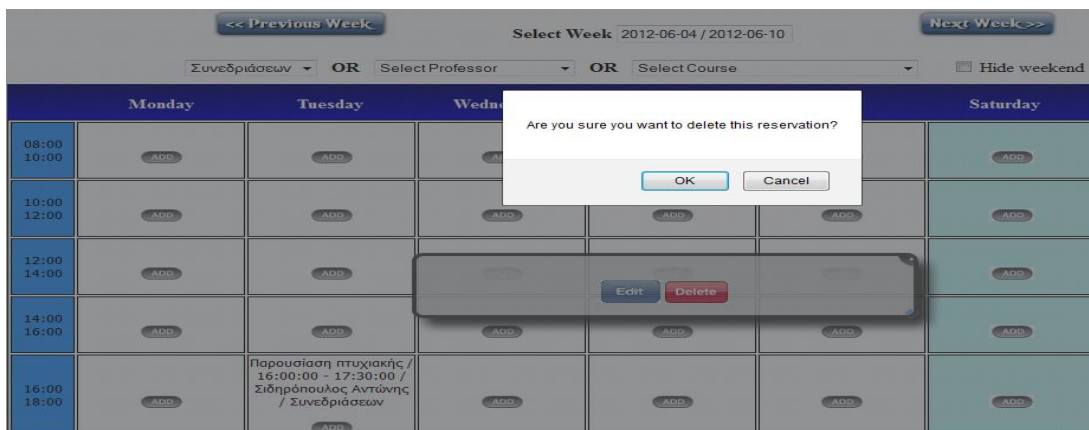
Όπως φαίνεται και από την εικόνα, υπάρχουν εννέα πεδία για επιλογή και συμπλήρωση όπου είναι ο καθηγητής, ο βοηθός καθηγητή, το μάθημα, η αίθουσα, η ημερομηνία, η ώρα αρχής και τέλους και ένα κείμενο για ανακοίνωση. Το προτελευταίο στοιχείο που δεν αναφέρθηκε είναι η επιλογή την οποία, εάν θέλει ο χρήστης, θα τεθεί ΜΙΑ μόνο δέσμευση (όπως έχει επιλεγεί που είναι και η default) ή εάν θέλει ο χρήστης, το σύστημα να κάνει όσες δεσμεύσεις είναι διαθέσιμες από αυτή την ημερομηνία που επιλεγεί μέχρι και το τέλος του εξαμήνου για την ίδια ημέρα και ώρα. Στη κάθε περίπτωση που είτε η κράτηση έγινε επιτυχώς είτε όχι, το σύστημα επιστρέφει το κατάλληλο μήνυμα.

Μήνυμα επιτυχημένης εισαγωγής.



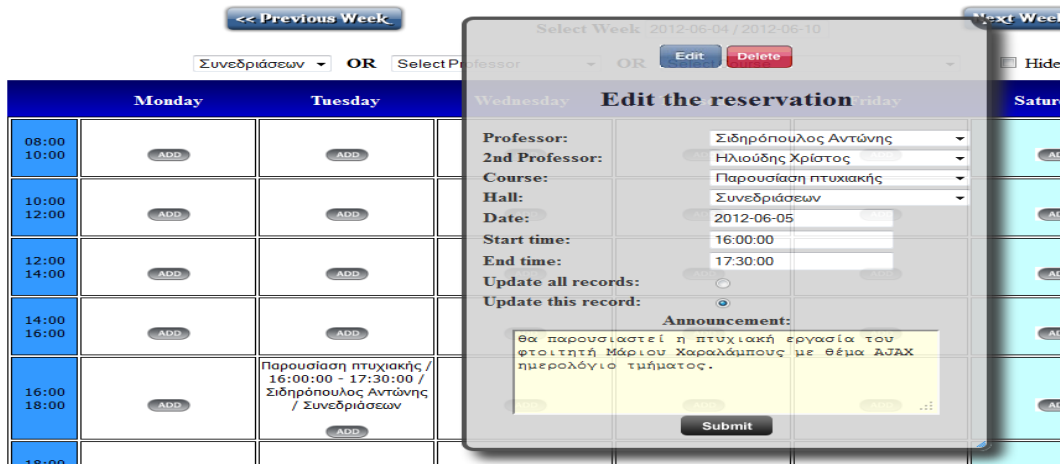
Μια άλλη δυνατότητα που έχει αυτός ο ρόλος είναι η δυνατότητα επεξεργασίας και διαγραφής των κρατήσεων των αιθουσών μόνο από τους «ιδιοκτήτες» τους, δηλαδή από τους χρήστες οι οποίοι τις έχουν δημιουργήσει και αναφέρονται σε αυτούς. Η διαδικασία είναι η ίδια με την προβολή των πληροφοριών της κράτησης όπως και στους απλούς χρήστες, με την διαφορά όμως ότι εάν η κράτηση είναι του συγκεκριμένου καθηγητή, αμέσως εμφανίζονται στην αρχή δύο επιλογές, **edit** και **delete**. Όταν ο χρήστης επιλέξει **delete** τότε το σύστημα κάνει την διαγραφή και επιστρέφει το κατάλληλο μήνυμα.

Επιλογή διαγραφής κράτησης



Όταν όμως επιλέξει **edit**, τότε ανοίγει η κατάλληλη φόρμα, που είναι παρόμοια με της εισαγωγής, στην οποία υπάρχουν τα στοιχεία συμπληρωμένα. Ο χρήστης αλλάζει ότι είναι για αλλαγή και ακολούθως πατά το **submit**. Το σύστημα και πάλι επιστρέφει το κατάλληλο μήνυμα για απάντηση στο χρήστη εάν έγινε η ενημέρωση ή όχι και τους λόγους για τους οποίους δεν έγινε η ενημέρωση, οι οποίοι μπορεί να οφείλονται σε λάθος μορφή (format) δεδομένων ή σε μία ήδη υπάρχουσα κράτηση εκείνης της επιλεγμένης μέρα-ώρα κλπ.

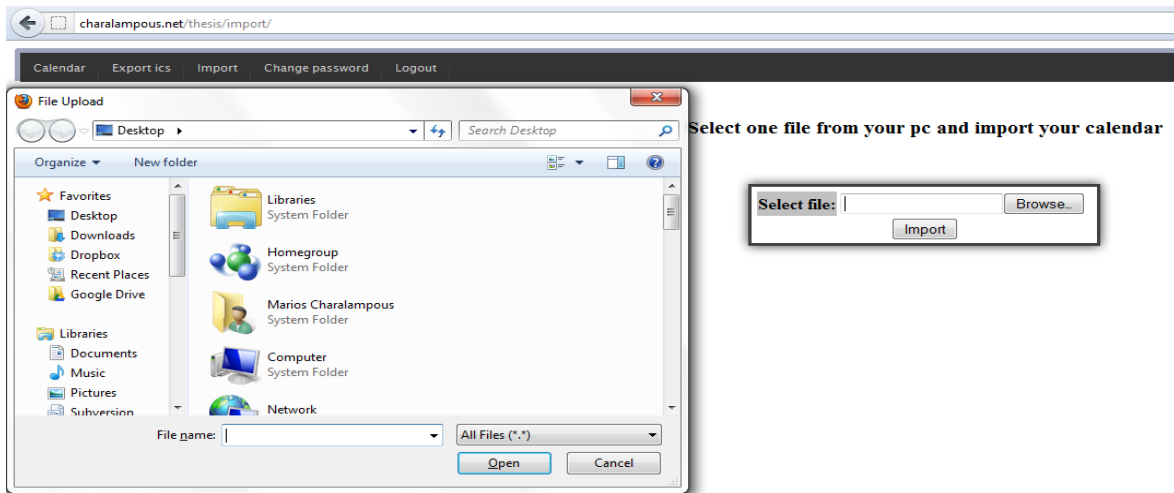
Επιλογή ενημέρωσης κράτησης



Μια άλλη δυνατότητα που έχουν οι χρήστες με αυτό το ρόλο είναι η εισαγωγή αρχείου στο σύστημα με τα δεδομένα τα οποία θέλουν να κάνουν μαζικές κρατήσεις.

Δηλαδή, ο χρήστης πρέπει να δημιουργήσει ένα αρχείο **.txt** με τις κρατήσεις βάζοντας τα δεδομένα σε μια γραμμή ανά κράτηση, έχοντας τα στοιχεία διαχωρισμένα με κόμμα «,». Έπειτα, εισάγει αυτό το αρχείο στο σύστημα. μεταβαίνοντας στην κατάλληλη σελίδα, το **link** της οποίας βρίσκει από το **menu**, επιλογή «**Import**», όπου εκεί που θα μεταφερθεί υπάρχει το κατάλληλο πεδίο για την επιλογή του αρχείου αυτού από τον υπολογιστή και η εισαγωγή του. Το σύστημα διαβάζει το αρχείο γραμμή προς γραμμή, σπάει τη γραμμή σε στοιχεία και ακολουθεί την ίδια διαδικασία που ακολουθεί και για την κανονική εισαγωγή με μόνη διαφορά ότι οι κρατήσεις που γίνονται είναι για όλο το υπόλοιπο του εξαμήνου. Και σε αυτή την περίπτωση το σύστημα κάνει όσες κρατήσεις είναι ελεύθερες και εφικτές.

Επιλογή εισαγωγής αρχείου κρατήσεων



Η τελευταία δυνατότητα που έχουν οι χρήστες με αυτό το ρόλο είναι να μπορούν οποιαδήποτε στιγμή θελήσουν να αλλάξουν τον κωδικό πρόσβασής τους στο σύστημα.

Αυτό μπορεί να γίνει πολύ εύκολα επιλέγοντας από το **menu** την αντίστοιχη επιλογή «**Change Password**» όπου θα μεταφερθεί στην κατάλληλη σελίδα. Εκεί θα ζητηθούν από το χρήστη να εισάγει τον κωδικό που χρησιμοποιεί και δύο φορές για επιβεβαίωση τον καινούριο κωδικό. Μετά από την εισαγωγή των στοιχείων αυτών και αφού πατήσει το κουμπί «**Change Password**» το σύστημα καταρχάς ελέγχει την ομοιότητα των δύο κωδικών που έβαλε ο χρήστης, νέος κωδικός και επιβεβαίωση κωδικού, και εφόσον είναι οι ίδιοι συνεχίζει στον επόμενο έλεγχο, ενώ σε αντίθετη περίπτωση επιστρέφει το κατάλληλο μήνυμα και τερματίζει.

Μήνυμα ανόμοιων κωδικών

Administration Calendar Export ics Import Change password Logout

Change password form

Old password:

New password:

Retype password:

Change Password

The new and retype passwords are not the same

Ο επόμενος έλεγχος στον οποίο προχωρεί το σύστημα είναι το μήκος του καινούριου κωδικού. Υπάρχει περιορισμός σε αυτό το σημείο και δεν αφήνει κωδικούς με μήκος κάτω των οκτώ χαρακτήρων να αποθηκευτούν (για τους λόγους που εξηγήθηκαν στο προηγούμενο κεφάλαιο).

Μήνυμα κωδικού με μικρό μήκος

The screenshot shows a web interface with a dark navigation bar containing 'Import', 'Change password', and 'Logout'. Below the bar is the title 'Change password form'. The form itself has three input fields labeled 'Old password:', 'New password:', and 'Retype password:'. A 'Change Password' button is located below the fields. A red error message at the bottom of the form reads: 'Too small password. Enter more than 8 chars'.

Στην συνέχεια ελέγχει εάν ο κωδικός που έχει εισάγει για τρέχων κωδικός είναι πράγματι ο δικός του και ο κωδικός ο οποίος έχει χρησιμοποιήσει για να εισέλθει στο σύστημα. Σε θετική περίπτωση προχωρά στην κρυπτογράφηση και αποθήκευση του κωδικού, ενώ σε περίπτωση λάθους, και πάλι ο χρήστης θα λάβει το κατάλληλο μήνυμα από το σύστημα.

Μήνυμα λάθους κωδικού

The screenshot shows the same 'Change password form' as above. The error message at the bottom of the form now reads: 'You dont own such a password'.

6.3 Διαχειριστές

Όπως έχει αναφερθεί μέχρι τώρα, στο σύστημα υπάρχουν οι δυνατότητες προβολής ημερολογίου και εξαγωγής **ics** καθώς και η δυνατότητα εισαγωγής - επεξεργασίας δικών τους κρατήσεων και για τους καθηγητές και τους απλούς χρήστες, και για τους καθηγητές υπάρχει η δυνατότητα εισαγωγής αρχείου με κρατήσεις.

Ο ρόλος των διαχειριστών είναι ο πιο κύριος, όπως θα δούμε, αφού έχει την δυνατότητα να κάνει όλα όσα αναφέρθηκαν για τους δύο προηγούμενους ρόλους αλλά και άλλες δυνατότητες οι οποίες θα εξηγηθούν στη συνέχεια.

Με την εισαγωγή του στο σύστημα υπάρχει μια επιπλέον επιλογή «**Administration**». Επιλέγοντας το σύνδεσμο (**link**) αυτό ο χρήστης μεταφέρεται στο back-end κομμάτι της εφαρμογής που είναι και οι επιπλέον δυνατότητες τους.

Κεντρικό menu του Back-end



Σε αυτό το κομμάτι της εφαρμογής ο διαχειριστής έχει την δυνατότητα να προσθέσει, διαγράψει και να τροποποιήσει τα δεδομένα τα οποία είναι καταχωρημένα στους πίνακες της βάσης όπως α) Εξάμηνα, β) Μαθήματα, γ) Αίθουσες, δ) Καθηγητές, ε) Μαθήματα – τμήματα εξαμήνου, στ) Αργίες.

Η εισαγωγή / διαγραφή / ενημέρωση των πινάκων στο σύστημα, κάνοντας χρήση κυρίως AJAX και του jQuery Datatables Plug-in, γίνεται με τον ίδιο τρόπο. Επειδή όμως υπάρχει μια μικρή ιδιαιτερότητα σε δύο πίνακες (β και ε), θωρήθηκε σημαντικό όπως περιγραφεί η διαδικασία για αυτούς τους δύο, περιγράφοντας την για την καλύτερη κατανόηση τους.

Ο πίνακας «Μαθήματα» κρατάει εγγραφές μαθημάτων, για παράδειγμα «Μαθηματική Ανάλυση», «Ψηφιακά Συστήματα», «Εισαγωγή στο προγραμματισμό Ι» κλπ, ενώ ο «Μαθήματα – τμήματα εξαμήνου» όπως έχει αναφερθεί στο κεφάλαιο 3.3.6 κρατάει εγγραφές για τα διαφορετικά τμήματα των εργαστηρίων και μία για τις θεωρίες.

Η αρχή θα γίνει με την προβολή του πίνακα μαθημάτων (**Courses**).

Προβολή πίνακα Courses

Code	Name	Type	Semester	Multitude
4702	Διαδικτυακές Υπηρεσίες	Θεωρία	7	1
4103	Εισαγωγή στην πληροφορική	Θεωρία	1	1
4702	Διαδικτυακές Υπηρεσίες	Εργαστήριο	7	1
4105	Δεξιότητες Επικοινωνίας	Θεωρία	1	1
4104	Δεξιότητες Επικοινωνίας	Εργαστήριο	2	3
4701	Αναπτυξη Διαδικτυακών Εφαρμογών	Εργαστήριο	7	2
4702	Παρουσίαση πτυχιακής	Παρουσίαση Πτυχιακής	7	1
1234	Συνεδρίαση	Συνεδρίαση	7	1
4101	Εισαγωγή στην πληροφορική	Εργαστήριο	1	5
4102	Προγραμματισμός Η/Υ Ι	Θεωρία	1	1

Φαίνεται ότι στο πάνω μέρος αριστερά υπάρχει η δυνατότητα προβολής αποτελεσμάτων ανά 10,25,50 και 100. Φυσικά αυτές οι τέσσερις επιλογές μπορούν οποιαδήποτε στιγμή να αλλάξουν από τον κώδικα όπου είναι ορισμένες. Επίσης στο κάτω δεξιό μέρος του πίνακα υπάρχουν κάποιες επιλογές first, last, next, previous και αριθμοί ανάλογα με το πλήθος των αποτελεσμάτων που είναι χωρισμένες με την πιο πάνω επιλογή.

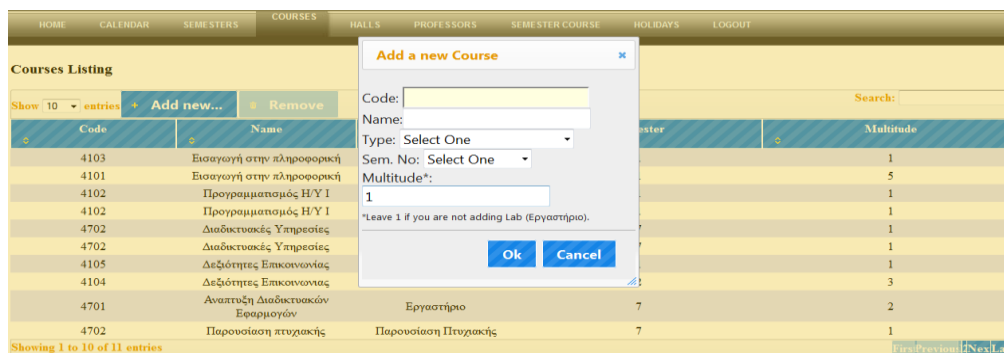
Στο κάτω αριστερό μέρος υπάρχει ένδειξη για τα αποτελέσματα τα οποία φαίνονται σε σύγκριση με όσα υπάρχουν (**Showing 1 to 10 of 11 entries**), ενώ στο πάνω δεξί μέρος υπάρχει ένα πεδίο για φιλτράρισμα των δεδομένων στο οποίο πληκτρολογώντας «Εργαστήριο» θα φιλτράρει τα δεδομένα και θα αφήνει για προβολή μόνο σε όσα υπάρχει κάπου στα πεδία τους η τιμή αυτή. Υπάρχει επίσης η δυνατότητα ταξινόμησης των δεδομένων σε αύξουσα ή φθίνουσα σειρά, πατώντας απλά σε ένα από τα πεδία κεφαλής του Datatable.

Όλα όσα περιγράφηκαν εκτελούνται στον browser, που σημαίνει ότι εκτελούνται με σχεδόν μηδέν καθυστέρηση και δεν χρειάζεται καμιά επαναφόρτωση της σελίδας.

Στο πάνω μέρος τώρα υπάρχουν άλλες δύο επιλογές για την προσθήκη νέου μαθήματος και την διαγραφή κάποιου μαθήματος.

Με την επιλογή της προσθήκης «Add New», ανοίγει μια φόρμα για τη συμπλήρωση στοιχείων, ενώ το υπόλοιπο της σελίδας είναι σκιασμένο και δεν είναι δυνατή η πρόσβαση σε αυτό έως ότου κλείσει το καινούριο παράθυρο. Στο συγκεκριμένο παράθυρο υπάρχει η δυνατότητα μετακίνησης και αλλαγής του μεγέθους του.

Add new form για προσθήκη μαθήματος



Όπως φαίνεται στην πιο πάνω εικόνα, το μόνο πεδίο το οποίο είναι συμπληρωμένο με το άνοιγμα της φόρμας είναι το πεδίο “Multitude” με τιμή “1”, το οποίο είναι για όλες τις νέες καταχωρήσεις, με τύπο μαθήματος εκτός του Εργαστηρίου (δηλαδή Θεωρία, Παρουσίαση πτυχιακής, Συνεδρίαση, Άλλο), όπου θα δημιουργήσει μια μόνο εγγραφή στον πίνακα “Courses” και μία εγγραφή στον πίνακα “Semester_courses”. Στην περίπτωση που ο διαχειριστής θέλει να εισάγει το μάθημα ενός εργαστηρίου, σε αυτό το πεδίο θα βάλει το πλήθος των εργαστηριακών τμημάτων που θα δημιουργηθούν. Έτσι εισάγοντας ένα μάθημα «Ψηφιακά Συστήματα», στον τύπο θα επιλέξει «Εργαστήριο» και

στο “**Multitude**” θα βάλει το πλήθος, πχ. 5 και τότε το σύστημα θα δημιουργήσει μια καταχώρηση σε αυτόν τον πίνακα με το όνομα «Ψηφιακά Συστήματα» και θα δημιουργήσει στον πίνακα “**Semester_courses**” πέντε νέες καταχωρήσεις με ονόματα «Ψηφιακά Συστήματα T1» , «Ψηφιακά Συστήματα T2», «Ψηφιακά Συστήματα T3», «Ψηφιακά Συστήματα T4», «Ψηφιακά Συστήματα T5» - μία για κάθε εργαστηριακό τμήμα.

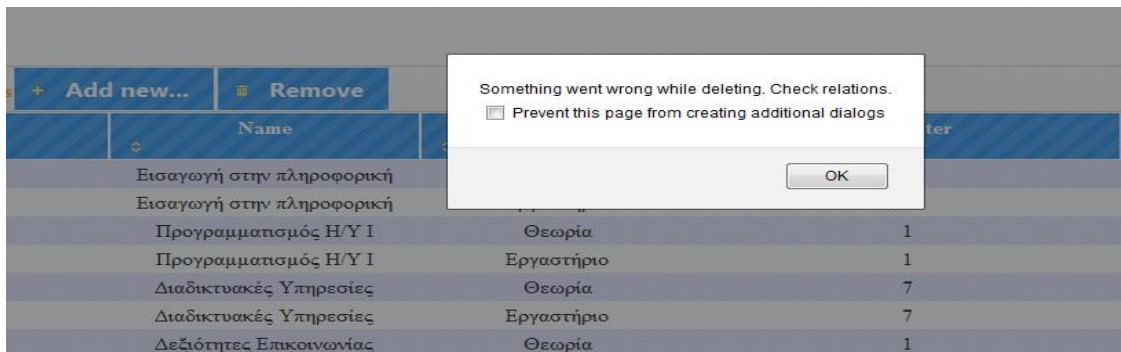
Με το πάτημα που «OK» για την αποστολή των δεδομένων για εισαγωγή, καλείται μία μέθοδος, η οποία ελέγχει τον τύπο και καταπόσο έχουν εισαχθεί δεδομένα σε όλα τα πεδία οπότεν, στην περίπτωση που ο χρήστης για κάποιο λόγο προσπαθήσει να στείλει τα δεδομένα χωρίς όμως να είναι όλα συμπληρωμένα, τότε στα πεδία τα οποία υπάρχει κάποιο πρόβλημα προστίθεται κόκκινο χρώμα και ένα μήνυμα προειδοποιώντας τον χρήστη για την λάθος εισαγωγή δεδομένων.

Λάθος εισαγωγή δεδομένων

Για να διαγραφεί μια εγγραφή αυτού του πίνακα, πρέπει πρώτα να εξακριβωθεί ότι δεν υπάρχουν εγγραφές στον πίνακα “**Semester_courses**” που να έχουν σχέση με αυτήν. Για να πραγματοποιηθεί διαγραφή χρειάζεται απλά η επιλογή της γραμμής με ένα κλικ και στη συνέχεια η επιλογή του “**Remove**”. Έτσι, αφού επιλέχθηκε η εγγραφή για διαγραφή, η επιλογή “**Remove**” έγινε ενεργή. Αμέσως εμφανίζεται ένα παράθυρο διαλόγου του browser που ζητάει από τον χρήστη την διαγραφή. Αφού ο χρήστης επιλέξει “ok” τότε γίνεται έλεγχος ότι δεν υπάρχουν εγγραφές στον άλλο πίνακα που να αναφέρονται σε αυτήν και εάν όλα έχουν καλώς, γίνεται η διαγραφή, επιστρέφει το κατάλληλο μήνυμα από τον sever το οποίο λαμβάνει το σύστημα και εάν η διαγραφή έχει γίνει επιτυχώς καλεί μία μέθοδο που καθαρίζει τα στοιχεία του πίνακα και τα ξαναφορτώνει.

Μήνυμα επιβεβαίωσης διαγραφής

Μήνυμα ύπαρξης συσχετίσεων σε άλλο πίνακα



Για την ενημέρωση των πεδίων το μόνο που χρειάζεται είναι διπλό κλικ στο κατάλληλο πεδίο και αυτομάτως θα αλλάξει ο τύπος του σε text field, text area ή dropdown ανάλογα με τον τύπο και από το που υποτίθεται πως πρέπει να πάρει τιμές. Για παράδειγμα, τα πεδία κωδικός, όνομα, και πλήθος τμημάτων, με ένα διπλό κλικ θα γίνουν text fields, ενώ τα πεδία εξαμήνου και τύπου θα γίνουν dropdown με τιμές που είναι προκαθορισμένες στον κώδικα, λόγω του ότι δεν είναι τιμές που υπάρχει μεγάλη πιθανότητα να αλλάξουν στο μέλλον. Υπάρχουν άλλα dropdowns για τα οποία τρέχει κατάλληλο Ajax post request για να γεμίσει με τιμές από τους πίνακες τα οποία θα γίνει η περιγραφή τους σε κατοπινό στάδιο.

Μετατροπή σε text fields

Code	Name	Type
4103	Εισαγωγή στην πληροφορική	Θεωρία
4101	Εισαγωγή στην πληροφορική	Εργαστήριο
4102	Προγραμματισμός Η/Υ Ι	Θεωρία
4102	Προγραμματισμός Η/Υ Ι	Εργαστήριο

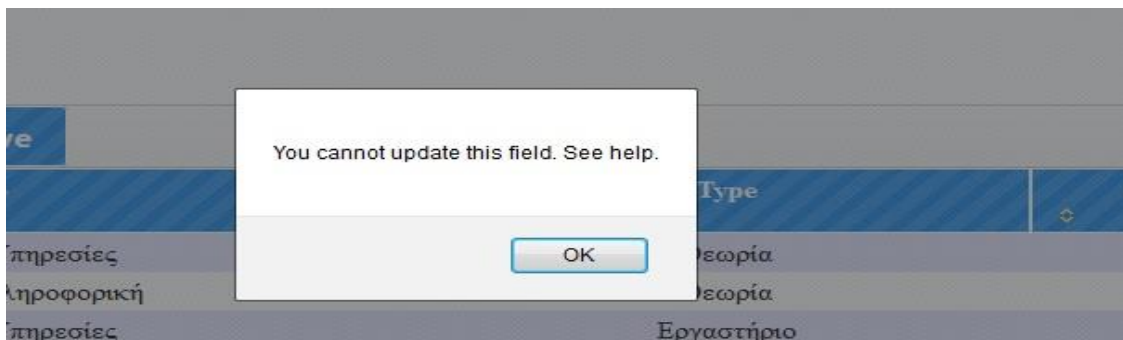
Μετατροπή σε dropdown

Name	Type	Semester
αγωγή στην πληροφορική	Θεωρία	Α εξάμηνο
αγωγή στην πληροφορική	Θεωρία	Α εξάμηνο
ρογραμματισμός Η/Υ Ι	Εργαστήριο	Β εξάμηνο
ρογραμματισμός Η/Υ Ι	Παρουσίαση Πτυχιακής	Γ εξάμηνο
ιαδικτυακές Υπηρεσίες	Συνεδρίαση	Δ εξάμηνο
	Άλλο	Ε εξάμηνο
	Θεωρία	ΣΤ εξάμηνο
		Ζ εξάμηνο

Υπάρχουν δύο γεγονότα, `onBlur` για τα dropdown και `onEnter` για τα text field και text areas, τα οποία μόλις ενεργοποιηθούν καλείται και πάλι ο κατάλληλος Javascript-AJAX κώδικας που στέλνει κάποια στοιχεία της καταχώρησης στον server, κτυπώντας το κατάλληλο script, όπου εκεί γίνεται ένας έλεγχος για τον τύπο και το περιεχόμενο των στοιχείων και εάν όλα είναι όπως πρέπει, τότε γίνεται η ενημέρωση του πίνακα στη βάση και ακολούθως η ενημέρωση του Datatable. Για ακόμη μια φορά όλα γίνονται χωρίς να υπάρξει καμία ανανέωση της ιστοσελίδας.

Κάτι τελευταίο για αυτό το πίνακα είναι ότι για το πεδίο “**Multitude**” δεν υπάρχει η δυνατότητα ενημέρωσης από το χρήστη και, εάν ο χρήστης για οποιοδήποτε λόγο το δοκιμάσει, θα του επιστραφεί το κατάλληλο μήνυμα.

Μήνυμα λάθους ενημέρωσης πεδίου “Multitude”



Αυτό το πεδίο ενημερώνεται αυτόματα με την προσθήκη ή διαγραφή εργαστηριακού τμήματος ή άλλου μαθήματος από τον πίνακα “**Semester_courses**”.

Σ’αυτό το σημείο επιλέται το **Semester_courses** από το κεντρικό μενού για να μεταφερθούμε στη σελίδα με το Datatable, η οποία δείχνει τα στοιχεία για το συγκεκριμένο πίνακα.

Σελίδα και Datatable Semester courses

A screenshot of a web application's data table. At the top, there are controls: "Show 10 entries", "Add new...", "Remove", and a search box. The table has three columns: "Course", "Semester", and "Name". It contains 15 rows of data. The first row is highlighted in light blue. At the bottom, there is a pagination bar showing "Showing 1 to 10 of 15 entries" and navigation buttons for "First", "Previous", "1", "2", "Next", and "Last".

Course	Semester	Name
Εισαγωγή στην πληροφορική - Ε	Εαρινό 2012	Εισαγωγή στην πληροφορική - T1
Εισαγωγή στην πληροφορική - Ε	Εαρινό 2012	Εισαγωγή στην πληροφορική - T2
Εισαγωγή στην πληροφορική - Ε	Εαρινό 2012	Εισαγωγή στην πληροφορική - T3
Εισαγωγή στην πληροφορική - Ε	Εαρινό 2012	Εισαγωγή στην πληροφορική - T4
Εισαγωγή στην πληροφορική - Θ	Εαρινό 2012	Εισαγωγή στην πληροφορική - Θ
Διαδικτυακές Υπηρεσίες - Θ	Εαρινό 2012	Διαδικτυακές Υπηρεσίες - Θ
Διαδικτυακές Υπηρεσίες - Ε	Εαρινό 2012	Διαδικτυακές Υπηρεσίες - T3
Προγραμματισμός Η/Υ Ι - Θ	Εαρινό 2012	Προγραμματισμός Η/Υ Ι - Θ
Εισαγωγή στην πληροφορική - Ε	Εαρινό 2012	Εισαγωγή στην πληροφορική - T5
Δεξιότητες Επικοινωνίας - Θ	Εαρινό 2012	Δεξιότητες Επικοινωνίας - Θ

Σε αυτόν τον πίνακα υπάρχουν πέντε τμήματα για το μάθημα «Εισαγωγή στην πληροφορική» και μια καταχώρηση στον προηγούμενο πίνακα.

Πτυχιακή Εργασία του Χαραλάμπους Μάριου

Code	Name	Type	Semester	Multitude
4101	Εισαγωγή στην πληροφορική	Θεωρία	1	1
4101	Εισαγωγή στην πληροφορική	Εργαστήριο	1	5

Showing 1 to 2 of 2 entries (filtered from 8 total entries)

Εάν σ' αυτό το σημείο προστεθεί ακόμη ένα εργαστηριακό τμήμα για αυτό το μάθημα, τότε το σύστημα θα αυξήσει μόνο του το πλήθος των τμημάτων και θα ενημερώσει την κατάλληλη εγγραφή στον πίνακα “Courses”.

Name	Type	Semester	Multitude
Εισαγωγή στην πληροφορική	Θεωρία	1	1
Εισαγωγή στην πληροφορική	Εργαστήριο	1	6

al entries)

Μία ιδιαιτερότητα αυτού του πίνακα που δεν υπάρχει στον προηγούμενο που εξηγήθηκε είναι ότι, τα δύο πεδία (Name και Semester) είναι dropdowns, ενώ τα δεδομένα για τις επιλογές τους δεν είναι κάπου προκαθορισμένα αλλά κάθε φορά που γίνεται διπλό κλικ και το πεδίο μετατρέπεται σε dropdown, το σύστημα στο παρασκήνιο για ακόμη μια φορά καλεί κάποιες μεθόδους στον client και στον server. Ο server παίρνει τα δεδομένα από την βάση, τα μορφοποιεί σε ένα αντικείμενο JSON και το στέλνει πίσω. Με την άφιξη του στον browser, το αντικείμενο αυτό, δέχεται μια επεξεργασία και μετατρέπεται σε επιλογές (<option>), όπου γεμίζει το κάθε dropdown.

Ο κώδικας της μεθόδου που δημιουργεί το JSON αντικείμενο θα παρατεθεί στο ΠΑΡΑΡΤΗΜΑ, ενώ ο κώδικας που το δέχεται και το μετατρέπει σε επιλογές έχει παρατεθεί στο κεφ. 3.4.4 Παράδειγμα 1 – Γέμισμα drop down στοιχείου.

Course	Semester	Name
Εισαγωγή στην πληροφορική - Θ	Εαρινό 2010	Εισαγωγή στην πληροφορική - Θ
Εισαγωγή στην πληροφορική - Θ	Εαρινό 2010	Εισαγωγή στην πληροφορική - Τ1
Εισαγωγή στην πληροφορική - Ε	Εαρινό 2010	Εισαγωγή στην πληροφορική - Τ1
Προγραμματισμός Η/Υ I - Θ	Χειμερινό 2011	Εισαγωγή στην πληροφορική - Τ1
Προγραμματισμός Η/Υ I - Ε	Εαρινό 2012	Εισαγωγή στην πληροφορική - Τ1
Διαδικτυακές Υπηρεσίες - Θ	Εαρινό 2012	Εισαγωγή στην πληροφορική - Τ1
Διαδικτυακές Υπηρεσίες - Ε	Εαρινό 2012	Εισαγωγή στην πληροφορική - Τ1
Δεξιότητες Επικοινωνίας - Θ	Εαρινό 2012	Εισαγωγή στην πληροφορική - Τ1
Δεξιότητες Επικοινωνίας - Ε	Εαρινό 2012	Εισαγωγή στην πληροφορική - Τ1

Επίσης κάτι που δεν αναφέρθηκε είναι ότι, κατά την δημιουργία των εγγραφών για τον πίνακα **Semester_courses**, τιμή για το πεδίο **Semester** χρησιμοποιείται η τιμή του τελευταίου εξαμήνου (πίνακας Semesters) που υπάρχει και που θεωρητικά είναι και αυτό για το οποίο καταχωρούνται τα μαθήματα τα οποία θα διεξάγονται.

Μια ακόμη και τελευταία δυνατότητα που έχουν οι χρήστες με ρόλο διαχειριστή είναι ότι, στο front-end κομμάτι, δηλαδή στην προβολή ημερολογίου, έχουν την δυνατότητα να προσθέσουν – ενημερώσουν – διαγράψουν όποια κράτηση θέλουν χωρίς να έχουν περιορισμό μόνο για τις δικές τους κρατήσεις, πράγμα που έχουν οι χρήστες με ρόλο καθηγητή.

Κεφάλαιο 7 Παράρτημα

Σε αυτό το κεφάλαιο θα παρουσιάσουμε κομμάτια κώδικα και μεθόδους, αναλύοντας την λειτουργικότητα τους και πως έχουν χρησιμοποιηθεί στην παρούσα πτυχιακή εργασία.

7.1 Δημιουργία αντικειμένου JSON στον server.

```

49 function feachHalls($link, $log){
50     $count=0;
51     $sql = " SELECT id, name FROM halls";
52     // Execute the query
53     $result = mysql_query($sql, $link);
54     if (!$result) {
55         $log->write('feachHalls(). Fail quering the database. MySQL Error: ' . mysql_error());
56         exit;
57     } else {
58         // Build the array
59         $options = array();
60         $halls = array();
61         while ($entry = mysql_fetch_assoc($result)) {
62             $count++;
63             // This format is used for the DataTable.
64             $options = array(
65                 'DT_RowId' => $entry['id'],
66                 '0' => $entry['id'],
67                 '1' => $entry['name']
68             );
69             $halls['sEcho'] = 1;
70             $halls['iTotalRecords'] = $count;
71             $halls['iTotalDisplayRecords'] = $count;
72             $halls['aaData'][] = $options;
73         }
74         //close connection
75         mysql_close($link);
76         // encode into json, and return it
77         echo json_encode($halls);
78     }
79     //close connection
80     mysql_close($link); exit;
81 }

```

Όταν ένα script στον διακομιστή καλείται με σκοπό να επιστρέψει κάποια δεδομένα, τότε το script «κτυπά» την βάση δεδομένων (γραμμή 53). Έτσι, ο χρήσης παίρνει τα αποτελέσματα (γραμμή 61), τα κτίζει σε ένα php array προσθέτοντας κάποια επιπλέον δεδομένα που χρειάζεται ο client (γραμμές 64-72). Στη συνέχεια, με την χρήση μιας build-in (ενσωματωμένης) μεθόδου της php, μετατρέπει αυτόν τον πίνακα σε αντικείμενο JSON και επιστρέφει τον client (γραμμή 77), ο οποίος το αποκωδικοποιεί και διατρέπει το αντικείμενο διαβάζοντας τα δεδομένα. (εικόνα κεφ. 3.4.4 Παράδειγμα 1 – Γέμισμα drop down στοιχείου).

7.2 Μέθοδοι επαλήθευσης ημερομηνίας, url και email.

Πέραν από τον client site κώδικα που έχει γραφτεί για την επαλήθευση (validation) των δεδομένων που εισάγει ο χρήστης, πρέπει να γράφεται και ο αντίστοιχος server site επειδή ποτέ δεν είναι ασφαλής ο web. Για τις επαληθεύσεις αυτές των ημερομηνιών και των διευθύνσεων URLs χρησιμοποιήθηκαν κανονικές εκφράσεις, που έβρισκαν σωστό μόνο ό,τι συμφωνούσε με το πρότυπο που καθορίστηκε. Για την επαλήθευση των διευθύνσεων ηλεκτρονικού ταχυδρομείου χρησιμοποιήθηκε ενσωματωμένη μέθοδος της PHP.

```

216 function validateEmail($email){
217     if(filter_var($email, FILTER_VALIDATE_EMAIL)) {
218         return true;
219     } else {
220         return false;
221     }
222 }
223
224 /**
225  * Validates a URL using the PHPs filter_var
226  * @param $url
227  */
228 function validateWebsite($url){
229     if(filter_var(preg_match('%^((https?://)|(www\.)+)([a-z0-9-]?)+(:[0-9]+)?(/.*)?$%i', $url))) {
230         return true;
231     } else {
232         return false;
233     }
234 }
235
236 /**
237  * Checks if the param its a date..
238  * @param date $date
239  */
240 if (!function_exists('validateDates')) {
241     function validateDates($date){
242         // Makes sure things like 1998-13-32 won't get past and validate.
243         if ( preg_match('/^[0-9]{4}-(0*[1-9]|1[0-2])-(0[1-9]|[1-2][0-9]|3[0-1])$/', $date) ) {
244             list($year , $month , $day) = explode('-', $date);
245             return checkdate($month , $day , $year);
246         } else {
247             echo "Date's format invalid.<br /> ";
248         }
249     }
250 }
251 }

```

7.3 Διαδικασία δημιουργίας ενός Datatable.

Όπως έχει αναφερθεί και σε προηγούμενο κεφάλαιο, το Datatable είναι ένα plugin, έτοιμος κώδικας, τον οποίο, εάν χρησιμοποιηθεί σωστά θέτοντας τις κατάλληλες παραμέτρους, μπορούν να δημιουργηθούν όμορφοι και δυναμικοί πίνακες.

Στην προκείμενη περίπτωση, τα Datatables θα δημιουργούνται δίνοντας ένα url, το οποίο επιστρέφει τα αποτελέσματα σε ένα αντικείμενο μορφής JSON, από το οποίο αντικείμενο εξάγει τα δεδομένα και τα τοποθετεί όπως θα του υποδείξει ο χρήστης μέσα από τις παραμέτρους και θα δώσει τις τιμές σε **class** και **id** που θα του πει ο χρήστης. Επίσης, ο χρήστης θέτει και τις HTTP μεθόδους που θα χρησιμοποιήσει (ποιος??) για να πάρει και να στείλει δεδομένα από το διακομιστή για δημιουργία, διαγραφή και ενημέρωση γραμμής, όπως και τι εικόνες που θα χρησιμοποιήσει και ποια μηνύματα θα δείχνει.

Πτυχιακή Εργασία του Χαραλάμπους Μάριου

```
213 /**
214  * When the DOM is ready, create the table and use ajax to fill it (Courses).
215  */
216 $(function(){
217     $(document).ready( function () {
218         $('#show-courses').append('<div class="add_delete_toolbar" />');
219         $('#show-courses').append('<table cellpadding="0" cellspacing="0" border="0" ' +
220             'class="display" id="coursesTable"> <thead> <tr> <td> ID </td> <td> Code </td> ' +
221             '<td> Name </td> <td> Type </td> <td> Semester </td> <td> Multitude </td> </tr> ' +
222             '</thead> <tbody> </tbody> </table>');
223         var id = -1; //simulation of id
224         coursesDataTable = $('#coursesTable').dataTable({
225             bJQueryUI: true,
226             "sPaginationType": "full_numbers",
227             "bProcessing": false,
228             "sAjaxSource": "./admin-courses-crud.php?action=fetchCourses",
229             "aoColumns": [{ "bVisible": false, sName:'id' },
230                 { sName: 'code' },
231                 { sName: 'name' },
232                 { sName: 'type' },
233                 { sName: 'sem_no' },
234                 { sName: 'multitude' }
235             ]
236         }).makeEditable({
237             sAddURL:      "./admin-courses-crud.php?action=addCourse",
238             sUpdateURL:  "./admin-courses-crud.php?action=updateCourse",
239             sDeleteURL:  "./admin-courses-crud.php?action=deleteCourse",
240             sAddHttpMethod:  "POST",
241             sDeleteHttpMethod: "POST",
242             sUpdateHttpMethod: "POST",
```

Γραμμές 218-222 : δημιουργείται ο πίνακας με μόνο το **<head></head>** μέρος όπου καθορίζονται τα ονόματα των στηλών που θα υπάρχουν.

Γραμμές 226-227 : δηλώνεται ότι θα χρησιμοποιηθεί **jQuery theme**, έτοιμο css αρχείο για την μορφοποίηση του και ότι είναι επιθυμητό από τον χρήστη να δείχνει στην σελιδοποίηση όλους τους αριθμούς των σελίδων.

Γραμμή 228 : Καθορίζεται το URL από το οποίο θα πάρει τα δεδομένα για να συμπληρώσει τον πίνακα.

Γραμμές 229-235: Εδώ δηλώνεται η ιδιότητα όνομα που θα δώσει στο κάθε κελί και επιπλέον ότι δεν θέλουμε να είναι ορατό το πρώτο πεδίο που είναι το id.

Γραμμές 236-242 : Γίνεται χρήση μιας επέκτασης αυτού η οποία δίνει την δυνατότητα στον χρήστη, κάνοντας διπλό κλικ στα κελιά να μετατρέπονται στον κατάλληλο τύπο για αλλαγή των τιμές και πιο κάτω υποδεικνύονται τα URLs και οι μέθοδοι που θα χρησιμοποιεί όπου θα στέλνει δεδομένα για τις ενέργειες.

Πτυχιακή Εργασία του Χαραλάμπους Μάριου

```
243 "aoColumns": [  
244     {  
245         tooltip: 'Double click to change Code.',  
246         type: 'text',  
247         indicator: 'Saving...',  
248         loadtext: 'loading...',  
249         enter : 'submit'  
250     },  
251     {  
252         tooltip: 'Double click to change Name.',  
253         type: 'text',  
254         indicator: 'Saving...',  
255         enter : 'submit'  
256     },  
257     {  
258         tooltip: 'Double click to change Type.',  
259         type: 'select',  
260         indicator: 'Saving...',  
261         loadtext: 'loading...',  
262         data: "{ 'Θεωρία': 'Θεωρία', 'Εργαστήριο': 'Εργαστήριο', " +  
263             "'Παρουσίαση Πτυχιακής': 'Παρουσίαση Πτυχιακής', " +  
264             "'Συνεδρίαση': 'Συνεδρίαση', 'Άλλο': 'Άλλο'}",  
265         onblur : 'submit'  
266     },  
267     {  
268         tooltip: 'Double click to change Semester .',  
269         type: 'select',  
270         data: "{ '1': 'Α εξάμηνο', '2': 'Β εξάμηνο', " +  
271             "'3': 'Γ εξάμηνο', '4': 'Δ εξάμηνο', '5': 'Ε εξάμηνο', " +  
272             "'6': 'ΣΤ εξάμηνο', '7': 'Ζ εξάμηνο'}",  
273         indicator: 'Saving...',  
274         loadtext: 'loading...',  
275         onblur : 'submit'
```

Γραμμές 243-283 : Καθορίζονται οι τύποι που θα παίρνουν τα στοιχεία μετά το διπλό κλικ και επίσης το γεγονός που θα εκτελεί την αποστολή των δεδομένων, το κείμενο που θα δείχνει όταν περιμένει ο χρήστης να φορτώσουν τα δεδομένα σε περίπτωση που αυτά θα παρθούν από άλλο AJAX call (όπως το παράδειγμα της επόμενης εικόνας), τι τύπου θα είναι και ποιες τιμή/τιμές, καθώς και το tooltip κείμενο που είναι το κείμενο που θα φαίνεται/εμφανίζεται με το πέρασμα του ποντικιού πάνω από το πεδίο.

```
"aoColumns": [  
    {  
        type: 'select',  
        tooltip: 'Click to change value.',  
        loadurl : './admin-semester_courses-crud.php?action=getCourses',  
        loadtype : 'POST',  
        enter : 'submit',  
        indicator: 'Saving...',  
        loadtext: 'loading...'  
    },  
    {
```

```

277         {
278             tooltip: 'Double click to change Multitude.',
279             type: 'text',
280             indicator: 'Saving...',
281             enter : 'submit'
282         }
283     ],
284     oAddNewRowButtonOptions: {
285         label: "Add new...",
286         icons: {primary:'ui-icon-plus'}
287     },
288     oDeleteRowButtonOptions: {
289         label: "Remove",
290         icons: {primary:'ui-icon-trash'}
291     },
292     oAddNewRowFormOptions: {
293         title: 'Add a new Course',
294         show: "blind",
295         hide: "explode",
296         modal: true
297     }
298     ,
299     sAddDeleteToolbarSelector: ".dataTables_length" ,
300     fnStartProcessingMode: function () {
301         $("#processing_message").dialog();
302     },
303     fnEndProcessingMode: function () {
304         $("#processing_message").dialog("close");
305         coursesDataTable.fnReloadAjax();
306     }
307     });
308 });

```

Γραμμές 284-291 : Καθορίζεται το κείμενο και οι εικόνες που θα υπάρχουν στα δύο κουμπιά, εισαγωγής και διαγραφής γραμμών.

Γραμμή 298 : Καθορίζεται η κλάση (class) που θα πάρει το toolbar στο πάνω μέρος του Datatable.

Και τέλος γραμμές 299-304 : Καθορίζονται οι μέθοδοι που θέλει ο χρήστης να ενεργοποιηθούν στην αρχή και στο τέλος της λειτουργίας επεξεργασίας (processing mode).

7.4 Διαδικασία προετοιμασίας δεδομένων το ημερολόγιο.

Σε αυτό το υποκεφάλαιο θα παρουσιαστούν μερικές μέθοδοι (functions) οι οποίες έχουν δημιουργηθεί με σκοπό να υπάρχει επικοινωνία μεταξύ πελάτη-διακομιστή (client-server), όπου επιστρέφει τα δεδομένα που χρειάζονται στην μορφή που τα θέλει ο χρήστης, έτσι ώστε, διασχίζοντας τα δεδομένα αυτά, τα στοιχεία να τοποθετούνται σε ένα τοπικό πίνακα (javascript array) με τιμή κλειδί κάθε θέσης του πίνακα αυτού το “id” του κελιού (<td>) στο ημερολόγιο που είναι να καταχωρηθεί.


```

573 /**
574  * General function that fills the weekly calendar with the days
575  * returned from the server.
576  */
577
578 function buildGeneralCalendarArray(searchFor){
579     // Re-initialize the table so there will be no previous data in
580     var i=1, x=0, y=0, deffHourse = 0, selectedOption;
581     generalArray = Array();
582     if(searchFor=='Hall'){
583         var selectedOption=$("#select#halls option:selected").val();
584         var dataString = "startDate="+varStartDate+"&endDate="+varEndDate+"&hall="+selectedOption;
585         postUrl='getByHall.php';
586     } else if(searchFor=='Professor'){
587         var selectedOption = $("#select#professors option:selected").val();
588         var dataString = "startDate="+varStartDate+"&endDate="+varEndDate+"&professor="+selectedOption;
589         postUrl='getByProfessor.php';
590     } else if(searchFor=='Course'){
591         var selectedOption = $("#select#courses option:selected").val();
592         var dataString = "startDate="+varStartDate+"&endDate="+varEndDate+"&course="+selectedOption;
593         postUrl='getByCourse.php';
594     }
595
596     $.ajax({
597         type : "POST",
598         url : postUrl,
599         data : dataString,
600         datatype : "json",
601         success:( function(json){
602             //decode json data
603             var json = JSON.parse(json);
604             // If there are objects means that the professors has reservations on this week
605             if(json.objects){

```

Στις γραμμές 576-593 φαίνεται η δήλωση της μεθόδου με μια παράμετρο, η οποία λέει το πρόγραμμα ποιαν από τις τρεις επιλογές καθηγητή/αίθουσα/μάθημα έχει επιλέξει ο χρήστης για να δει. Αρχικοποιούνται κάποιες μεταβλητές και η παράμετρος αυτή περνιέται από τρεις εκφράσεις “if”, μια για κάθε επιλογή, και έτσι δημιουργείται ένα αλφαριθμητικό (string) που θα αναπαριστά τα δεδομένα που θα στέλνονται με την χρήση AJAX στον διακομιστή και στο κατάλληλο PHP σενάριο.

Στη γραμμή 596 φαίνεται πώς καλείται η μέθοδος της jQuery βιβλιοθήκης που θα στείλει τα δεδομένα και ακολούθως στις γραμμές 597-600 δηλώνεται ο τύπος της μεθόδου που θα χρησιμοποιηθεί για την αποστολή, το URL, το αλφαριθμητικό που κρατά τα δεδομένα και τον τύπο των αποτελεσμάτων που θα επιστραφούν από τον διακομιστή.

Από την γραμμή 601 μέχρι και την 637 χρησιμοποιείται η απάντηση από τον διακομιστή, αποκωδικοποιείται με την `JSON.parse()`. Εάν το αποτέλεσμα είναι ένα αντικείμενο JSON, σημαίνει πως επιλέγηκαν τα επιθυμητά από τον χρήστη δεδομένα και ο διακομιστής διατρέχει τα στοιχεία του κάνοντας χρήση κάποιων μικρών απλών μεθόδων που έχουν καταγραφεί (αμέσως μετά εικόνα) δημιουργώντας έτσι τον JS array που κρατά πλέον τα στοιχεία της κάθε δέσμευσης.

Στη συνέχεια όπως είναι εμφανές, καθορίζουμε τι θα τρέξει σε περίπτωση που υπάρξει κάποιο σφάλμα κατά την εκτέλεση της AJAX κλήσης και τι θα τρέξει όταν ολοκληρωθεί.

Στην περίπτωση που δεν θα υπάρξει κάποιο σφάλμα, καλείται η μέθοδος που συμπληρώνει το ημερολόγιο, θέτει σε δύο καθολικές μεταβλητές (global variables) τις τιμές που έχουν χρησιμοποιηθεί και τέλος, το σύστημα ελέγχει εάν είναι κάποιος καθηγητής ή διαχειριστής συνδεδεμένος για να προσθέσει την επιλογή της εισαγωγής δεσμεύσεως στο ημερολόγιο.

Πτυχιακή Εργασία του Χαραλάμπους Μάριου

```
606     $.each(json.objects, function(key, value) {
607         var day, x=0, y=0, generalReturned = Array();
608         $.each(json.objects[key], function(inKey, inValue){
609             generalReturned[inKey] = inValue;
610             if(inKey=="day"){
611                 day = inValue;
612             }
613         });
614         // Get the int for this day (eg. 1,2,3,4,5,6,7)
615         x = _DayInInt(day);
616         durSt = ((generalReturned['start_time'].substr(0,2)-8)/2);
617         if(generalReturned['start_time'].substr(3,5)=='30'){
618             durSt+=0.5;
619         }
620         durEt = (generalReturned['end_time'].substr(0,2)-8)/2;
621         if(generalReturned['end_time'].substr(3,5)=='30'){
622             durEt+=0.5;
623         }
624         y = _CellForHour(durSt);
625         i = y+x;
626         while(durEt>durSt){
627             // If there is nothing in this 'coursesArray[i]', create a new array();
628             if(!generalArray[i]){
629                 generalArray[i] = Array();
630             }
631             generalArray[i][generalArray[i].length] = generalReturned;
632             durEt-=1;
633             i+=7;
634         }
635     });
636 }
637 },
638 error: (function() { alert("Build calendar Ajax error"); }),
639 complete: (function() {
640     fillCalendar(generalArray);
641     // Set that the user is watching the calendar field data according a selecte Hall
642     resultsFor = searchFor;
643     resultsForValue = selectedOption;
644     if(role == "professor" || role == "administrator"){
645         addTheAddButtons(resultsFor);
646     }
647 })
648 });
649 }
```

Μέθοδοι που επιστρέφουν αριθμό ημέρας και κελί για την ώρα

```
/**
 * _CellForHour(startHour)
 *
 * Return the cell number that will be used to
 * place the reservation
 * @param: hour (eg. 8,9,10,11,12,13,.....,19).
 */
function _CellForHour(startHour){
    if(startHour<1){
        return 0;
    }else if(startHour<2){
        return 7;
    }else if(startHour<3){
        return 14;
    }else if(startHour<4){
        return 21;
    }else if(startHour<5){
        return 28;
    }else if(startHour){
        return 35;
    }
}
```

```
/**
 * _DayInInt(day)
 *
 * Returns the nubder of the day from 1 to 7.
 * @param: day. eg. Monday, Tuesday, ..... Sunday.
 */
function _DayInInt(day){
    switch(day)
    {
        case "Monday":    return 1;
        case "Tuesday":   return 2;
        case "Wednesday": return 3;
        case "Thursday":  return 4;
        case "Friday":    return 5;
        case "Saturday":  return 6;
        case "Sunday":    return 7;
    }
}
```

7.5 Εμφάνιση δεδομένων στο ημερολόγιο.

Σε αυτό το υποκεφάλαιο θα παρουσιαστούν και θα επεξηγηθούν η μέθοδος που καλείται, αφού έχει τελειώσει η προετοιμασία του πίνακα (JS array) με τα δεδομένα των δεσμεύσεων και λίγο πριν τερματίσει η προηγούμενη μέθοδος.

Μέθοδος που τοποθετεί τα δεδομένα στο ημερολόγιο

```

/**
 * The function that will fill the calendar table with the data
 * from the coursesArray array.
 */
function fillCalendar(theArray){
  //For each <td> in the calendar.php
  $('td').each(function(){
    thisId = $(this).attr('id'); //get the id of it.
    thisClass = $(this).attr('class'); //and the class of it.
    var val = "";
    // Make sure that the time <td> wont disappear
    if(thisId != 'time'){
      // Make sure that the other form td will not disappear
      if(thisClass!='fillFormTd' && thisClass!='add_fillFormTd' &&
        thisClass!='fillFormTd-1' && thisClass!='add_fillFormTd-1'){
        $(this).text(""); // Set the value of the cell equals null.
        // and then fill it with reservations if they exist.
        if(theArray[thisId] != null){
          // build on string that has all td's content with reservations
          for(i=0; i<theArray[thisId].length; i++){
            val = val+ "<div class='courseCell' id='"+thisId+"."+i+"' >" +
              theArray[thisId][i]['semester_course'] + " / " +
              theArray[thisId][i]['start_time'] + " - " + theArray[thisId][i]['end_time'] +
              " / " + theArray[thisId][i]['professor'] +
              " / " + theArray[thisId][i]['hall'] + "</div>";
            if((i+1) != theArray[thisId].length)
              { val = val + " <hr /> "; }
            //Replce the content of the td with the string built.
            $(this).html(val);
          }
        }
      }
    }
  });
}

```

Βιβλιογραφία

1. <http://download.eclipse.org/eclipse/downloads/>. Ιστοσελίδα για την ανάκτηση του eclipse
2. <http://www.eclipse.org/pdt/install-old.php>. PHP Development Tools (PDT) - Old Installation Instructions
3. <http://www.eclipse.org/pdt/downloads/>. Ιστοσελίδα για την ανάκτηση του PDT (PHP development tool)
4. http://www.phpmyadmin.net/home_page/index.php. Ιστοσελίδα για την ανάκτηση της MySql.
5. <http://www.apachefriends.org/en/xampp.html>. Ιστοσελίδα για την ανάκτηση του xampp.
6. <http://www.tizag.com/cssT/external.php>. «How to» use external css.
7. <http://www.w3.org/Style/CSS/Overview.en.html>. CSS tutorial starting with HTML + CSS
8. <http://www.w3.org/TR/CSS2/selector.html>. Css selectors.
9. <http://www.quirksmode.org/js/intro.html>. General introduction στην JavaScript
10. <http://en.wikipedia.org/wiki/JavaScript>. Αγγλική *Wikipedia* για Javascript.
11. <http://users.sch.gr/gkapog/sites/default/files/Javascript.pdf>. KENTPO ΠΛΗ.ΝΕ.Τ. Ν. ΦΛΩΡΙΝΑΣ. Η Γλώσσα Προγραμματισμού JavaScript.
12. http://docs.jquery.com/Downloading_jQuery. Ιστοσελίδα για κατέβασμα της βιβλιοθήκης jQuery.
13. http://docs.jquery.com/How_jQuery_Works. Πως λειτουργεί η βιβλιοθήκη.
14. <http://pietschsoft.com/post/2006/06/01/Javascript-Loop-through-all-elements-in-a-form.aspx>
15. http://books.google.com.cy/books?id=2T7F7AslMukC&printsec=frontcover&dq=php&hl=en&sa=X&ei=Lf6bT5TUFOam0AWclrz2Dg&redir_esc=y#v=onepage&q&f=true. Beginning PHP and MySql. From Novice to Professional. Fourth edition. W. Jason Gilmore.
16. http://books.google.com.cy/books?id=iBPa11350b4C&printsec=frontcover&dq=ajax+php&hl=en&sa=X&ei=If2bT8XxH6fV0QXgupGZDw&redir_esc=y#v=onepage&q=ajax%20php&f=false. Beginning with AJAX and PHP. From Novice to Professional. Lee Babin.
17. http://www.w3schools.com/webservices/ws_intro.asp

18. <http://sixrevisions.com/javascript/the-power-of-jquery-with-ajax/>
19. <http://community.topcoder.com/tc?module=Static&d1=features&d2=071706>
20. <http://www.json.org/json-el.html>. JSON home page.
21. <http://www.ituts.gr/about-json>. Εισαγωγή στην JSON και σύγκριση με XML.
22. <http://www.ironspeed.com/articles/ajax-bridging%20the%20thin-client%20performance%20gap/article.aspx>.
23. <http://i.technet.microsoft.com/dynimg/IC210892.gif>
24. http://www.webopedia.com/TERM/C/client_server_architecture.html
25. <http://conferences.ellak.gr/2011/2011/05/10/%CE%B1%CF%83%CF%86%CE%AC%CE%BB%CE%B5%CE%B9%CE%B1-web-%CE%B5%CF%86%CE%B1%CF%81%CE%BC%CE%BF%CE%B3%CF%8E%CE%BD-2/>. Συνέδριο ΕΛ/ΛΑΚ 2011 με θέμα «Ασφάλεια Web Εφαρμογών»
26. http://en.wikipedia.org/wiki/List_of_applications_with_iCalendar_support
27. http://en.wikipedia.org/wiki/Rainbow_table. *Wikipedia* Rainbow tables.
28. http://en.wikipedia.org/wiki/Brute-force_attack. *Wikipedia* Brute force attack.
29. http://en.wikipedia.org/wiki/Denial_of_service. *Wikipedia* Denial of service.
30. http://en.wikipedia.org/wiki/Sql_injection. *Wikipedia* Sql injection.
31. <http://vlm1.uta.edu/~akotsif/ELF22-OWASP.pdf>
32. Μελέτη στην τεχνολογία WebServices μέσω JavaScript. ΤΥΜΠΑΚΙΑΝΑΚΗΣ ΑΠΟΣΤΟΛΟΣ / ΗΡΑΚΛΕΙΟ ΑΠΡΙΛΙΟΣ 2010.
33. http://en.wikipedia.org/wiki/Document_Object_Model. *Wikipedia* Document Object Model.
34. <http://www.w3.org/DOM/> W3C Document Object Model (DOM).
35. http://web-seminar.softlab.ntua.gr/slides/18_javascriptdom.pdf. Διαφάνειες Μαθήματος Διδάσκων Π. Αγγελάτος, Δ. Ζηνδρος Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Ηλεκτρονικών Υπολογιστών DOM Σχεδίαση διαδραστικού διαδικτυακού ιστότοπου για το Εργαστήριο Υψηλών Τάσεων του Αριστοτελείου Πανεπιστημίου Θεσσαλονίκης. Γεωργία Καλλονά.
36. Programming and human factors, ένα εργαλείο για rainbow hash cracking. <http://www.codinghorror.com/blog/2007/09/rainbow-hash-cracking.html>
37. <http://stackoverflow.com/questions/110575/do-htmlescape-specialchars-and-mysql-real-escape-string-keep-my-php-code-safe-from-injection>. Συζήτηση από forum.

38. https://www.owasp.org/index.php/Denial_of_Service. Denial of Service, OWASP wiki page.
39. <http://datatables.net/>. Κύρια ιστοσελίδα του plug-in .
40. AJAX and the Future of Web Applications
41. <http://api.jquery.com/jquery.ajax/>. jQuery online api, περιγραφή μεθόδου jQuery.ajax().