



Τμήμα Μηχανικών
Πληροφορικής ΑΤΕΙΘ

ΑΛΕΞΑΝΔΡΕΙΟ Τ.Ε.Ι. ΘΕΣΣΑΛΟΝΙΚΗΣ

ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΜΕΤΑΦΟΡΤΩΣΗ ΠΤΥΧΙΑΚΩΝ ΕΡΓΑΣΙΩΝ



**Του φοιτητή
Ματούλα Θωμά**

**Επιβλέπων καθηγητής
Αμανατιάδης Δημήτριος**

Αρ. Μητρώου: 08/3317

Θεσσαλονίκη 2014

Πρόλογος

Στόχος της πτυχιακής αυτής είναι ουσιαστικά η παρουσίαση της διαδικασίας μεταφόρτωσης ενός website ή ενός ολοκληρωμένου διαδικτυακού πληροφοριακού συστήματος, είτε από έναν Server σε έναν άλλον, είτε στον ίδιο Server από μία τοποθεσία σε μία άλλη.

Είναι γεγονός ότι οι φοιτητές συνήθως ολοκληρώνουν την πτυχιακή τους στο χώρο του προσωπικού τους λογαριασμού. Δυστυχώς πολλές από αυτές χάνονται μαζί με την διαγραφή του λογαριασμού του χρήστη (απόφοιτου πλέον) μετά από ένα χρόνο από την ορκωμοσία του. Η αρχική σκέψη ήταν για τη μεταφόρτωση τριών Πτυχιακών Εργασιών αλλά λόγω χρονικών περιορισμών και διαφορετικότητας της τρίτης πτυχιακής, η παρούσα πτυχιακή εργασία έχει τελικά τη μεταφόρτωση δύο πτυχιακών εργασιών. Η πρώτη αφορούσε αποκλειστικά και μόνο τη μεταφορά της πτυχιακής σαν τελικό προορισμό σε ένα φάκελο του προσωπικού λογαριασμού του επιβλέποντα καθηγητή.

Όσον αφορά για τη δεύτερη πτυχιακή, ζητήθηκε εκτός από τη μεταφόρτωση να γίνει και η βελτίωση και επέκτασή της σε ένα ολοκληρωμένο διαδικτυακό πληροφοριακό σύστημα, που θα αποτελεί ένα εργαλείο που θα δίνει τη δυνατότητα να διατηρείται online ιστορικό παρακολούθησης αναθέσεων πτυχιακών εργασιών.

Οι πτυχιακές για τις οποίες γίνεται λόγος είναι:

1. Οδηγός δημιουργίας ιστοσελίδων με λογισμικό ανοιχτού κώδικα (Χατζής Αντώνιος).
2. Σύστημα επίβλεψης Πτυχιακών Εργασιών (Παλάσχος Κωνσταντίνος).

Περίληψη

Σε αυτήν την πτυχιακή εργασία παρουσιάζεται η διαδικασία μεταφόρτωσης δύο πτυχιακών εργασιών, του φοιτητή Χατζή Αντωνίου με τίτλο “Οδηγός δημιουργίας ιστοσελίδων με λογισμικό ανοιχτού κώδικα”, και του φοιτητή Παλάσχα Κωνσταντίνου με τίτλο “Σύστημα επίβλεψης Πτυχιακών Εργασιών”. Πρόκειται για δύο αξιόλογες πτυχιακές που ο επιβλέπων καθηγητής ήθελε να διατηρήσει την ύπαρξή τους όταν οι φοιτητές αυτοί θα διαγραφούν από τον server της σχολής. Για το λόγο αυτό έγινε η μεταφόρτωσή τους από την τοποθεσία στην οποία βρίσκονταν τα αρχεία των πτυχιακών (προσωπικός λογαριασμός φοιτητών) στον λογαριασμό του επιβλέποντα καθηγητή. Στο παράρτημα παρουσιάζονται και περιγράφονται όλα τα βήματα που ακολουθήθηκαν, ώστε διαβάζοντας κάποιος αυτήν την πτυχιακή να είναι σε θέση μεταφορτώσει παρόμοιες πτυχιακές εργασίες.

Abstract

This dissertation paper presents the transshipment procedure of two other dissertations. The first comes from the student Hatzis Antonios, entitled "Website with open source software creation guide" and the other from Palashas Konstantinos, entitled "Dissertations supervision system". The dissertations mentioned above, are two notable papers, which the supervising professor wanted to preserve when these students get erased from the school server. For this reason they were transhiped from the location in which the files of the papers were located (the students' personal account), to the supervising professor's account. In the annex, all the steps followed are presented and described, so that someone who reads this dissertation will be able to tranship similar papers.

...Ένα μεγάλο ευχαριστώ στον επιβλέποντα καθηγητή μου κύριο Αμανατιάδη Δημήτριο για την συνεργασία που είχαμε τους τελευταίους μήνες, αλλά και για την πολύτιμη βοήθεια και την καθοδήγηση που μου έδωσε για να αντιμετωπιστούν τα διάφορα προβλήματα που προέκυψαν κατά τη διάρκεια της πτυχιακής, ώστε να επιτύχουμε το καλύτερο δυνατό αποτέλεσμα. Επίσης θεωρώ υποχρέωσή μου να ευχαριστήσω τους γονείς μου, διότι χωρίς τη στήριξή τους και τη βοήθειά τους δε θα είχα τη δυνατότητα να σπουδάσω αυτό που πάντα ήθελα....

Πίνακας Περιεχομένων

ΠΡΟΛΟΓΟΣ	2
ΠΕΡΙΛΗΨΗ.....	3
ABSTRACT.....	4
ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ	6
ΕΙΣΑΓΩΓΗ	9

ΜΕΡΟΣ Α

ΚΕΦΑΛΑΙΟ 1 ΜΙΑ ΣΥΝΤΟΜΗ ΙΣΤΟΡΙΑ ΤΟΥ ΔΙΑΔΙΚΤΥΟΥ

1.1 Το Διαδίκτυο	11
1.2 Το δίκτυο ARPANET	11
1.3 Ο Παγκόσμιος Ιστός (World Wide Web).....	12

ΚΕΦΑΛΑΙΟ 2 ΔΙΑΔΙΚΤΥΑΚΟΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ

2.1 Ανάπτυξη διαδικτυακών εφαρμογών.....	13
2.2 Ιστορία Διαδικτυακού Προγραμματισμού	13
2.2.1 Στατικό περιεχόμενο	14
2.2.2 Δυναμικό περιεχόμενο Gateway Programming.....	14
2.2.3 Client Side Scripting.....	15
2.2.4 Server Side Scripting.....	16
2.3 Αρχιτεκτονική Client/Server.....	17
2.4 Client Side Προγραμματισμός	18
2.4.1 HTML	18
2.4.2 XHTML	20

2.4.3 CSS.....	22
2.4.5 JavaScript.....	23
2.4.6 JQuery	24
2.5 Server Side Προγραμματισμός.....	25
2.5.1 PHP.....	26
2.5.3 Perl.....	29
2.5.4 MySQL	30
2.5.5 Postgress	31
2.6 Τεχνολογία Ajax.....	32
2.7 Μεταφορά δεδομένων μεταξύ διαδικτυακών εφαρμογών	33
2.7.1 Χρήση XML	33
2.7.2 Χρήση JSON	33

ΚΕΦΑΛΑΙΟ 3 ΣΥΣΤΗΜΑΤΑ ΔΙΑΧΕΙΡΙΣΗΣ ΠΕΡΙΕΧΟΜΕΝΟΥ (CMS)

3.1 Σύστημα διαχείρισης περιεχομένου.....	35
3.2 Ιστορικά.....	35
3.3 Γνωστά συστήματα διαχείρισης περιεχομένου	35

ΜΕΡΟΣ Β

ΚΕΦΑΛΑΙΟ 4 ΜΕΤΑΦΟΡΤΩΣΗ ΠΡΩΤΗΣ ΠΤΥΧΙΑΚΗΣ

4.1 Περιγραφή πτυχιακής	39
4.2 Διαδικασία Μεταφόρτωσης	39
4.3 Ολοκλήρωση Μεταφόρτωσης	46
4.4 Προβλήματα που παρουσιάστηκαν	47

ΚΕΦΑΛΑΙΟ 5 ΜΕΤΑΦΟΡΤΩΣΗ ΔΕΥΤΕΡΗΣ ΠΤΥΧΙΑΚΗΣ

5.1 Περιγραφή Πτυχιακής	52
5.2 Δημιουργία της διαδικτυακής βάσης	53
5.3 Ενημέρωση της διαδικτυακής βάσης.....	58
5.4 Σχεδίαση της διαδικτυακής διεπαφής.....	59
5.5 Υλοποίηση Συστήματος.....	66
5.5.1 Ολοκλήρωση Συστήματος	66
5.6 Μεταφόρτωση πτυχιακής	68
5.7 Προβλήματα που αντιμετωπίστηκαν	69

ΜΕΡΟΣ Γ

ΠΑΡΑΡΤΗΜΑ.....	73
6.1 Δημιουργία προσωπικής ιστοσελίδας στον Server (aeto).....	73
6.2 Χρήση Mysql.....	74
6.3 Πτυχιακή Χατζή.....	76
6.4 Πτυχιακή Παλάσχα	77
6.5 Αξιοσημείωτα Css και JQuery-PHP κώδικας που χρησιμοποιήθηκε	81
ΒΙΒΛΙΟΓΡΑΦΙΑ	84

Εισαγωγή

Έχει παρατηρηθεί πως πολλές από τις πτυχιακές που υλοποιούνται από τους φοιτητές είναι αρκετά καλές και λειτουργούν σαν εργαλείο για τον επιβλέποντα καθηγητή. Ωστόσο, το γεγονός ότι οι περισσότερες από αυτές δημιουργούνται στον προσωπικό λογαριασμό των φοιτητών οδηγεί στην αυτόματη διαγραφή τους, όταν οι φοιτητές αυτοί πάρουν πτυχίο και σβηστούν από το server της σχολής. Σκοπός αυτής της πτυχιακής είναι η μεταφόρτωση συγκεκριμένων πτυχιακών σε μια άλλη τοποθεσία (λογαριασμός επιβλέποντα καθηγητή) ώστε να μη χαθούν και να συνεχίσουν να λειτουργούν ως εργαλείο είτε για τον ίδιο τον επιβλέποντα, είτε για άλλους φοιτητές του τμήματος.

Η παρούσα πτυχιακή αποτελείται από τέσσερα κεφάλαια και το παράρτημα.

Στο πρώτο κεφάλαιο γίνεται μια εισαγωγή στον παγκόσμιο ιστό καθώς και μια μικρή αναφορά στην ιστορία του. Στο δεύτερο κεφάλαιο υπάρχει μια ανάλυση της βασικής φιλοσοφίας που επικρατεί στον διαδικτυακό προγραμματισμό καθώς και στις πιο ευρέως χρησιμοποιούμενες γλώσσες προγραμματισμού. Παρουσιάζονται αναλυτικά τα στάδια τα οποία πέρασαν οι βασικές γλώσσες του διαδικτύου. Το τρίτο κεφάλαιο περιέχει μια περιγραφή των πιο γνωστών συστημάτων διαχείρισης περιεχομένου (CMS).

Το τρίτο και το τέταρτο κεφάλαιο αφορούν το πρακτικό κομμάτι της πτυχιακής, τη μεταφόρτωση των δύο πτυχιακών. Πριν από την ανάλυση των βημάτων υπάρχει μια μικρή εισαγωγή για την κάθε πτυχιακή. Στη συνέχεια ακολουθεί η ανάλυση των βημάτων που ακολουθήθηκαν και η περιγραφή των προβλημάτων που συναντήθηκαν καθώς και πως αντιμετωπίστηκαν. Τέλος υπάρχει ένα συμπέρασμα για το που αποσκοπεί η κάθε μεταφόρτωση. Να σημειωθεί πως μεγάλο κομμάτι της παρούσας πτυχιακής αφορά την επέκταση της δεύτερης εργασίας, γεγονός που περιγράφεται αναλυτικά πριν τα βήματα της μεταφόρτωσής της.

Τέλος υπάρχει το παράρτημα όπου υπάρχουν όλες οι οδηγίες για το πως μπορεί να συνδεθεί κάποιος στον server της σχολής αλλά και να χρησιμοποιήσει προσωπική Mysql. Υπάρχουν κάποια κομμάτια κώδικα που

χρησιμοποιήθηκαν για τις μεταφορτώσεις των πτυχιακών, καθώς και η επεξήγησή τους. Επίσης στο παράρτημα βρίσκονται και επιπλέον κομμάτια κώδικα που χρησιμοποιούνται ευρέως και αξίζουν να αναφερθούν καθώς και χρήσιμες οδηγίες όσον αφορά τις μεταφορτώσεις.

ΜΕΡΟΣ Α

Κεφάλαιο 1

Μια σύντομη ιστορία του Διαδικτύου

1.1 Το Διαδίκτυο

Η επικοινωνία μέσω Διαδικτύου αποτελεί πλέον αναπόσπαστο κομμάτι της ζωής μας. Στον παγκόσμιο ιστό βρίσκονται πληροφορίες για ποικίλες κατηγορίες θεμάτων, τις οποίες τις χρησιμοποιούμε σε μεγάλο βαθμό όλοι στην καθημερινότητά μας.

Θα μπορούσε κανείς να περιγράψει το **Διαδίκτυο** ως ένα παγκόσμιο δίκτυο από επιμέρους μικρότερα δίκτυα υπολογιστών, τα οποία επικοινωνούν μεταξύ τους με ένα σύνολο κοινών αποδεκτών πρωτοκόλλων (το βασικότερο εκ των οποίων είναι το **TCP/IP**) και περιλαμβάνει σχεδόν κάθε χώρα του πλανήτη μας. Με λίγα λόγια έχει σχεδιαστεί με τέτοιο τρόπο που μπορεί και ενσωματώνει πολλές διαφορετικές τεχνολογίες υλικού και με τη χρήση των πρωτοκόλλων που προαναφέρθηκαν, είναι εφικτή η συνεργασία μεταξύ των δικτύων αυτών.

1.2 Το δίκτυο ARPANET

Το Διαδίκτυο θεωρείται το ιστορικά ταχύτερα αναπτυσσόμενο δίκτυο και έχει επιφέρει πραγματική επανάσταση στον τρόπο επικοινωνίας τα τελευταία χρόνια σε παγκόσμια κλίμακα. Αν και έγινε ευρέως γνωστό κυρίως κατά την τελευταία δεκαετία, παρόλα αυτά οι βάσεις σχεδίασης και ανάπτυξης του τέθηκαν αρκετά παλαιότερα. Πρόγονος του σημερινού Διαδικτύου θεωρείται το δίκτυο **ARPANET**. Πρόκειται για ένα πειραματικό δίκτυο του υπουργείου άμυνας των ΗΠΑ, το οποίο άρχισε να αναπτύσσεται στα τέλη της δεκαετίας του 1960. Σκοπός του ήταν η διασύνδεση στρατιωτικών μονάδων με πανεπιστήμια με τα οποία πραγματοποιήθηκαν κοινά ερευνητικά στρατιωτικά προγράμματα.

1.3 Ο Παγκόσμιος Ιστός (*World Wide Web*)

Ο παγκόσμιος Ιστός (**World Wide Web** ή **www**) είναι σαφώς ή σημαντικότερη υπηρεσία του διαδικτύου. Είναι ένα κατανεμημένο σύστημα υπερμέσων που αποτελείται από τρία συστατικά μέρη, όπου το ένα συμπληρώνει το άλλο:

1. Το πρωτόκολλο **HTTP** (*HyperText Transfer Protocol*).
2. Το μηχανισμό προσδιορισμού διευθύνσεων **www** εξυπηρετητών (servers) **URL** (*Uniform Resource Location*).
3. Μία γλώσσα που ονομάζεται **HTML** (*HyperText Markup Language*).

Πατέρας του παγκόσμιου ιστού θεωρείται ο Βρετανός *Tim Berners-Lee*, οποίος εκείνη την εποχή (1989) εργαζόταν στον ευρωπαϊκό οργανισμό πυρηνικών ερευνών (CERN) στη Γενεύη της Ελβετίας. Ο Lee επινόησε το όνομα *World Wide Web*, έγραψε τον πρώτο εξυπηρετητή του παγκόσμιου ιστού “HyperText Transfer Protocol Daemon” (httpd), καθώς και το πρώτο πρόγραμμα χρηστών (web browser) “World Wide Web”. Επιπλέον δημιούργησε την πρώτη έκδοση της γλώσσας μορφοποίησης εγγράφου με τη δυνατότητα χρήσης υπερσυνδέσμων, HTML.

Κεφαλαίο 2

Διαδικτυακός προγραμματισμός

2.1 Ανάπτυξη διαδικτυακών εφαρμογών

Η ανάπτυξη **διαδικτυακών εφαρμογών** διαφέρει σε πολλά σημεία από την ανάπτυξη λογισμικού σε κλασικότερες μορφές όπως τη γνώριζαν οι επιστήμονες πληροφορικής μέχρι τώρα. Οι διαφορές μιας διαδικτυακής εφαρμογής σε σύγκριση με τις πιο παραδοσιακές μορφές, προκύπτουν κυρίως από το γεγονός ότι η πλατφόρμα, στην οποία γίνεται η εγκατάσταση και λειτουργεί η εφαρμογή, είναι το *διαδίκτυο (Internet)*. Επομένως λόγω των ιδιαίτερων χαρακτηριστικών που έχει το διαδίκτυο ως πλατφόρμα υποδομής, η ανάπτυξη διαδικτυακών εφαρμογών γίνεται αρκετά διαφορετική, ενδιαφέρουσα και ιδιαίτερη.

2.2 Ιστορία Διαδικτυακού Προγραμματισμού

Για να γίνει κατανοητή η φιλοσοφία που επικρατεί σήμερα στον προγραμματισμό διαδικτυακών εφαρμογών, θα γίνει μια αναφορά στο πως ξεκίνησε, πως εξελίχθηκε, καθώς επίσης και στα διάφορα στάδια τα οποία μεσολάβησαν τα χρόνια που πέρασαν. Τα κύρια στάδια από τα οποία πέρασε ο διαδικτυακός προγραμματισμός είναι τα εξής:

- Πρώτο στάδιο: στατικό περιεχόμενο (στατική HTML).
- Δεύτερο στάδιο: χρήση του **CGI** πρωτοκόλλου (*Common Gateway Interface*) ώστε να υπάρχει πλέον η δυνατότητα εκτέλεσης προγραμμάτων στον server και να παράγεται δυναμικό περιεχόμενο.
- Τρίτο στάδιο: δυνατότητα εκτέλεσης προγραμμάτων στη μεριά του χρήστη (client side), για τη δημιουργία ιστοσελίδων με μεγαλύτερη αλληλεπίδραση με το χρήστη.
- Τέταρτο στάδιο: δημιουργία τεχνολογιών με σκοπό την ανάπτυξη ολοκληρωμένων διαδικτυακών εφαρμογών χρησιμοποιώντας server-side-scripting.

Ακολουθεί μία ανάλυση κάθε σταδίου.

2.2.1 Στατικό περιεχόμενο

Πρόκειται για το πρώτο στάδιο του διαδικτυακού προγραμματισμού. Σε αυτό το στάδιο άρχισαν να κάνουν την εμφάνισή τους οι πρώτες σελίδες, οι οποίες φυσικά ήταν στατικές. Όταν λέμε ότι μία σελίδα είναι στατική εννοούμε ότι το περιεχόμενό της μεταφέρεται στο χρήστη ακριβώς στη μορφή που είναι αποθηκευμένο στον εξυπηρετητή (web server). Τα περιεχόμενα μιας στατικής ιστοσελίδας εμφανίζονται με την ίδια μορφή σε όλους τους χρήστες. Είναι αποθηκευμένες συνήθως σε μορφή HTML και μεταφέρονται χρησιμοποιώντας το πρωτόκολλο HTTP. Επίσης έκαναν την εμφάνισή τους οι πρώτοι απλοί HTML editors αλλά και οι πρώτες υπηρεσίες όπως μηχανές αναζήτησης και διαδικτυακά προγράμματα που βοηθούν ακόμη περισσότερο στην ανάπτυξη του παγκόσμιου ιστού.

Αξίζει να σημειωθεί πως για να δημιουργήσει κάποιος μια στατική σελίδα δεν χρειάζεται να κατέχει προγραμματιστικές δεξιότητες, και πως η σελίδα μπορεί να βρίσκεται με εύκολο τρόπο κλωνοποιημένη σε περισσότερους από έναν εξυπηρετητές (αντιγράφοντας απλά τα αρχεία HTML από τον ένα εξυπηρετητή στον άλλον). Ακόμα μία στατική σελίδα μπορεί να είναι διαθέσιμη στον φυλλομετρητή (Web Browser) κατευθείαν από ένα αποθηκευτικό μέσο. Βέβαια η στατική ιστοσελίδα έχει και ένα πολύ βασικό μειονέκτημα καθώς είναι εύκολη η διαδραστικότητά της με τον χρήστη.



Εικόνα 1: Αίτημα του χρήστη (client) προς το server για την προβολή μιας στατικής ιστοσελίδας

2.2.2 Δυναμικό περιεχόμενο Gateway Programming

Η αφορμή για τη μετάβαση στο δεύτερο στάδιο ήταν η ανάγκη για την ύπαρξη δυναμικού περιεχομένου στις ιστοσελίδες. Το πρώτο βήμα προς αυτήν την κατεύθυνση ήταν η δημιουργία κάποιων προγραμμάτων (πχ σε C) τα οποία εκτελούνταν στο server και επέστρεφαν στο χρήστη το δυναμικό περιεχόμενο HTML που δημιουργούσαν.

Η όλη διαδικασία γινόταν με τη χρήση του πρωτοκόλλου. Το συγκεκριμένο πρωτόκολλο είχε βασιστεί κυρίως στην έννοια του standard input (stdin) και standard output (stdout), το οποίο ξεκίνησε να αναπτύσσεται αρχικά στο λειτουργικό σύστημα Unix, και υπήρχε και σε γλώσσες 3^{ης} γενιάς όπως C, C++, κ.α. Όταν ο server καλούσε ένα πρόγραμμα C χρησιμοποιώντας το πρωτόκολλο CGI, τότε κάθε έξοδος στο stdout από το C πρόγραμμα κατευθύνονταν προς τον browser.

Ένα από τα βασικά μειονεκτήματα του CGI προγραμμάτων ήταν όταν δεν μπορούσαν να ενσωματωθούν σε HTML αρχεία, με αποτέλεσμα τη αύξηση φόρτου στο server. Πάντως αποτέλεσε το πρώτο βήμα για την δημιουργία δυναμικού περιεχομένου και προς το server side προγραμματισμό.

2.2.3 Client Side Scripting

Το γεγονός ότι ο παγκόσμιος άρχισε να αναπτύσσεται ραγδαία, οδήγησε στην δημιουργία ενός τεράστιου αριθμού στατικών ιστοσελίδων. Έτσι σιγά σιγά άρχισε να γίνεται κατανοητή η ανάγκη για τη δημιουργία ιστοσελίδων με μεγαλύτερη αλληλεπίδραση με το χρήστη. Η αρχή έγινε με την παρουσίαση του Netscape Navigator 2.0 όπου παρουσιάστηκε για πρώτη φορά η λειτουργία mouseover – mouseout. Συγκεκριμένα όταν πήγαινε το ποντίκι πάνω σε μια εικόνα (mouseover) άλλαζε το περιεχόμενο της και όταν έφευγε το ποντίκι (mouseout) άλλαζε πάλι. Έτσι αναπτύχθηκαν διάφοροι μηχανισμοί για τη δημιουργία κυρίως εντυπωσιακών μενού αλλά και άλλων εφφε, που στον τότε “στατικό κόσμο” της HTML κέρδισαν γρήγορα το ενδιαφέρον.

Η εταιρεία Netscape, που οδηγούσε την εξέλιξη των browsers προχώρησε στη δημιουργία ενός μηχανισμού που θα παρείχε τη δυνατότητα στους χρήστες να δημιουργήσουν εντυπωσιακές ιστοσελίδες με μεγαλύτερη αλληλεπίδραση με το χρήστη, καθώς και με δυναμικό περιεχόμενο. Ο μηχανισμός αυτός ήταν η γλώσσα προγραμματισμού JavaScript (γίνεται αναλυτική αναφορά αργότερα στο κεφάλαιο). Οι προγραμματιστές είχαν τη δυνατότητα να γράψουν προγράμματα σε JavaScript και να τα ενσωματώσουν σε HTML σελίδες ώστε να γίνουν πιο δυναμικές.

Λόγω της τεράστιας δημοσιότητας που πήρε η JavaScript, η Microsoft, που είχε αρχίσει να ενδιαφέρεται για το Internet, αποφάσισε να δημιουργήσει τη δικιά της scripting γλώσσα τη VBscript. Η VBscript στηριζόταν στην αρκετά δημοφιλή και επιτυχημένη Visual Basic. Έτσι όποιος ήξερε να γράφει σε Visual Basic, μπορούσε εύκολα να γράψει και σε VBscript. Αυτό που ήθελε η Microsoft ήταν να ανταγωνιστεί η VBscript τη Javascript. Κάτι τέτοιο όμως δε συνέβει ποτέ, διότι ποτέ η VBscript δεν είχε την αποδοχή που είχε η Javascript.

Γενικά στο Client Side Scripting ο κώδικας τρέχει στον web client και όχι στη μεριά του server. Αυτό είναι και το βασικό χαρακτηριστικό του σταδίου αυτού. Το scripting κομμάτι που ενσωματώνεται στο HTML αρχείο, εκτελείται από τον web browser, αφού αποσταλεί η σελίδα από τον server.

2.2.4 Server Side Scripting

Λόγω της τεράστιας ανάπτυξης που είχε πάρει το διαδίκτυο, ολοένα και περισσότερες επιχειρήσεις και οργανισμοί το χρησιμοποιούσαν. Διαπιστώθηκε λοιπόν πως θα μπορούσε να γίνει μια πλατφόρμα ανάπτυξης ολοκληρωμένων επιχειρηματικών εφαρμογών. Η ανάπτυξη του ηλεκτρονικού εμπορίου ήταν ένα επιπλέον έναυσμα σε αυτήν τη διαπίστωση.

Σε αυτό το στάδιο ανήκει η εξέλιξη και η πρόοδος των server side τεχνολογιών. Μία μικρή αρχή είχε γίνει στο δεύτερο στάδιο με την δημιουργία του πρωτοκόλλου CGI. Ωστόσο όμως γρήγορα έγινε αντιληπτό πως η τρέχουσα τεχνολογία δεν μπορούσε να καλύψει πλήρως τις ανάγκες που υπήρχαν, καθώς το συγκεκριμένο πρωτόκολλο παρείχε μόνο τη δυνατότητα αποστολής δεδομένων από τον client στον server και αντίθετα.

Στο τέταρτο στάδιο έκαναν την εμφάνιση τους server side γλώσσες προγραμματισμού που παρείχαν δυνατότητες τις οποίες το CGI δεν μπορούσε να καλύψει. Για παράδειγμα τέτοιες λειτουργίες θα μπορούσαν να είναι η αποθήκευση κάποιων μεταβλητών όταν ο χρήστης μεταφερόταν απο σελίδα σε σελίδα, ή η διαχείριση κάποιων χρηστών ανάλογα με την περιοχό- χώρα στην οποία βρίσκονται. Τα προχωρημένα περιβάλλοντα για server-side εφαρμογές έχουν ενσωματωμένες και έτοιμες προς χρήση λειτουργίες όπως

αυτές που προαναφέρθηκαν και έλυσαν τα χέρια των προγραμματιστών διαδικτυακών εφαρμογών.



Εικόνα 2: Αίτημα του χρήστη (client) προς το server για την αυθεντικοποίηση του χρήστη και προβολή του κατάλληλου περιεχομένου

2.3 Αρχιτεκτονική Client/Server

Είναι κατανοητό πως το διαδίκτυο βασίζεται στην αρχιτεκτονική **Client/Server**. Μια διαδικτυακή εφαρμογή μπορεί να χωριστεί σε δύο τμήματα, το τμήμα που αφορά τον Client (τον χρήστη δηλαδή, στην ουσία τον web browser) και το τμήμα που αφορά τον Server (το τμήμα αυτό περιέχει τον κώδικα που εκτελείται στον server και επιστρέφει στον client τα αποτελέσματα).

Η συγκεκριμένη αρχιτεκτονική επικρατεί στο διαδίκτυο, συνεπώς και ο διαδικτυακός προγραμματισμός ακολουθεί αυτό το πρότυπο. Σύμφωνα με αυτό το πρότυπο, ο web client (που όπως προαναφέρθηκε θα μπορούσε να είναι ένας web browser) επικοινωνεί με έναν web server χρησιμοποιώντας κάποιο πρωτόκολλο (το HTTP σε αυτήν την περίπτωση) κάνοντας κάποια αίτηση για να λάβει κάποια δεδομένα ή να αποκτήσει πρόσβαση σε κάποια υπηρεσία. Ο server από τη μεριά του λαμβάνει την αίτηση του browser, την επεξεργάζεται και απαντάει είτε στέλνοντας πίσω κάποια δεδομένα (συνήθως ένα HTML αρχείο) είτε τρέχει κάποιο πρόγραμμα στη μεριά του αποστέλλοντας τα αποτελέσματα κάποιας επεξεργασίας (πχ άντληση δεδομένων από βάση).

Μπορεί η συγκεκριμένη αρχιτεκτονική να είναι αρκετά διαδεδομένη και αρκετά χρήσιμη στην λειτουργικότητα των διαδικτυακών εφαρμογών, ωστόσο όμως έχει ένα πολύ σοβαρό μειονέκτημα. Λόγω της τεράστιας απήχησης που έχει το διαδίκτυο σε όλο τον κόσμο, έχει αυξηθεί κατά πολύ ο αριθμός των

clients. Αυτό έχει σαν αποτέλεσμα, την υπερφόρτωση των server. Με λίγα λόγια αν ένας server λάβει υπερβολικά πολλές αιτήσεις την ίδια στιγμή δε θα έχει τη δυνατότητα να τους εξυπηρετήσει και έτσι δε θα λειτουργεί σωστά η διαδικτυακή εφαρμογή που φιλοξενείται σε αυτόν.

2.4 Client Side Προγραμματισμός

Με τον όρο **Client Side προγραμματισμός** αναφερόμαστε στην ανάπτυξη προγραμμάτων που εκτελούνται αποκλειστικά και μόνο στην μεριά του χρήστη από το πρόγραμμα περιήγησης του υπολογιστή. Συνήθως πρόκειται για HTML/ΧHTML σελίδες με κάποιες οδηγίες μορφοποίησης CSS, αλλά και client side scripting γλώσσες όπως Javascript, ActionScript, VBscript κ.ά., οι οποίες βοηθούν στην καλύτερη αλληλεπίδραση με το χρήστη, κάνοντας τις σελίδες πιο εντυπωσιακές. Τα αρχεία αυτά φορτώνονται από το πρόγραμμα περιήγησης το οποίο τα μεταφράζει, τα εκτελεί και εμφανίζει στο χρήστη το τελικό αποτέλεσμα. Παρακάτω ακολουθεί μια παρουσίαση από τις πιο σημαντικές γλώσσες προγραμματισμού που εκτελούνται στην μεριά του χρήστη.

2.4.1 HTML

Η γλώσσα HTML αποτελεί τη γλώσσα του παγκόσμιου ιστού και στηρίζεται στην **SGML** (*Standardized Generalized Markup Language*). Η HTML δεν είναι στην πραγματικότητα γλώσσα προγραμματισμού αλλά γλώσσα σήμανσης. Μια γλώσσα σήμανσης, παράλληλα με τον προσδιορισμό κειμένου, παρέχει και τη δυνατότητα για μαρκάρισμα του κειμένου ώστε να οριστούν διάφορα δομικά στοιχεία για το πως θα παρουσιαστεί το κείμενο. Η HTML ορίζει ένα σύνολο από σημάνσεις που προφανώς δεν αφορούν μόνο τη μορφοποίηση κειμένου αλλά εξυπηρετούν και τις ανάγκες υπερκειμένων-υπερσυνδέσμων, παρέχουν λειτουργίες ενσωμάτωσης γραφικών κ.ά.

Κάθε HTML έγγραφο αρχίζει με τη σήμανση <HTML> και χωρίζεται στη συνέχεια σε δύο μέρη. Το πρώτο μέρος, το οποίο ονομάζεται κεφαλή, βρίσκεται ανάμεσα στις σημάνσεις <HEAD> και </HEAD>. Το δεύτερο μέρος (σώμα) βρίσκεται ανάμεσα στις σημάνσεις <BODY></BODY>. Να σημειωθεί πως κάθε σήμανση στην HTML ξεκινάει με <όνομα σήμανσης> και τελειώνει με </όνομα σήμανσης>. Το σύνολο μιας σήμανσης αρχής και μιας σήμανσης

τέλους ονομάζεται HTML στοιχείο. Επίσης κάθε HTML στοιχείο μπορεί να λάβει κάποιες επιπλέον ιδιότητες-οδηγίες μορφοποίησης. Ακολουθεί ένα απλό παράδειγμα HTML.

```
<HTML>
  <HEAD>
    <TITLE>Παράδειγμα HTML</TITLE>
  </HEAD>
  <BODY>
    <P>Αυτό είναι ένα απλό παράδειγμα HTML</P>
  </BODY>
</HTML>
```

Παράδειγμα 1: Απλός κώδικας HTML

Ιστορικό εκδόσεων HTML

- *Νοέμβριος 1995*

Δημοσιεύτηκε η HTML 2.0, όπου έως και το 1996 προστέθηκαν σταδιακά οι εξής λειτουργίες:

1. Ανέβασμα αρχείου από φόρμα
2. Υποστήριξη πινάκων
3. Πελατοκεντρικοί χάρτες εικόνων

- *Ιανουάριος 1997*

Παρουσιάζεται η HTML 3.2 ως σύσταση από το **W3C** (World Wide Web Consortium). Πρόκειται για την πρώτη έκδοση που αναπτύχθηκε αποκλειστικά από το W3C. Σε αυτήν την έκδοση αφαιρέθηκαν εντελώς οι μαθηματικοί τύποι. Υιοθετήθηκαν οι περισσότερες από τις σημάνσεις του Netscape. Προκειμένου να καλυφθεί η απώλεια των μαθηματικών τύπων προτυποποιήθηκε το MathML.

- *Δεκέμβριος 1997*

Δημοσιεύτηκε η HTML 4.0 από το W3C η οποία παρείχε τρεις εκδοχές:

1. Την Strict, όπου απαγορευόταν η χρήση ξεπερασμένων στοιχείων
2. Την Transitional, όπου επιτρεπόταν η χρήση ξεπερασμένων στοιχείων
3. Την Frameset, όπου επιτρέπονταν μόνο στοιχεία frame.

- *Δεκέμβριος 1999*
Δημοσιεύτηκε η HTML 4.01, η οποία προσέφερε τις ίδιες τρεις εκδοχές με την HTML 4.0
- *Ιανουάριος 2008*
Παρουσιάστηκε η HTML5, η οποία έχει σαν στόχο να μην αποτελεί πλέον εφαρμογή της SGML, όπως η HTML, αλλά να ορίζεται ως αυτόνομη, μαζί με την XHTML5 που βασίζεται στην **XML** (*Extensible Markup Language*).

2.4.2 XHTML

Λόγω της εξαιρετικά γρήγορης εξέλιξης του παγκόσμιου ιστού, απέκτησαν πρόσβαση σε αυτόν αρκετές συσκευές, όπως υπολογιστές χειρός, κινητά τηλέφωνα, συσκευές τηλεόρασης, καθώς και άλλες “έξυπνες συσκευές”. Αυτό είχε σαν αποτέλεσμα να φανερωθούν όλες οι αδυναμίες της HTML, η οποία δεν επαρκούσε για να καλυφθούν οι ανάγκες για όλες τις συσκευές που ήθελαν πρόσβαση στο διαδίκτυο.

Ένα χαρακτηριστικό παράδειγμα είναι το γεγονός ότι η HTML δεν έχει αυστηρό συντακτικό. Επομένως αν ένας συγγραφέας ενός HTML εγγράφου κάνει κάποιο λάθος στη σύνταξη του εγγράφου, ίσως και να μη γίνει ποτέ αντιληπτό στο χρήστη, καθώς μπορεί να μην έχει σοβαρές επιπτώσεις στην αναπαράσταση του εγγράφου στους φυλλομετρητές. Βέβαια λόγω των λαθών αυτών μπορεί η αναπαράσταση του εγγράφου να διαφέρει από φυλλομετρητή σε φυλλομετρητή. Γενικά σε ένα λάθος συντακτικά HTML έγγραφο, δεν είναι απόλυτα προβλέψιμο πως θα γίνει η αναπαράστασή του.

Για να είναι δυνατή η καλύτερη αντιμετώπιση αυτών των προβλημάτων, δημιουργήθηκε από τον οργανισμό W3C η **XHTML** (*Extensive HyperText Markup Language*). Η XHTML βασίζεται στην HTML 4.01 αλλά πλέον αποτελεί εφαρμογή της XML, όχι της SGML. Έρχεται για να αντικαταστήσει πλήρως την HTML, και να βοηθήσει στην ανάπτυξη του Internet αξιοποιώντας κάποια από τα πλεονεκτήματα της XML.

Ιστορικό εκδόσεων XHTML

- *2000*

Η XHTML 1.0 ήταν η πρώτη έκδοση που προτάθηκε από το W3C. Η έκδοση αυτή ήταν η υλοποίηση της HTML 4.01 ως εφαρμογή της XML πλέον. Η XHTML 1.0 είχε τρεις εκδοχές, όπου η κάθε μία δημιουργήθηκε για την κάλυψη διαφορετικών αναγκών.

1. XHTML 1.0 Strict
2. XHTML 1.0 Transitional
3. XHTML 1.0 Frameset

Η έκδοση που χρησιμοποιείται στο κάθε έγγραφο δηλώνεται στην αρχή του εγγράφου με την κατάλληλη δήλωση. Π.χ. με την παρακάτω δήλωση:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

- *Μάιος 2001*

Δημοσιεύτηκε η XHTML 1.1, που βασίστηκε κυρίως στην XHTML 1.0 Strict με ελάχιστες διαφορές.

- *Αύγουστος 2002 – Ιούλιος 2006*

XHTML 2.0

- *Μάρτιος 2012*

XHTML5

Εξελισσεται παράλληλα με την HTML5.

Κύριες διαφορές με την HTML

Οι κυριότερες διαφορές ανάμεσα σε XHTML και HTML είναι οι εξής:

- Τα στοιχεία της XHTML πρέπει να είναι σωστά δομημένα και φωλιασμένα (nested).
- Τα έγγραφα XHTML πρέπει να είναι σωστά μορφοποιημένα (well-formed).
- Τα ονόματα των XHTML στοιχείων πρέπει να γράφονται με πεζά γράμματα και όχι με κεφαλαία.

- Όλα τα στοιχεία της XHTML πρέπει να είναι κλειστά.

2.4.3 CSS

Με τη χρήση των **CSS** (*Cascading Style Sheets*) επιχειρείται στην HTML η μετάβαση από τη σειριακή σήμανση στη γενική. Τα διαδοχικά φύλλα μορφοποίησης δημιουργήθηκαν ουσιαστικά για την ομαδοποίηση μορφοποιήσεων ώστε να υπάρχει καλύτερη οργάνωση στο HTML έγγραφο, καθώς και λιγότερος κώδικας. Αν για παράδειγμα είκοσι στοιχεία στο έγγραφο έχουν ακριβώς τις ίδιες οδηγίες μορφοποίησης, τότε υπάρχουν δύο προβλήματα:

1. Αν χρειαστεί να γίνει κάποια αλλαγή στη μορφοποίηση, θα πρέπει ο προγραμματιστής να αλλάξει τις οδηγίες από όλα αυτά τα στοιχεία, το οποίο είναι εξαιρετικά χρονοβόρο και κουραστικό.
2. Υπάρχει επάναληψη κώδικα, καθώς ο προγραμματιστής γράφει είκοσι φορές ακριβώς τις ίδιες οδηγίες μορφοποίησης.

Για να λυθούν αυτά τα προβλήματα προστέθηκε η ιδιότητα **STYLE** και η έννοια των **Css**. Έτσι είναι πλέον εφικτή η ομαδοποίηση των κοινών οδηγιών μορφοποιήσεων σε μία. Με αυτόν το τρόπο τα παραπάνω προβλήματα αντιμετωπίζονται ως εξής:

- Δημιουργούμε μία οδηγία μορφοποίησης ίδια με τις είκοσι και συνδέουμε τα είκοσι στοιχεία με αυτήν την οδηγία.

Υπάρχουν τρεις τρόποι εφαρμογής CSS στην HTML:

1. Σε επίπεδο ενός συγκεκριμένου στοιχείου (*in-line styles*). Εισάγονται κατευθείαν μέσα σε μία σήμανση χρησιμοποιώντας την ιδιότητα **style**.
2. Σε επίπεδο σελίδας (*embedded styles*). Χρησιμοποιούνται σε ολόκληρο το έγγραφο HTML. Τα φύλλα στυλ (*style sheet*) ορίζονται μέσα στη σήμανση **HEAD**.
3. Σε επίπεδο ενός συνόλου σελίδων (*external*). Τα εξωτερικά φύλλα στυλ αποθηκεύονται σε ξεχωριστό αρχείο το οποίο ο προγραμματιστής το συνδέει με το HTML έγγραφο στο οποίο θέλει να εφαρμόσει τις οδηγίες μορφοποίησης.

Συγκρούσεις στην χρήση πολλών τρόπων εφαρμογής

Το γεγονός ότι υπάρχουν τρεις τρόποι εφαρμογής των CSS δε σημαίνει πως δεν είναι δυνατόν να χρησιμοποιηθούν και οι τρεις ταυτόχρονα. Το ερώτημα είναι ποιος τρόπος θα εφαρμοστεί σε περίπτωση που χρησιμοποιηθεί παραπάνω από ένας. Σε γενικές γραμμές όσα τα στυλ θα συγκλίνουν σε ένα νέο στυλ που αποτελεί τη σύνθεση όλων των στυλ με την εξής προτεραιότητα:

1. Inline Style, Μέσα στο HTML στοιχείο.
2. Εσωτερικό φύλλο στυλ μέσα στο HEAD του HTML εγγράφου.
3. Εξωτερικό φύλλο στυλ.
4. Προεπιλογή του browser.

2.4.5 JavaScript

Η **JavaScript** είναι μία γλώσσα προγραμματισμού που έχει κάποιες ιδιότητες και επηρεασμούς από αντικειμενοστραφείς γλώσσες προγραμματισμού, όπως η C++ και η Java. Βέβαια αρκετοί υποστηρίζουν πως δεν πρόκειται για μια γλώσσα προγραμματισμού όπως οι άλλες, χωρίς να έχει ξεκαθαρίσει το τοπίο.

Η JavaScript βρίσκεται σχεδόν παντού στον παγκόσμιο ιστό. Οι πιο γνωστοί πλοηγητές διαδικτύου υποστηρίζουν τη JavaScript. Έτσι όταν σε μία HTML σελίδα υπάρχει ενσωματωμένος JavaScript κώδικας, εκτελείται κατευθείαν στον browser στη μεριά του χρήστη (client side). Πρόκειται για την πιο διάσημη και ευρέως χρησιμοποιούμενη γλώσσα προγραμματισμού στον client side προγραμματισμού.

Η δημιουργία της στηρίχτηκε στη δομή των γλωσσών προγραμματισμού που υπήρχαν την εποχή εκείνη (C, C++, Java), τις οποίες μάλιστα χρησιμοποιούσαν αρκετά οι προγραμματιστές. Το γεγονός αυτό ήταν ένα επιπλέον στοιχείο για την ανάπτυξή της, αφού αν κάποιος ήταν εξοικειωμένος με τις παραπάνω γλώσσες, θα χρειαζόταν λίγο χρόνο για να μάθει να γράφει προγράμματα σε JavaScript.

Ιστορία της JavaScript

Η JavaScript γεννήθηκε από τη Netscape Communications το 1995. Ήταν εκείνη η εποχή όπου ο παγκόσμιος ιστός άρχισε να αναπτύσσεται ραγδαία, με αποτέλεσμα να δημιουργηθεί ένας τεράστιος αριθμός από στατικές HTML ιστοσελίδες. Ωστόσο η ανάγκη των χρηστών για δημιουργία πιο εντυπωσιακών σελίδων (είτε με δυναμικό περιεχόμενο, είτε με μεγαλύτερη αλληλεπίδραση με το χρήστη) οδήγησε στη δημιουργία μίας γλώσσας που εξυπηρετούσε αυτό το σκοπό.

Αρχικά είχε δημιουργηθεί μία γλώσσα που ενσωματώνονταν στον δημοφιλή τότε Netscape Navigator, η οποία παρείχε τη δυνατότητα σε μη έμπειρους προγραμματιστές να γράψουν εύκολα ένα μικρό κομμάτι κώδικα και να προσθέσουν κάποια επιπλέον λειτουργικότητα στις στατικές HTML σελίδες. Παρόλα αυτά η JavaScript αναπτύχθηκε πολύ και έγινε πάρα πολύ δημοφιλής για την ανάπτυξη κώδικα που θα έτρεχε στους web browsers.

2.4.6 JQuery

Η **Jquery** είναι μία βιβλιοθήκη της JavaScript συμβατή με τους περισσότερους φυλλομετρητές και σχεδιάστηκε για να γίνει πιο απλός ο διαδικτυακός προγραμματισμός στη μεριά του πελάτη (client-side-scripting). Χαρακτηριστικά της jquery είναι η ευχρηστία της και η διαδραστικότητα που προσφέρει στις διαδικτυακές εφαρμογές όπου χρησιμοποιείται. Πρωτοεμφανίστηκε τον Ιανουάριο του 2006 στο BarCamp από τον John Resig. Είναι μία βιβλιοθήκη ανοιχτού κώδικα, υπό τις άδειες MIT License και την **GNU** (*Generic Public License*).

Πλεονεκτήματα της JQuery

- Ακολουθεί την αρχή KISS (Keep It Simple Stupid). Η βιβλιοθήκη jquery προσπαθεί να κάνει πιο απλό τον προγραμματισμό σε Javascript προσφέροντας απλούς μηχανισμούς και εντολές.
- Υπάρχει άφθονο υλικό τεκμηρίωσης και επεξήγησης της κάθε εντολής και μηχανισμού της jquery, ώστε να μπορούν οι προγραμματιστές να εκμεταλλεύονται τις δυνατότητες της στο 100%.

- Το γεγονός ότι το βασικό πακέτο της JQuery είναι μόλις 95kb είναι ένα επιπλέον στοιχείο που αποδεικνύει πως είναι εξαιρετικά απλή και φιλική προς το χρήστη.
- Η JQuery δίνει τη δυνατότητα στον χρήστη να χρησιμοποιήσει σχεδόν το σύνολο των δυνατοτήτων που προσφέρει η γλώσσα JavaScript. Από απλά χαρακτηριστικά που σχετίζονται με βασικές λειτουργίες εμφάνισης / απόκρυψης, ως Ajax (γίνεται αναλυτική αναφορά σε επόμενο κεφάλαιο) κλήσεις και σύνθετα εφέ.

2.5 Server Side Προγραμματισμός

Όπως αναφέρθηκε σε προηγούμενο κεφάλαιο, ο **server side προγραμματισμός** αφορά τα προγράμματα που εκτελούνται στη μεριά του Server, δηλαδή στον υπολογιστή που φιλοξενεί την διαδικτυακή εφαρμογή. Ο κάθε χρήστης (client) στέλνει ένα αίτημα σε κάποιον server (πχ για εμφάνιση μιας ιστοσελίδας), ο οποίος με τη σειρά του απαντάει στο αίτημα στέλνοντας το κατάλληλο HTML αρχείο. Το αρχείο αυτό το λαμβάνει το πρόγραμμα περιήγησης του χρήστη, το μεταφράζει και εμφανίζει το τελικό αποτέλεσμα την οθόνη του χρήστη, στην περίπτωση αυτή μία απλή σελίδα. Βέβαια ένα αίτημα δεν χρειάζεται απαραίτητα να σχετίζεται με μια εμφάνιση μιας ιστοσελίδας. Μπορεί να αφορά και άλλα πράγματα, όπως ταυτοποίηση στοιχείων, αναζήτηση, κατέβασμα-ανέβασμα αρχείων ή ακόμα και μεταφορά αρχείων από server σε server.

Τα server-side scripts αρκετές φορές δεν γίνονται αντιληπτά από τους χρήστες αφού οι λειτουργίες που πραγματοποιούν αφορούν κυρίως τον server. Επεξεργάζονται πληροφορίες που εισάγει ο χρήστης με διάφορους τρόπους, όπως φόρμες επικοινωνίας, αναζήτηση πληροφοριών κ.α, ή μπορούν ακόμα να χρησιμοποιήσουν κάποιες πληροφορίες από βάσεις δεδομένων. Είναι εξαιρετικά χρήσιμα καθώς βοηθούν στη δημιουργία δυναμικού περιεχομένου και στην σωστή εμφάνιση των πληροφοριών στους χρήστες ανάλογα με την ώρα, την χώρα στην οποία βρίσκονται κ.ά.

Ένα επιπλέον χαρακτηριστικό των server side προγραμμάτων είναι ότι δεν επιβαρύνουν τον χρήστη, αφού όπως λέει και το όνομα τους εκτελούνται στο server. Βέβαια η φιλοσοφία που ακολουθείται, δηλαδή στέλνει ο χρήστης

αίτημα και λαμβάνει από το server την ανάλογη απάντηση, κρύβει κάποιους κινδύνους. Ένας Server θα πρέπει συνεχώς να συντηρείται και να αναβαθμίζεται, για την αποφυγή προβλημάτων ασφάλειας αλλά και σταθερότητας. Αν κάποιο server side script δεν είναι σωστά γραμμένο (όχι από άποψης συντακτικού αλλά από λογικής άποψης), τότε θα υπάρχει η λεγόμενη τρύπα στην διαδικτυακή εφαρμογή. Έτσι σε περίπτωση που γίνει κάτι τέτοιο αντιληπτό σε έναν χρήστη, τότε μπορεί να στείλει ένα συγκεκριμένο αίτημα (πχ Sql Injection) και να σπάσει την διαδικτυακή εφαρμογή αποκτώντας πρόσβαση σε πληροφορίες που δεν θα έπρεπε να έχει. Επομένως θέλει ιδιαίτερη προσοχή όταν κάποιος γράφει server side script ώστε να αποφευχθούν τέτοιες περιπτώσεις. Παρακάτω ακολουθεί μια μικρή ανάλυση για της πιο γνωστές τεχνολογίες που χρησιμοποιούνται στον server side προγραμματισμό.

2.5.1 PHP

Η **PHP** είναι ίσως η πιο διάσημη και ευρέως χρησιμοποιούμενη γλώσσα script από την μεριά του server και σχεδιάστηκε ειδικά για τη δημιουργία ολοκληρωμένων διαδικτυακών εφαρμογών. Οι προγραμματιστές έχουν τη δυνατότητα να ενσωματώσουν PHP κώδικα σε μια HTML σελίδα, οποίος θα εκτελείται κάθε φορά που κάποιος θέλει να την επισκεφτεί. Ο κώδικας αυτός μεταφράζεται στον server και δημιουργεί είτε το περιεχόμενο HTML, που είναι δυναμικό το οποίο το βλέπει ο χρήστης, είτε πραγματοποιεί άλλες λειτουργίες στο server και επιστρέφει στο χρήστη το ανάλογο αποτέλεσμα. Για να γίνει αυτή η μετάφραση (η εκτέλεση ουσιαστικά του κώδικα) θα πρέπει ο server να είναι συμβατός με PHP, να είναι δηλαδή η PHP εγκατεστημένη σε αυτόν.

Η PHP είναι ένα προϊόν ανοιχτού κώδικα, που σημαίνει ότι ο καθένας μπορεί να έχει πρόσβαση στον κώδικα προέλευσης, να τον χρησιμοποιήσει, να τον αλλάξει και να τον αναδιανέμει, χωρίς καμία απολύτως χρέωση. Αρχικά ήταν ακρωνύμιο του *Personal Home Page*, αλλά στη συνέχεια άλλαξε σύμφωνα με τη σύμβαση GNU και πλέον τα αρχικά της PHP σημαίνουν *PHP Hypertext Preprocessor*. Η τρέχουσα κύρια έκδοση είναι η PHP 5.

Ιστορικά

Η ιστορία της PHP ξεκινά από το 1994, όταν ένας φοιτητής, ο Rasmus Lerdorf δημιούργησε ένα απλό script χρησιμοποιώντας τη γλώσσα προγραμματισμού Perl με όνομα php.cgi, για προσωπική χρήση. Το script αυτό είχε σαν σκοπό να διατηρεί μια λίστα στατιστικών για τα άτομα που έβλεπαν το online βιογραφικό του σημείωμα. Αργότερα αυτό το script το διέθεσε και σε φίλους του, οι οποίοι άρχισαν να του ζητούν να προσθέσει περισσότερες δυνατότητες. Η γλώσσα τότε ονομαζόταν PHP/FI από τα αρχικά Personal Home Page/Form Interpreter.

Το 1997 η PHP/FI έφθασε στην έκδοση 2.0, βασιζόμενη αυτή τη φορά στη γλώσσα C και αριθμώντας περισσότερους από 50.000 ιστότοπους που τη χρησιμοποιούσαν, ενώ αργότερα την ίδια χρονιά οι Andi Gutmans και Zeev Suraski ξαναέγραψαν τη γλώσσα από την αρχή, βασιζόμενοι όμως αρκετά στην PHP/FI 2.0. Έτσι η PHP έφθασε στην έκδοση 3.0 η οποία θύμιζε περισσότερο τη σημερινή μορφή της. Στη συνέχεια, οι Zeev και Andi δημιούργησαν την εταιρεία Zend (από τα αρχικά των ονομάτων τους), η οποία συνεχίζει μέχρι και σήμερα την ανάπτυξη και εξέλιξη της γλώσσας PHP.

Ακολούθησε το 1998 η έκδοση 4 της PHP, τον Ιούλιο του 2004 διατέθηκε η έκδοση 5, ενώ αυτή τη στιγμή έχουν ήδη διατεθεί και οι πρώτες δοκιμαστικές εκδόσεις της επερχόμενης PHP 6, για οποιονδήποτε προγραμματιστή θέλει να τη χρησιμοποιήσει.

Το Νοέμβριο του 2007, ήταν εγκατεστημένη σχεδόν σε εικοσιένα εκατομμύρια τομείς παγκόσμια και αυτός ο αριθμός μεγαλώνει συνεχώς.

Οι περισσότεροι ιστότοποι επί του παρόντος χρησιμοποιούν κυρίως τις εκδόσεις 4 και 5 της PHP.2.5.2 Perl.

Πλεονεκτήματα της Php

- **Κόστος**

Η PHP είναι δωρεάν. Μπορεί ο καθένας να μπει στην επίσημη ιστοσελίδα της και να κατεβάσει την τελευταία έκδοση χωρίς καμμία οικονομική επιβάρυνση.

- Ευκολία εκμάθησης
Η σύνταξη της βασίζεται σε άλλες γλώσσες προγραμματισμού, κυρίως στην C και την Perl. Αν κάποιος ξέρει ήδη Perl ή C μπορεί εύκολα να προσαρμοστεί και να εξοικειωθεί στην PHP.
- Αντικειμενοστραφής υποστήριξη
Η PHP 5 διαθέτει καλά σχεδιασμένες αντικειμενοστραφείς λειτουργίες. Εάν κάποιος διαθέτει καλές προγραμματιστικές γνώσεις σε Java ή C++, θα διαπιστώσει ότι και η PHP έχει την ίδια περίπου σύνταξη σε παρόμοιες εντολές, κληρονομικότητα, ιδιωτικές ή προστατευμένες ιδιότητες, αφηρημένες κλάσεις, διασυνδέσεις κ.ά.
- Μεταφερσιμότητα
Η PHP είναι διαθέσιμη για πολλά διαφορετικά λειτουργικά συστήματα. Ο καλογραμμένος κώδικας συνήθως θα δουλεύει χωρίς πολλές αλλαγές σε διαφορετικά συστήματα, στα οποία φυσικά είναι εγκατεστημένη η PHP.
- Υποστήριξη βάσεων δεδομένων
Διαθέτει τις δικές της διασυνδέσεις με τις οποίες ο προγραμματιστής μπορεί να συνδεθεί σε πολλά συστήματα βάσεων δεδομένων, όπως MySQL, PostgreSQL, Oracle, IBM, FilePro DB2, Hyperware, Informix, InterBase, Sybase.
- Ενσωματωμένες βιβλιοθήκες
Η PHP έχει σχεδιαστεί με τέτοιο τρόπο ώστε να είναι δυνατή η χρήση της στον παγκόσμιο ιστό. Επομένως έχει αρκετές ενσωματωμένες βιβλιοθήκες που εκτελούν πολλές χρήσιμες λειτουργίες σχετικές με τον παγκόσμιο ιστό. Τέτοιες λειτουργίες θα μπορούσαν να είναι δημιουργία δυναμικών εικόνων, σύνδεση με άλλες διαδικτυακές υπηρεσίες, ανάλυση XML δεδομένων, χρήση ηλεκτρονικού ταχυδρομείου και άλλες πολλές, και όλα αυτά με τη χρήση λίγων γραμμών κώδικα.
- Απόδοση
Τέλος η PHP είναι εξαιρετικά γρήγορη, καθώς με έναν server χαμηλού κόστους, είναι δυνατή η εξυπηρέτηση εκατομμυρίων επισκέψεων καθημερινά.

2.5.3 Perl

Τα αρχικά **Perl** σημαίνουν *Practical Extraction and Report Language*. Η Perl είναι μια γλώσσα προγραμματισμού που αναπτύχθηκε από τον *Larry Wall* και σχεδιάστηκε ειδικά για την επεξεργασία κειμένου. Εξαιτίας όμως των ισχυρών δυνατοτήτων επεξεργασίας κειμένου που διαθέτει, η Perl έχει γίνει μια από τις πιο δημοφιλείς γλώσσες για το γράψιμο των γνωστών CGI scripts.

Η Perl είναι διαθέσιμη δωρεάν στο Web και χρησιμοποιείται σε πολλές εφαρμογές, όπως το να γράψει κάποιος CGI scripts που βοηθούν τους administrators στην συντήρηση των συστημάτων τους. Η Perl δημιουργήθηκε, και συντηρείται ακόμα, από τον *Larry Wall*. Είναι πιο αργή από την C, αλλά πιο γρήγορη από μια κανονική διερμηνευόμενη γλώσσα.

Ένα από τα πιο ωραία πράγματα σχετικά με την Perl είναι το ότι διατίθεται δωρεάν. Διανέμεται με την άδεια GNU license και ο πηγαίος κώδικάς της είναι διαθέσιμος από την αρχική ιστοσελίδα της Perl (Perl Home Page). Αυτό, σε συνδυασμό με το ότι είναι ευέλικτη και έχει πολύ λίγους περιορισμούς, κάνουν την Perl μια δημοφιλή γλώσσα. Ακόμα, υποστηρίζεται από πολλές πλατφόρμες όπως Unix, Windows και DOS.

Ιστορικά

Η γλώσσα προγραμματισμού Perl σχεδιάστηκε επειδή ο *Larry Wall* ήθελε μία γλώσσα η οποία να συνδυάζει μόνο τα καλά στοιχεία του sed, της C, της awk και του κελύφους Bourne του λειτουργικού συστήματος Unix. Η πρώτη έκδοση της γλώσσας ανακοινώθηκε στο alt.comp.sources του Usenet στις 18 Δεκεμβρίου του 1987. Μετά την ταχύτατη διάδοση της γλώσσας ακολούθησαν οι εκδόσεις 2 (1988) και 3 (1989). Η τέταρτη έκδοση του 1991 δεν είχε σημαντικές διαφορές από την 3η αλλά δημιουργήθηκε ως έκδοση αναφοράς για το πρώτο βιβλίο με θέμα τη γλώσσα. Η πέμπτη έκδοση ξεκίνησε να γράφεται το 1993 με την σταθερή έκδοση της να εκδίδεται τον Οκτώβριο του 1994. Η έκδοση αυτή έφερε σημαντικές αλλαγές και προσέθεσε σημαντικές δυνατότητες ενώ ταυτόχρονα αύξησε τον αριθμό των υποστηριζόμενων λειτουργικών συστημάτων.

Η έκδοση 5 συνεχίζει να αναπτύσσεται ακόμα και βρίσκεται στην έκδοση 5.14.1. Η *Perl 6* είναι η επόμενη έκδοση της γλώσσας η οποία θα επιφέρει αρκετές αλλαγές τόσο στο συντακτικό όσο και στις δυνατότητες της γλώσσας.

2.5.4 MySQL

Πρόκειται για ένα πολύ γρήγορο και δυνατό, σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων. Μια βάση δεδομένων δίνει τη δυνατότητα στον προγραμματιστή να αποθηκεύει, να αναζητά, να ταξινομεί καθώς και να ανακαλεί τα δεδομένα του με αποτελεσματικό τρόπο. Ο **MySQL** διακομιστής ελέγχει την πρόσβαση στα δεδομένα ώστε να διασφαλίσει ότι πολλοί χρήστες θα είναι σε θέση να δουλεύουν ταυτόχρονα, παρέχει γρήγορη πρόσβαση σε αυτά, και επιπλέον εξασφαλίζει ότι μόνο οι πιστοποιημένοι χρήστες θα έχουν πρόσβαση σε σημαντικές πληροφορίες.

Η MySQL είναι ένας πολυνηματικός διακομιστής πολλαπλών χρηστών. Χρησιμοποιεί την **SQL** (*Structured Query Language*), που είναι και η τυπική γλώσσα ερωτημάτων για βάσεις δεδομένων. Είναι πλέον διαθέσιμη με άδειες δύο ειδών. Οι προγραμματιστές μπορούν να τη χρησιμοποιήσουν δωρεάν με άδεια ανοιχτού κώδικα, αρκεί να συμφωνήσουν με τους όρους αυτής της άδειας. Αν όμως κάποιος θέλει να δημιουργήσει μία εφαρμογή που δεν είναι ανοιχτού κώδικα που να περιλαμβάνει την MySQL, τότε θα πρέπει να αγοράσει μία εμπορική άδεια.

Ιστορία της MySQL

Η MySQL δημιουργήθηκε από μια σουηδική εταιρεία, την MySQL AB, η οποία ιδρύθηκε από τους David Axmark, Allan Larsson και Michael “Monty” Widenius.

Η πρώτη έκδοση της MySQL εμφανίστηκε στις 23 Μαΐου 1995. Αρχικά είχε δημιουργηθεί για προσωπική χρήση από mSQL (Mini SQL) βασισμένη στην χαμηλού επιπέδου γλώσσα ISAM, που οι δημιουργοί της την θεωρούσαν πολύ αργή και δύσχρηστη. Στη συνέχεια δημιούργησαν έτσι μία νέα διεπαφή SQL, ενώ συνέχιζαν να διατηρούν την ίδια βιβλιοθήκη ως mSQL.

2.5.5 Postgress

Η **PostgreSQL** αποτελεί μια ανοιχτού κώδικα σχεσιακή βάση δεδομένων με πολλές δυνατότητες. Η ανάπτυξη της ήδη διαρκεί πάνω από 20 χρόνια και βασίζεται σε μια αποδεδειγμένα καλή αρχιτεκτονική η οποία έχει δημιουργήσει μια ισχυρή αντίληψη των χρηστών της γύρω από την αξιοπιστία, την ακεραιότητα δεδομένων και την ορθή λειτουργία.

Τρέχει σε όλα τα βασικά λειτουργικά συστήματα, περιλαμβάνοντας Linux, UNIX (AIX, BSD, HP-UX, SGI IRIX, Mac OS X, Solaris, Tru64), και Windows. Είναι ACID συμβατή (ACID compliant), έχει ολοκληρωμένη υποστήριξη για foreign keys, joins, views, triggers, και stored procedures (σε διάφορες γλώσσες προγραμματισμού). Συμπεριλαμβάνει τα περισσότερα SQL92 και SQL99 data types, συμπεριλαμβανομένων INTEGER, NUMERIC, BOOLEAN, CHAR, VARCHAR, DATE, INTERVAL και TIMESTAMP. Επίσης υποστηρίζει αποθήκευση binary large objects, όπως εικόνες, ήχοι ή video. Διαθέτει native programming interfaces για C/C++, Java, .Net, Perl, Python, Ruby, Tcl, ODBC, κ.α.

Ιστορικά

- Σχέδιο POSTGRES

Το σχέδιο *POSTGRES* υπό την ηγεσία του καθηγητή Michael Stonebreaker ξεκίνησε να εφαρμόζεται το 1986. Η *POSTGRES* έχει αρκετά σημαντικές διανομές από τότε. Η πρώτη δοκιμαστική έκδοση του συστήματος παρουσιάστηκε το 1988 στο συνέδριο *ACM-SIGMOD* και κυκλοφόρησε σε μικρή ομάδα ατόμων τον Ιούνιο του 1989.

Η *POSTGRES* χρησιμοποιήθηκε για την εφαρμογή ποικίλων εφαρμογών παραγωγής και έρευνας, όπως για παράδειγμα ένα οικονομικό σύστημα ανάλυσης δεδομένων, ένα πακέτο παρακολούθησης της απόδοσης ενός κινητήρα τζετ, βάση δεδομένων εντοπισμού αστεροειδή, μιας ιατρικής βάσης δεδομένων και μερικών συστημάτων γεωγραφικής πληροφορίας. Χρησιμοποιήθηκε επίσης ως εργαλείο εκπαίδευσης σε ορισμένα πανεπιστήμια.

- Postgres95

Το 1994 προστέθηκε διερμηνέας γλώσσας SQL από τους Andrew Yu και Jolly Chen. Υπό την ονομασία *Postgres95* απελευθερώθηκε στο διαδίκτυο ως απόγονος ανοιχτού κώδικα του αρχικού κώδικα POSTGRES Berkeley. Ο κώδικας της *Postgres95* αποτελούνταν από ANSI C και περιορίστηκε το μέγεθός του κατά 25% σε σχέση με τον κώδικα της POSTGRES. Πολλές εσωτερικές αλλαγές βελτίωσαν την αποδοτικότητα και την δυνατότητα συντήρησής της.

- PostgreSQL

Το 1996 έγινε σαφές ότι η ονομασία *Postgres95* δεν θα αντέξει με την πάροδο του χρόνου, γιαυτό και επιλέχθηκε η ονομασία *PostgreSQL* ώστε να αντικατοπτρίσει την σχέση ανάμεσα στην αρχική POSTGRES και τις πιο πρόσφατες εκδόσεις με συμβατότητα SQL. Την ίδια χρονιά ορίστηκε η αρίθμηση των εκδόσεων να ξεκινά από το νούμερο 6.0, έτσι ώστε να συνεχίζεται από την αρίθμηση που ξεκίνησε από το αρχικό σχέδιο POSTGRES Berkeley.

2.6 Τεχνολογία Ajax

Ο όρος **Ajax** προέρχεται από το *Asynchronous JavaScript and XML*. Ο όρος αυτό είχε αρχικά προταθεί από τον Jesse James Garret. Πρόκειται για έναν συνδυασμό των τεχνολογιών που υπάρχουν ήδη, και που βοηθούν τον προγραμματιστή να δημιουργήσει διαδικτυακές εφαρμογές με ακόμα μεγαλύτερη αλληλεπίδραση με το χρήστη. Όταν κάποιος δημιουργεί μία διαδικτυακή εφαρμογή, ουσιαστικά χρησιμοποιεί HTML και CSS για τη μορφοποίηση και JavaScript για να προσθέσει κάποια αλληλεπίδραση. Στην τεχνολογία Ajax ο προγραμματιστής μπορεί να χρησιμοποιήσει ένα επιπλέον αντικείμενο, το XMLHttpRequest (XHR), το οποίο θα τον βοηθήσει να στείλει αιτήματα στον server και να λάβει κάποια δεδομένα ως απάντηση, χωρίς να γίνεται επαναφόρτωση της σελίδας, ασύγχρονα δηλαδή.

Αν και δεν είναι ακριβώς νέα τεχνολογία, η νέα προσέγγιση που προτείνει η τεχνική Ajax είναι μία σημαντική αλλαγή στην κλασσική request/response προσέγγιση που υπήρχε στον παγκόσμιο ιστό. Με την

ύπαρξη της δυνατότητας για ασύγχρονη επικοινωνία με τον εξυπηρετητή, οι πιθανές εφαρμογές είναι πλέον πολλές. Όλες οι εφαρμογές πηγάζουν από τη δυνατότητα που έχει κάποιος, χρησιμοποιώντας το XHR αντικείμενο, να κάνει web εφαρμογές που θα είναι πολύ πιο δυναμικές από τις εφαρμογές που υπήρχαν ήδη.

2.7 Μεταφορά δεδομένων μεταξύ διαδικτυακών εφαρμογών

Όπως προαναφέρθηκε, η τεράστια ανάπτυξη του διαδικτύου οδήγησε στην δημιουργία ενός τεράστιου αριθμού διαδικτυακών εφαρμογών. Το γεγονός αυτό είχε σαν αποτέλεσμα την δημιουργία κάποιων διαδικτυακών υπηρεσιών που αποτελούν εργαλεία για κάποιες άλλες υπηρεσίες, αλλά και τη συνεργασία διαφορετικών υπηρεσιών μεταξύ τους. Επομένως παρατηρήθηκε η ανάγκη για την εύρεση ενός τρόπου επικοινωνίας μεταξύ διαφορετικών διαδικτυακών υπηρεσιών. Οι δύο πιο διαδεδομένοι τρόποι επικοινωνίας είναι η χρήση XML ή Json για ανταλλαγή δεδομένων στο διαδίκτυο.

2.7.1 Χρήση XML

Αν κάποιος θέλει να περιγράψει την XML με μία λέξη, τότε σίγουρα αυτή θα ήταν η λέξη δομή. Η XML προσπαθεί πάνω από όλα να δώσει μία περιγραφή στο πως είναι δομημένο ένα έγγραφο αλλά και να περιγράψει με πιο εκφραστικούς τρόπους ένα κείμενο, χτίζοντας έτσι σιγά σιγά τα θεμέλια για μια επιτυχημένη μεταφορά δεδομένων στο διαδίκτυο μεταξύ διαφορετικών κατανεμημένων εφαρμογών.

Η XML σχεδιάστηκε για να περιγράφει δεδομένα, κάτι που επιτυγχάνεται με τη χρήση των γνωστών tags. Η διαφορά με τα κλασσικά tags στην HTML είναι ότι στην XML δεν υπάρχουν προκαθορισμένες σημάνσεις. Ο χρήστης θα πρέπει να ορίσει δικές του και να περιγράψει τα δεδομένα με όποιον τρόπο πιστεύει αυτός. Επομένως η XML δεν αφορά μόνο τους σχεδιαστές του παγκόσμιου ιστού, αλλά γενικότερα όποιον ασχολείται με τον προγραμματισμό διαδικτυακών εφαρμογών.

2.7.2 Χρήση JSON

Το **JSON** (*JavaScript Object Notation*) είναι ένα ελαφρύ πρότυπο ανταλλαγής δεδομένων. Είναι εύκολο για τους ανθρώπους να το διαβάσουν

και γράφουν. Είναι εύκολο για τις μηχανές να το αναλύσουν (parse) και να το παράγουν (generate).

Είναι βασισμένο πάνω σε ένα υποσύνολο της γλώσσας προγραμματισμού JavaScript, Standard ECMA-262 Έκδοση 3η - Δεκέμβριος 1999. Το JSON 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 είναι χτισμένο σε δύο δομές:

- Μια συλλογή από ζευγάρια ονομάτων/τιμών. Σε διάφορες γλώσσες προγραμματισμού, αυτό αντιλαμβάνεται ως ένα object, καταχώριση, δομή, λεξικό, πίνακα hash (hash table), λίστα κλειδιών, ή associative πίνακα.
- Μία ταξινομημένη λίστα τιμών. Στις περισσότερες γλώσσες προγραμματισμού, αυτό αντιλαμβάνεται ως ένας πίνακας (array), διάνυσμα, λίστα, ή ακολουθία.

Κεφάλαιο 3

Συστήματα διαχείρισης περιεχομένου (CMS)

3.1 Σύστημα διαχείρισης περιεχομένου

Ο όρος **Συστήματα Διαχείρισης Περιεχομένου**, ή αλλιώς **CMS** (*Content Management System*) αναφέρεται στις εφαρμογές που επιτρέπουν στον πελάτη να διαχειρίζεται το δικτυακό του περιεχόμενο, όπως κείμενα, εικόνες, πίνακες κ.λπ., με εύκολο τρόπο, συνήθως παρόμοιο με αυτόν της χρήσης ενός κειμενογράφου. Οι εφαρμογές διαχείρισης περιεχομένου επιτρέπουν την αλλαγή του περιεχομένου χωρίς να είναι απαραίτητες ειδικές γνώσεις σχετικές με τη δημιουργία ιστοσελίδων ή γραφικών, καθώς συνήθως τα κείμενα γράφονται μέσω κάποιων online WYSIWYG ("What You See Is What You Get") html editors, ειδικών δηλαδή κειμενογράφων, παρόμοιων με το MS Word, που επιτρέπουν τη μορφοποίηση των κειμένων όποτε υπάρχει ανάγκη.

Οι αλλαγές του site μπορούν να γίνουν από οποιονδήποτε υπολογιστή που είναι συνδεδεμένος στο Διαδίκτυο, χωρίς να χρειάζεται να έχει εγκατεστημένα ειδικά προγράμματα επεξεργασίας ιστοσελίδων, γραφικών κ.λπ. Μέσω ενός απλού φυλλομετρητή ιστοσελίδων (browser), ο χρήστης μπορεί να συντάξει ένα κείμενο και να ενημερώσει άμεσα το δικτυακό του τόπο.

3.2 Ιστορικά

Το πρώτο παράδειγμα ενός πραγματικού Εργαλείου Διαχείρισης Περιεχομένου προήλθε από την Vignette, με ονομασία StoryServer, γύρω στο 1996. Τα επόμενα χρόνια πολλά πακέτα CMS απελευθερώθηκαν. Μεταξύ το 2000 και το 2005, ο τομέας πέρασε ένα τεράστιο κύμα συγχωνεύσεων και εξαγορών αφήνοντας ένα αριθμό χρηστών χωρίς υποστήριξη, από πακέτα που είχαν εγκαταλειφθεί και δυσκολίες καθώς τα πακέτα συγχωνεύτηκαν.

3.3 Γνωστά συστήματα διαχείρισης περιεχομένου

Υπάρχουν πάρα πολλά συστήματα διαχείρισης περιεχομένου, κάποια δωρεάν προς στους προγραμματιστές και κάποια επί πληρωμή. Παρακάτω

ακολουθεί μία μικρή περιγραφή για τα πιο γνωστά δωρεάν συστήματα διαχείρισης περιεχομένου.

- Joomla

Το **Joomla** είναι ένα ελεύθερο και ανοικτού κώδικα σύστημα διαχείρισης περιεχομένου. Χρησιμοποιείται για τη δημοσίευση περιεχομένου στον παγκόσμιο ιστό (World Wide Web). Είναι γραμμένο σε PHP και αποθηκεύει τα δεδομένα του στη βάση MySQL. Το βασικό χαρακτηριστικό του είναι ότι οι σελίδες που εμφανίζει είναι δυναμικές, δηλαδή δημιουργούνται την στιγμή που ζητούνται. Ένα σύστημα διακομιστή(server) όπως είναι ο Apache λαμβάνει τις αιτήσεις των χρηστών και τις εξυπηρετεί.

- Drupal

Το **Drupal** είναι ένα σύστημα διαχείρισης περιεχομένου ανοικτού/ελεύθερου λογισμικού, γραμμένο στη γλώσσα προγραμματισμού PHP. Το Drupal, όπως πολλά σύγχρονα CMS, επιτρέπει στο διαχειριστή συστήματος να οργανώνει το περιεχόμενο, να προσαρμόζει την παρουσίαση, να αυτοματοποιεί διαχειριστικές εργασίες και να διαχειρίζεται τους επισκέπτες του ιστότοπου και αυτούς που συνεισφέρουν. Παρόλο που υπάρχει μια πολύπλοκη προγραμματιστική διεπαφή, οι περισσότερες εργασίες μπορούν να γίνουν με λίγο ή και καθόλου προγραμματισμό. Το Drupal ορισμένες φορές περιγράφεται ως «υποδομή για εφαρμογές ιστού», καθώς οι δυνατότητές του προχωρούν παραπέρα από τη διαχείριση περιεχομένου, επιτρέποντας ένα μεγάλο εύρος υπηρεσιών και συναλλαγών.

- Wordpress

Το **WordPress** είναι ελεύθερο και ανοικτού κώδικα λογισμικό ιστολογίου και πλατφόρμα δημοσιεύσεων, γραμμένο σε PHP και MySQL. Έχει πολλές δυνατότητες, συμπεριλαμβανομένων μιας αρχιτεκτονικής για πρόσθετες λειτουργίες, και ενός συστήματος προτύπων. Χρησιμοποιείται σε περισσότερα από το 14% των 1.000.000 μεγαλύτερων ιστοτόπων. Κυκλοφόρησε για πρώτη φορά στις 27 Μαΐου 2003, από τον Matt Mullenweg ως παραλλαγή του

b2/cafelog. Από τον Φεβρουάριο του 2011, η έκδοση 3.0 έχει ληφθεί περισσότερες από 32,5 εκατομμύρια φορές

- *PhpBB*

Το **PhpBB** είναι ένα διαδικτυακό πακέτο για τη δημιουργία forum και έχει γραφτεί σε γλώσσα PHP. Το όνομα PhpBB είναι τα αρχικά των PHP Bulletin Board. Είναι ελεύθερο λογισμικό και ο κώδικας του είναι ανοιχτός προς όλους. Υποστηρίζει πολλές μηχανές βάσεων δεδομένων όπως Mysql, PostgreSQL, SQLite, Oracle Database, και υπάρχουν αρκετά έτοιμα Plugins που μπορούν να ενσωματωθούν στο PhpBB και να χρησιμοποιηθούν για διάφορες λειτουργίες όπως μαζική αποστολή email, κλείδωμα χρήστη, ομαδοποίηση χρηστών κ.α.

ΜΕΡΟΣ Β

Κεφάλαιο 4

Μεταφόρτωση πρώτης πτυχιακής

Στο δεύτερο μέρος θα γίνει η πλήρης περιγραφή των βημάτων που ακολουθήθηκαν για την επιτυχή μεταφόρτωση, καθώς και τα προβλήματα που συναντήθηκαν και πως αντιμετωπίστηκαν. Έτσι εάν κάποιος στο μέλλον θελήσει να πραγματοποιήσει μεταφόρτωση παρόμοιων πτυχιακών, αρκεί να ακολουθήσει αυτά τα βήματα.

Η πρώτη πτυχιακή που μεταφορτώθηκε ήταν η Πτυχιακή Εργασία του φοιτητή Χατζή Αντωνίου, με τίτλο “Οδηγός δημιουργίας ιστοσελίδων με λογισμικό ανοιχτού κώδικα”.

Πριν ξεκινήσει η παρουσίαση της μεταφόρτωσης να σημειωθεί πως αρχικά προηγήθηκαν κάποιες δοκιμές ώστε να αποκτηθεί η κατάλληλη γνώση και εμπειρία για τις προϋποθέσεις που απαιτούνται για μια τέτοια μεταφόρτωση. Είναι προφανές πως όλες αυτές οι δοκιμές δε θα μπορούσαν να γίνουν στον λογαριασμό του καθηγητή για ευνόητους λόγους.

Οι δοκιμές αυτές πραγματοποιήθηκαν στον προσωπικό λογαριασμό για δύο λόγους:

1. Ήταν πιο ασφαλές, επειδή ήταν τελείως άγνωστο το αντικείμενο
2. Μόνο εκεί υπήρχε πρόσβαση

Το μόνο που ζητήθηκε από τον διαχειριστή του εξυπηρετητή ήταν να αλλαχτούν οι ιδιοκτήτες από τα νέα αρχεία ώστε να είναι εφικτές οι επόμενες ενέργειες που ακολούθησαν. Όλες οι εντολές που περιγράφονται παρακάτω χρησιμοποιήθηκαν αρχικά για την μεταφόρτωση της πτυχιακής στον προσωπικό λογαριασμό. Εφόσον πραγματοποιήθηκε με επιτυχία και ήταν γνωστές οι ενέργειες που απαιτούνταν, μπορούσε να ξεκινήσει η τελική μεταφόρτωση από τον προσωπικό λογαριασμό στον λογαριασμό του καθηγητή.

4.1 Περιγραφή πτυχιακής

Η πτυχιακή αυτή εργασία είχε στόχο να δώσει τις κατάλληλες πληροφορίες και οδηγίες σε ό,τι αφορά τα Συστήματα Διαχείρισης Περιεχομένου. Κύριο μέλημα ήταν να βοηθηθούν οι φοιτητές του τμήματος Μηχανικών Πληροφορικής του ΑΤΕΙΘ στη δημιουργία της δικής τους προσωπικής ιστοσελίδας μέσω CMS. Είναι με λίγα λόγια ένας πλήρης οδηγός δημιουργίας ιστοσελίδων με τη χρήση CMS.

Το κυρίως μέρος της πτυχιακής αποτελούνταν από πέντε μέρη. Τα 4 πρώτα αφορούσαν τις οδηγίες για δημιουργία ιστοσελίδας χρησιμοποιώντας Joomla, Wordpress, Drupal και PhpBB, ενώ το πέμπτο μέρος σχετιζόταν με την περιγραφή των βημάτων για τον τρόπο σύνδεσης στο server της σχολής.

Ο φοιτητής, είχε δημιουργήσει μία ιστοσελίδα όπου περιέγραφε αναλυτικά τα βήματα για τη δημιουργία ιστοσελίδας χρησιμοποιώντας το κάθε CMS από αυτά που προαναφέρθηκαν. Επιπλέον είχε δημιουργήσει ένα site για κάθε CMS, για λόγους επίδειξης.

Η μεταφόρτωση της πρώτης πτυχιακής αφορά στην ουσία τη μεταφορά της κεντρικής σελίδας που δημιουργήθηκε από το φοιτητή, δηλαδή τις οδηγίες για κάθε CMS, αλλά και τη μεταφορά των ενδεικτικών ιστοσελίδων που δημιούργησε με τη χρήση των CMS που αναφέρθηκαν.

4.2 Διαδικασία Μεταφόρτωσης

Από τη στιγμή που γίνεται λόγος για μεταφορά αρχείων στο server, τότε προφανώς θα πρέπει να γίνει μία μικρή αναφορά στα δικαιώματα των αρχείων. Εφόσον πρόκειται για διαδικτυακή ιστοσελίδα τότε είναι λογικό πως θα μπορεί ο καθένας να έχει πρόσβαση σε αυτήν και να την βλέπει. Επομένως πρέπει να τηρούνται κάποιοι κανόνες. Τα αρχεία της ιστοσελίδας που βλέπουν οι επισκέπτες προφανώς θα έχουν το κατάλληλο δικαίωμα ανάγνωσης, όμως δεν θα πρέπει να μπορούν να το επεξεργαστούν και να γράψουν σε αυτό. Άρα τα δικαιώματα που θα πρέπει να έχουν τα αρχεία είναι τα εξής:

*Δικαίωμα ανάγνωσης-εκτέλεσης από απλούς χρήστες
Δικαίωμα εγγραφής-ανάγνωσης-εκτέλεσης από εξουσιοδοτημένους χρήστες
(δηλαδή οι διαχειριστές της ιστοσελίδας).*

Τα συγκεκριμένα δικαιώματα έχουν αυτόν το συμβολισμό:

rwx r-x r-x

όπου η πρώτη τριάδα αφορά τους εξουσιοδοτημένους χρήστες, η δεύτερη ομάδες χρηστών, και η τρίτη απλούς χρήστες.

Η πρώτη ενέργεια που έπρεπε να γίνει, εφόσον έγινε κατανοητή η πτυχιακή και το πως είναι δομημένη, ήταν να δημιουργηθεί ο φάκελος προορισμού, εκεί δηλαδή που θα τοποθετούνταν τα νέα αρχεία της πτυχιακής. Στη συγκεκριμένη περίπτωση έπρεπε να δημιουργηθεί ένας φάκελος στον προσωπικό χώρο του επιβλέποντα καθηγητή. Να σημειωθεί ότι ο φάκελος έπρεπε να είχε δικαίωμα εγγραφής από εξωτερικούς χρήστες για να είναι δυνατή η αντιγραφή των αρχείων.

Σε αυτό το σημείο είναι αναγκαίο να αναφερθεί πως η όλη διαδικασία της μεταφόρτωσης έγινε με τη χρήση γραμμής εντολών κατά τη διάρκεια απομακρυσμένης σύνδεσης με το server της σχολής. Το λειτουργικό σύστημα που χρησιμοποιείται από το server είναι UNIX (το πιο ευρέως χρησιμοποιούμενο λειτουργικό σύστημα σε server). Επομένως οι εντολές που χρησιμοποιήθηκαν και αναφέρονται αφορούν το λειτουργικό σύστημα UNIX.

Ζητήθηκε λοιπόν από τον επιβλέποντα καθηγητή η δημιουργία του φακέλου προορισμού με τα κατάλληλα δικαιώματα. Η ενέργεια αυτή πραγματοποιήθηκε με την εντολή:

*mkdir chatzis
chmod 777 chatzis*

Ο αριθμός 777 είναι ένας άλλος τρόπος που μπορεί να ορίσει κάποιος τα δικαιώματα σε αρχεία. Στην ουσία δημιουργήθηκε ένας φάκελος με όλα τα δικαιώματα για όλους τους τύπους χρηστών.

Εφόσον δημιουργήθηκε ο φάκελος προορισμός των αρχείων της πτυχιακής, σειρά είχε η πρώτη απόπειρα μεταφοράς των αρχείων της κεντρικής ιστοσελίδας της πτυχιακής. Ουσιαστικά επρόκειτο για μια απλή στατική ιστοσελίδα που αποτελούνταν από HTML αρχεία, αρχεία CSS για τη μορφοποίησή της, εικόνες, ένα μενού που είχε γραφτεί σε γλώσσα ActionScript και κάποιο κώδικα JavaScript για μια στοιχειώδη αλληλεπίδραση με το χρήστη.

Από τη στιγμή που δεν υπήρχε κάποια βάση δεδομένων στην οποία να συνδέεται η σελίδα, μπορούσε να προχωρήσει κανονικά η αντιγραφή των αρχείων στον φάκελο προορισμού. Επειδή ο αριθμός των αρχείων αυτών ήταν σχετικά μεγάλος, δεν ήταν δυνατόν να γνωρίζει κάποιος τα ακριβή δικαιώματά τους, τα οποία προφανώς πρέπει να διατηρηθούν ώστε τα τελικά αρχεία να είναι ίδια με τα αρχικά. Η εντολή που χρησιμοποιήθηκε για αυτήν την ενέργεια ήταν:

```
cp -r -p public_html/* /home/staff/tech/dima/public_html/ptyxiakes/chatzis/
```

Η εντολή **cp** χρησιμοποιείται για αντιγραφή αρχείων. Αρχικά ορίζεται ποιο αρχείο θέλει ο χρήστης να αντιγράψει (στη συγκεκριμένη περίπτωση θέλουμε να αντιγράψουμε όλα τα αρχεία που βρίσκονται μέσα στον φάκελο public_html) και έπειτα ορίζουμε τον προορισμό του αντεγραμμένου αρχείου. Τα ορίσματα **-r** και **-p** σημαίνουν τα εξής:

- **-r**: να αντιγραφούν και οι φάκελοι/υποφακέλοι με τα περιεχόμενά τους
- **-p**: τα αντεγραμμένα αρχεία να έχουν ακριβώς τα ίδια δικαιώματα με τα αρχικά αρχεία.

Ο φοιτητής είχε τοποθετήσει στον φάκελο public_html τα αρχεία της στατικής ιστοσελίδας που δημιούργησε, καθώς και τα αρχεία που αφορούσαν τις ενδεικτικές ιστοσελίδες που δημιούργησε με τα CMS που αναφέρθηκαν στην αρχή του κεφαλαίου. Με την παραπάνω εντολή πραγματοποιήθηκε η αντιγραφή όλων αυτών των αρχείων χωρίς όμως να σημαίνει ότι τελείωσε και η μεταφόρτωση. Ακόμα και στην μεταφορά των στατικών HTML αρχείων, παρουσιάστηκαν κάποια προβλήματα, τα οποία όμως θα αναλυθούν σε επόμενο κεφάλαιο. Στη συνέχεια ακολουθεί η περιγραφή των βημάτων για την

ολοκλήρωση της μεταφόρτωσης των ιστοσελίδων που δημιουργήθηκαν με τη χρήση των CMS.

Joomla

Μία ιστοσελίδα που φτιάχνεται με Joomla χρησιμοποιεί μία βάση δεδομένων. Επομένως για να μεταφορτώσει κάποιος μία τέτοια ιστοσελίδα θα πρέπει πέρα από την αντιγραφή των αρχείων, να δημιουργήσει ένα αντίγραφο της βάσης στην οποία συνδέεται η ιστοσελίδα και να συνδέσει τα νέα αρχεία με την νέα αντεγραμμένη βάση. Στο server της σχολής η βάση που χρησιμοποιείται είναι η Mysql, επομένως για να γίνει η αντιγραφή της βάσης του Joomla πρέπει να χρησιμοποιηθεί η παρακάτω εντολή:

```
mysqldump -hostname -u username -ppassword databasename > copied_database.sql
```

Εφόσον έγινε η εξαγωγή του sql αρχείου της βάσης του Joomla, έπρεπε στη συνέχεια να μεταφερθεί στον φάκελο προορισμού των αρχείων της ιστοσελίδας, το οποίο έγινε με την εντολή:

```
cp copied_database.sql /home/staff/tech/dima/public_html/ptyxiakes/chatzis/
```

Στη συνέχεια έπρεπε να δημιουργηθεί η βάση όπου θα συνδεθεί το νέο Joomla site. Η εντολή είναι:

```
create database databasename
```

Έπειτα χρειαζόταν να περαστούν τα δεδομένα στην νέα βάση ώστε να προκύψει το πλήρες αντίγραφο και στη συνέχεια να γίνει η τελική σύνδεση νέων αρχείων – νέας βάσης.

```
mysql -hostname -u username -ppassword databasename < copied_database.sql
```

Με την παραπάνω εντολή ολοκληρώθηκε η αντιγραφή της βάσης. Το μόνο που έμεινε ήταν να γίνει η σύνδεση.

Οι παραπάνω διαδικασίες είναι ακριβώς ίδιες και για τα υπόλοιπα CMS. Για το λόγο αυτό δεν θα γίνει καμία αναφορά για το πως δημιουργείται πλήρες αντίγραφο της βάσης καθώς η όλη διαδικασία γίνεται μέσω γραμμής εντολών στο server.

Για να συνδεθεί μία ιστοσελίδα που έχει φτιαχτεί με Joomla με μία βάση δεδομένων υπάρχουν δύο τρόποι. Ο πρώτος τρόπος είναι να δοθούν τα κατάλληλα στοιχεία μέσω του γραφικού περιβάλλοντος που δίνει το ίδιο το CMS. Δεν χρειάζεται κάποιος δηλαδή να έχει προγραμματιστικές γνώσεις για να χρησιμοποιήσει τον πρώτο τρόπο.

Ο δεύτερος τρόπος (ο οποίος χρησιμοποιήθηκε και για την τελική σύνδεση) είναι ο προγραμματιστικός τρόπος. Στα αρχεία του Joomla υπάρχει ένα αρχείο ονομάζεται configuration.php. Σε αυτό το αρχείο βρίσκονται κάποιες μεταβλητές που χρησιμοποιούνται για την επεξεργασία κάποιων αρχικών ρυθμίσεων στο CMS. Σε αυτό το αρχείο λοιπόν βρίσκονται και οι μεταβλητές που έχουν να κάνουν με τη σύνδεση του CMS με τη βάση δεδομένων. Οι μεταβλητές αυτές είναι:

- *dbtype*
Ο τύπος της βάσης (συνήθως MYSQL ή MYSQLi). Στη δικιά μας περίπτωση MYSQL.
- *host*
το όνομα του διακομιστή όπου βρίσκεται η βάση δεδομένων ,
(:/home/staff/tech/dima/mysql/run/mysql.sock) στη συγκεκριμένη περίπτωση.
- *user*
username, όπου για τη συγκεκριμένη μεταφόρτωση χρησιμοποιήθηκε το root (που είναι το default username για mysql βάση δεδομένων).
- *password*
το password της βάσης για την επιτυχή σύνδεση
- *db*
το όνομα της νέας αντεγραμμένης βάσης δεδομένων,
(chatzis_joomla).
- *dbprefix*
το πρόθεμα που έχουν τα ονόματα των πινάκων που βρίσκονται στην νέα βάση, (dxuno_). Τα προθέματα αποδείχτηκαν πάρα πολύ χρήσιμα καθώς βοήθησαν στην επίλυση ενός πολύ σοβαρού προβλήματος. Το πρόβλημα

παρουσιάζεται στο κεφάλαιο “Προβλήματα που παρουσιάστηκαν”.

Από τη στιγμή που δόθηκαν τα κατάλληλα στοιχεία στις παραπάνω μεταβλητές, έχει σχεδόν ολοκληρωθεί η μεταφορά του Joomla site. Στο κεφάλαιο “Ολοκλήρωση μεταφόρτωσης” περιγράφεται το τελευταίο βήμα που χρειάστηκε.

Drupal

Όπως και στην περίπτωση του Joomla έτσι και στο Drupal υπάρχουν δύο τρόποι για να συνδεθεί η ιστοσελίδα με την αντίστοιχη βάση δεδομένων, είτε μέσω του γραφικού περιβάλλοντος που προσφέρει, είτε μέσω της επεξεργασίας του κατάλληλου αρχείου, βάζοντας τις κατάλληλες τιμές στις κατάλληλες μεταβλητές. Το αρχείο βρίσκεται στην τοποθεσία sites/default και είναι το settings.php. Στο αρχείο αυτό βρίσκεται ο κώδικας για τη διαχείριση αρκετών ρυθμίσεων του Drupal. Ένα τμήμα του αρχείου αφορά τη σύνδεση του CMS με τη βάση δεδομένων, (ο συγκεκριμένος κώδικας βρίσκεται στο παράρτημα).

Όπως και στο Joomla δόθηκαν οι εξής τιμές:

‘database’ => ‘chatzis_drupal’

‘username’ => ‘root’

‘password’ => ‘το κατάλληλο password’

‘host’ => ‘localhost’

διότι η βάση βρίσκεται τοπικά

‘unix_socket’ => ‘/home/staff/tech/dima/mysql/run/mysql.sock’

Σε αντίθεση με το Joomla στο Drupal χρειάζεται να ορίσουμε και όνομα διακομιστή (hostname), δηλαδή localhost στην προκειμένου περίπτωση, αλλά και τη διαδρομή στο mysql socket αν χρησιμοποιούμε, δηλαδή home/staff/tech/dima/mysql/run/mysql.sock.

‘driver’ => ‘mysql’

Ο τύπος της βάσης που χρησιμοποιείται (συνήθως MSQL).

PhpBB

Σε ότι αφορά τη μεταφόρτωση της ενδεικτικής ιστοσελίδας που δημιουργήθηκε μέσω PhpBB, υπήρξε ένα πολύ σοβαρό πρόβλημα, που εμπόδιζε όχι μόνο τη μεταφόρτωση της ίδιας ιστοσελίδας, αλλά και τη μεταφόρτωση όλων των ιστοσελίδων που δημιουργήθηκαν με τα υπόλοιπα CMS. Το πρόβλημα παρουσιάζεται σε επόμενο κεφάλαιο. Για το λόγο αυτό δημιουργήθηκε από την αρχή ένα στοιχειώδης forum για ώστε να είναι δυνατή η επίδειξη μεταφόρτωσης ενός PhpBB forum.

Όπως και στις δύο παραπάνω περιπτώσεις ο πρώτος τρόπος σύνδεσης με τη βάση δεδομένων είναι μέσω του γραφικού περιβάλλοντος και ο δεύτερος είναι μέσω της επεξεργασίας του αρχείου config.php. Οι τιμές που δόθηκαν στις κατάλληλες μεταβλητές ήταν:

```
$dbms = 'mysql';  
$dbhost = ':/home/staff/tech/dima/mysql/run/mysql.sock';  
$dbport = '';  
$dbname = 'chatzis_phpbb';  
$dbuser = 'root';  
$dbpasswd ='το κατάλληλο password';  
$table_prefix = 'phpbb_';
```

Wordpress

Το Wordpress προσφέρει ένα γραφικό περιβάλλον στον διαδίκτυο, στο οποίο μπορεί ο καθένας να δημιουργήσει μια ιστοσελίδα, χωρίς να κατεβάσει το ίδιο το CMS. Το μόνο που χρειάζεται είναι να κάνει μία εγγραφή. Συνήθως το Wordpress χρησιμοποιείται για τη δημιουργία blog. Βέβαια αν κάποιος θέλει να δημιουργήσει μία ολοκληρωμένη διαδικτυακή υπηρεσία, μπορεί να κατεβάσει το CMS, να το κάνει εγκατάσταση, είτε τοπικά είτε σε κάποιο server και να δουλέψει όπως ακριβώς θα δούλευε με Joomla, Drupal, phpbb.

Η σελίδα που δημιουργήθηκε μέσω Wordpress δεν έγινε τοπικά στο server αλλά online. Επομένως δεν πραγματοποιήθηκε μεταφόρτωση της

ιστοσελίδας που δημιουργήθηκε με wordpress, διότι τα αρχεία αυτά δεν βρίσκονταν στο server.

Αν το είχε κατεβάσει και δημιουργούσε την ιστοσελίδα στον ίδιο χώρο με τις υπόλοιπες, τότε σε περίπτωση που κάποιος ήθελε να μεταφορτώσει την ιστοσελίδα του θα πρέπει να ακολουθήσει ακριβώς την ίδια λογική που αναφέρθηκε στις μεταφορτώσεις των προηγούμενων CMS. Το αρχείο που πρέπει να επεξεργαστεί ο προγραμματιστής είναι το wp-config.php, βάζοντας τα κατάλληλα στοιχεία στο εξής σημείο:

```
define('DB_NAME', 'το όνομα της βάσης');  
define('DB_USER', 'το username για τη σύνδεση');  
define('DB_PASSWORD', 'το κατάλληλο password');  
define('DB_HOST', 'το όνομα του διακομιστή');
```

4.3 Ολοκλήρωση Μεταφόρτωσης

Το τελευταίο που χρειαζόταν να γίνει ήταν να αλλάξουν τα στοιχεία του ιδιοκτήτη των νέων αρχείων. Η εντολή `cp -r -g` και μεν αντιγράφει αρχεία από μία τοποθεσία σε μία άλλη μαζί με φακέλους και διατηρώντας τα ίδια δικαιώματα, ωστόσο δεν αλλάζει τον ιδιοκτήτη των αρχείων. Έτσι μπορεί τα νέα αρχεία να βρίσκονται πλέον στον φάκελο του επιβλέποντα καθηγητή, συνεχίζουν όμως να έχουν ιδιοκτήτη τον φοιτητή, με αποτέλεσμα να είναι επεξεργάσιμα και προσβάσιμα μόνο από τον ίδιο. Η εντολή που έπρεπε να χρησιμοποιηθεί ήταν η εξής:

Η γενική σύνταξη της εντολής είναι:

```
chown -R ονομα_χρήστη:ομάδα_χρηστών φάκελος_αρχείων
```

όπου στο όνομα χρήστη πρέπει να ορίσουμε τον επιβλέποντα καθηγητή (dima) και στην ομάδα χρηστών την ομάδα στην οποία ανήκει ο νέος χρήστης (tech). Το όρισμα `-R` βοηθάει στο να συμπεριληφθούν φάκελοι-υποφάκελοι και όλα τους τα περιεχόμενα στην εκτέλεση της εντολής.

Η εντολή που χρησιμοποιήθηκε για την παραπάνω ενέργεια είναι

```
chown -R dima:tech chatzis/*
```

Εφόσον έγινε και το τελευταίο βήμα, ολοκληρώθηκε η μεταφόρτωση της πρώτης πτυχιακής. Ο σύνδεσμος όπου υπήρχε αρχικά ήταν στον προσωπικό λογαριασμό του φοιτητή, <http://aetos.it.teithe.gr/~antcha/>. Με την ολοκλήρωση της μεταφόρτωσης, η ιστοσελίδα της πτυχιακής βρίσκεται στον προσωπικό χώρο του επιβλέποντα καθηγητή.

<http://aetos.it.teithe.gr/~dima/ptyxiakes/chatzis/>.

Έτσι όταν σβηστεί ο συγκεκριμένος φοιτητής από το server της σχολής, η πτυχιακή του θα έχει διασωθεί και θα είναι προσπελάσιμη από τον παραπάνω σύνδεσμο.

4.4 Προβλήματα που παρουσιάστηκαν

Οι παραπάνω ενέργειες θα ολοκληρώνονταν με επιτυχία αν ήταν όλα σχεδιασμένα και δομημένα όπως αναμενόταν. Ωστόσο όμως κάτι τέτοιο είναι εξαιρετικά σπάνιο να συμβεί διότι ο κάθε προγραμματιστής εργάζεται και δομεί την δουλειά του με τον δικό του τρόπο. Έτσι ήταν αναμενόμενο πως προβλήματα δε θα έλειπαν. Τα προβλήματα που συναντήθηκαν ήταν τα εξής:

1. Απόλυτα-σχετικά URL's
2. Πρόβλημα αντιγραφής βάσεων δεδομένων
3. Δημιουργία Keep Alive Script

Απόλυτα-σχετικά URL's

Το πρώτο πρόβλημα παρουσιάστηκε μετά την αντιγραφή των αρχείων της στατικής ιστοσελίδας του φοιτητή στον φάκελο του επιβλέποντα καθηγητή, ήταν το γεγονός ότι ο φοιτητής είχε χρησιμοποιήσει παντού απόλυτα Url's. Παρακάτω υπάρχει ένα παράδειγμα απόλυτου-σχετικού url για την καλύτερη κατανόηση της ορολογίας:

Απόλυτο URL

<http://aetos.it.teithe.gr/~anctha/info.html>

Σχετικό URL

[./info.html](#)

Η διαφορά στα δύο παραπάνω links είναι ότι, αν υπάρχει ο πρώτος υπερσύνδεσμος στην σελίδα και τον επιλέξει ο χρήστης, τότε θα κατευθυνθεί

στη σελίδα info.html που βρίσκεται στον προσωπικό χώρο του φοιτητή, ανεξάρτητα από το βρίσκεται ο ίδιος εκείνη τη στιγμή στο διαδίκτυο. Αν όμως υπάρχει ο δεύτερος, τότε αν τον επιλέξει ο χρήστης θα κατευθυνθεί στη σελίδα info.html που βρίσκεται στην τρέχουσα τοποθεσία του χρήστη. Έτσι στην πρώτη θα περίπτωση ο χρήστης θα κατευθυνόταν αποκλειστικά στη σελίδα <http://aetos.it.teithe.gr/~antcha/info.html>, ενώ στη δεύτερη περίπτωση θα κατευθυνόταν στην σελίδα:

<http://aetos.it.teithe.gr/~dima/ptyxiakes/chatzis/info.html>

αν βρισκόταν ήδη στην τοποθεσία

<http://aetos.it.teithe.gr/~dima/ptyxiakes/chatzis/>

Παρόμοιο πρόβλημα υπήρξε και με τη χρήση εικόνων. Ο φοιτητής είχε χρησιμοποιήσει απόλυτους συνδέσμους και στη χρήση εικόνων, με αποτέλεσμα αν κάποιος βρισκόταν στην τοποθεσία:

<http://aetos.it.teithe.gr/~dima/ptyxiakes/chatzis/>

και στο αρχείο index.html υπήρχε η σήμανση της HTML, τότε το source της εικόνας θα ήταν <http://aetos.it.teithe.gr/~antcha/images/image1.jpg>. Έτσι αν χανόταν η εικόνα από το χώρο του φοιτητή τότε δε θα ήταν ορατή από τον χρήστη που βρισκόταν στην τοποθεσία με τα αντεγραμμένα αρχεία. Για να ήταν ορατή η εικόνα έπρεπε το source της να είναι ./images/image1.jpg/

Για να ξεπεραστούν τα παραπάνω προβλήματα των απολύτων url, έπρεπε να μετατραπούν όλοι οι απόλυτοι υπερσύνδεσμοι σε σχετικούς. Η ενέργεια αυτή πραγματοποιήθηκε με τη βοήθεια του προγράμματος επεξεργασίας NotePad++ μέσω της λειτουργικότητας Replace with που προσφέρει. Η λειτουργία αυτή μπορεί να πραγματοποιηθεί βέβαια με οποιοδήποτε πρόγραμμα επεξεργασίας κειμένου. Έτσι έπρεπε να αντικατασταθεί η συμβολοσειρά 'http://aetos.it.teithe.gr/~antcha/' με την συμβολοσειρά './'. Αυτό είχε σαν αποτέλεσμα το εξής:

Ο σύνδεσμος <http://aetos.it.teithe.gr/~anctha/info.html> να γίνει ./info.html

και τα source των εικόνων να γίνουν από
<http://aetos.it.teithe.gr/~antcha/images/image1.jpg> σε
`./images/image1.jpg`

Στο menu υπήρχε επίσης η χρήση των απολύτων Url που είχε δημιουργήσει με Action Script με αποτέλεσμα, όποιον σύνδεσμο και να επέλεγε ο χρήστης να κατευθυνόταν στον χώρο του φοιτητή αντί να μένει στην τοποθεσία που βρισκόταν. Και επειδή το menu ήταν ένα εκτελέσιμο αρχείο Action Script έπρεπε να βρεθεί τρόπος αποκωδικοποίησης του ώστε να επεξεργαστεί ο πηγαίος κώδικας και να αλλάξουν οι σύνδεσμοι. Το πρόβλημα λύθηκε με τη χρήση του προγράμματος *Sothink SWF Decompiler*.

Η χρήση των απολύτων URL συνεχίστηκε και στην δημιουργία των ιστοσελίδων μέσω CMS. Έτσι αν ο χρήστης ήθελε να δει το ενδεικτικό Joomla site, και επέλεγε κάποιο σύνδεσμο, τότε θα κατευθυνόταν και πάλι στον προσωπικό του χώρο του φοιτητή. Και σε αυτήν την περίπτωση το πρόβλημα λύθηκε με τον ίδιο ακριβώς τρόπο. Έπρεπε και στα δύο sql αρχεία (Joomla, Drupal), να εφαρμοστεί η ίδια ενέργεια όπως και παραπάνω, replace <http://aetos.it.teithe.gr/~antcha/joomla/> with `./`. Όταν κάποιος δημιουργεί μία ιστοσελίδα με ένα CMS τότε όλα τα περιεχόμενα της ιστοσελίδας αποθηκεύονται στη βάση στην οποία είναι συνδεδεμένη η ιστοσελίδα, συνεπώς και όλοι οι υπερσύνδεσμοι που χρησιμοποιούνται. Όπως έχει προαναφερθεί, δεν πραγματοποιήθηκε μεταφόρτωση του Wordpress και του PhpBB.

Πρόβλημα αντιγραφής βάσης δεδομένων

Ένα άλλο αρκετά σοβαρό πρόβλημα που παρουσιάστηκε αφορούσε την αντιγραφή των βάσεων δεδομένων των CMS.

Όταν κάποιος δουλεύει με πολλά διαφορετικά συστήματα διαχείρισης περιεχομένου, τότε προφανώς θα γνωρίζει ότι για την καλύτερη οργάνωση, θα ήταν καλύτερο να χρησιμοποιείται μία βάση δεδομένων για κάθε CMS. Δηλαδή στη συγκεκριμένη περίπτωση το ιδανικό θα ήταν να υπήρχε μία βάση δεδομένων για το Joomla site, μία άλλη βάση δεδομένων για το Drupal site και μία τρίτη βάση για το PhpBB forum.

Ο φοιτητής, είχε χρησιμοποιήσει μία βάση και για τα 3 CMS με αποτέλεσμα να δημιουργηθεί μία βάση με 366 πίνακες. Ο λόγος που χρησιμοποίησε μία βάση βρίσκεται στο παράρτημα στις οδηγίες χρήσης της Mysql στο server της σχολής. Επιπλέον ίσως είχε χρησιμοποιήσει κάποια έτοιμη βάση όσον αφορά τη δημιουργία του PhpBB forum, κάτι το οποίο είχε σαν συνέπεια το συγκεκριμένο forum να φαινόταν ότι ήταν ενεργό, και επιπλέον να αυξήσει κατά πολύ το μέγεθος της βάσης δεδομένων (περίπου 5GB). Έτσι, η αντιγραφή της βάσης, που αναφέρθηκε σε προηγούμενο υποκεφάλαιο, ήταν εξαιρετικά δύσκολη καθώς υπήρχε πρόβλημα με τον διαθέσιμο χώρο που έχει ο κάθε χρήστης στο server, αλλά κυρίως ήταν πολύ χρονοβόρο και πολύ επίπονο για το server, αφού κάποιες φορές δεν λειτουργούσε σωστά όταν γινόταν επιχείρηση της αντιγραφής (έπεφτε).

Έπειτα από όλα αυτά κρίθηκε σκόπιμο να γίνει προσπάθεια τεμαχισμού της βάσης σε κομμάτια. Δηλαδή να απομονωθούν οι πίνακες που αφορούν το κάθε CMS σε τρία ξεχωριστά sql αρχεία. Παρατηρήθηκε πως, και μεν ο φοιτητής είχε χρησιμοποιήσει μία βάση και για τα τρία CMS, ωστόσο είχε χρησιμοποιήσει διαφορετικά προθέματα για το Joomla site και διαφορετικά για το PhpBB forum. Επομένως η παραπάνω σκέψη ήταν πλέον εύκολη στην υλοποίησή της. Ο κώδικας που χρησιμοποιήθηκε για την εξαγωγή του sql αρχείου για το κάθε CMS βρίσκεται στο παράρτημα.

Να σημειωθεί πως λόγω του προβλήματος που δημιούργησε το PhpBB, (πολύ μεγάλη βάση) δεν ήταν δυνατή η παραπάνω ενέργεια, και έτσι αποφασίστηκε να δημιουργηθεί από την αρχή ένα ενδεικτικό forum.

Από τη στιγμή λοιπόν που απομονώθηκαν οι πίνακες και δημιουργήθηκαν τα δύο sql αρχεία, συνεχίστηκε η μετατροπή των απόλυτων συνδέσμων σε σχετικούς που περιγράφηκε παραπάνω.

Δημιουργία keep alive script

Επειδή ο φοιτητής χρησιμοποίησε μία βάση για την πτυχιακή του, δε σήμαινε απαραίτητα πως έπρεπε να χρησιμοποιηθεί μία βάση και στην νέα αντεγραμμένη πτυχιακή. Ήταν προτιμότερο να χρησιμοποιηθούν περισσότερες από μία βάσεις, καθώς πέρα από την μεταφόρτωση αυτής της

πτυχιακής, πραγματοποιήθηκε η μεταφόρτωση άλλης μίας πτυχιακής. Επομένως χρησιμοποιήθηκε η λεγόμενη προσωπική Mysql αντί της κεντρικής Mysql, την οποία κεντρική χρησιμοποίησε και ο φοιτητή. Μέσω της προσωπικής Mysql μπορεί κανείς να έχει παραπάνω από μία βάση δεδομένων. Το πρόβλημα σε αυτήν την περίπτωση ήταν όμως το εξής. Αν γίνει επανεκκίνηση στο server τότε η βάση δεν θα μπορεί να χρησιμοποιηθεί, χωρίς όμως να έχει σβηστεί. Το πρόβλημα λύθηκε με την δημιουργία ενός keep alive script που καλείται απο το crontab κάθε 30 λεπτά. Ο κώδικας του συγκεκριμένου script βρίσκεται στο παράρτημα. Ουσιαστικά το συγκεκριμένο script ελέγχει αν τρέχει η mysql. Αν δεν τρέχει την ξαναξεκινάει, χωρίς όμως να σβήνεται καμία από τις βάσεις δεδομένων που υπάρχουν.

Από τη στιγμή που λύθηκαν όλα τα παραπάνω προβλήματα, ολοκληρώθηκε με απόλυτη επιτυχία η μεταφόρτωση της πτυχιακής του φοιτητή στον προσωπικό λογαριασμό του επιβλέποντα καθηγητή.

Κεφάλαιο 5

Μεταφόρτωση δεύτερης πτυχιακής

Η δεύτερη πτυχιακή εργασία που ζητήθηκε από τον επιβλέποντα καθηγητή να μεταφορτωθεί, ήταν η πτυχιακή του φοιτητή Παλάσχα Κωνσταντίνου και είχε τίτλο 'Σύστημα επίβλεψης πτυχιακών εργασιών'. Στη δεύτερη πτυχιακή, πέρα από τη μεταφόρτωσή της ζητήθηκε και η επέκτασή της σε ένα ολοκληρωμένο πληροφοριακό σύστημα, καθώς ο φοιτητής δεν την είχε ολοκληρώσει στο 100% της.

Σε αυτό λοιπόν το διαδικτυακό πληροφοριακό σύστημα ο επιβλέπων καθηγητής θα μπορεί όχι μόνο να παρακολουθεί τις πτυχιακές εργασίες που η εκπόνησή τους βρίσκεται σε εξέλιξη, αλλά και να διατηρεί ιστορικό παλαιότερων αναθέσεων, ολοκληρωμένων και ακυρωμένων.

Όπως και στη μεταφόρτωση της πρώτης πτυχιακής, (ήταν πλέον γνωστός ο τρόπος μεταφόρτωσης μία διαδικτυακής εφαρμογής), έγινε μεταφόρτωση αρχικά της πτυχιακής του φοιτητή στον προσωπικό λογαριασμό του γράφοντα. Αυτά που έπρεπε να γίνουν φυσικά ήταν η αλλαγή των ιδιοκτητών των νέων αρχείων και η σύνδεση των νέων αρχείων με τη νέα βάση. Επομένως όλες οι παρακάτω ενέργειες, η επέκταση και βελτίωση της πτυχιακής πραγματοποιήθηκαν στον προσωπικό λογαριασμό. Όταν ολοκληρώθηκε η υλοποίηση του συστήματος τότε έγινε η τελική μεταφόρτωση από τον προσωπικό λογαριασμό στο λογαριασμό του επιβλέποντα καθηγητή.

5.1 Περιγραφή Πτυχιακής

Σκοπός της πτυχιακής αυτής ήταν η δημιουργία μιας διαδικτυακής εφαρμογής, με την οποία θα γίνεται η επίβλεψη των πτυχιακών εργασιών του τμήματος Πληροφορικής του ΑΤΕΙ Θεσσαλονίκης. Ωστόσο όπως προαναφέρθηκε η πτυχιακή δεν είχε ολοκληρωθεί από το φοιτητή λόγω αλλαγής (από υποχρεωτική σε προαιρετική) της υποβολής χρονοδιαγράμματος και έκθεσης προόδου των πτυχιακών και για το λόγο αυτό ζητήθηκε από τον επιβλέποντα καθηγητή ο επανασχεδιασμός του συστήματος και η προσθήκη κάποιων επιπλέον λειτουργιών.

Ο επιβλέπων καθηγητής διατηρούσε ένα αρχείο Microsoft Access στο οποίο υπήρχαν πληροφορίες για παλαιότερες αναθέσεις πτυχιακών, για ακυρωμένες αναθέσεις, για ολοκληρωμένες αλλά και τις αναθέσεις που ήταν σε εξέλιξη κάθε εξάμηνο. Το γεγονός όμως ότι όλες αυτές οι πληροφορίες βρίσκονταν σε ένα αρχείο τοπικά και όχι στο διαδίκτυο, καθιστούσε ιδιαίτερα δύσκολη την εξυπηρέτησή του, όχι μόνο όταν αναζητούσε πληροφορίες για προσωπικούς λόγους, αλλά και όταν ζητούσαν να μάθουν πληροφορίες άλλοι καθηγητές και υπάλληλοι της γραμματείας. Ο καθηγητής έπρεπε να αναζητεί τις σωστές πληροφορίες από το ενημερωμένο, ή να το ενημερώσει με νέες. Για το λόγο αυτό θεώρησε χρήσιμο όλες αυτές οι πληροφορίες να μεταφερθούν σε μία διαδικτυακή βάση και η όλη εργασία που πραγματοποιούσε τοπικά στο αρχείο access, να την πραγματοποιεί μέσω του νέου πληροφοριακού συστήματος, που είναι η βελτίωση και επέκταση της πτυχιακής του φοιτητή.

5.2 Δημιουργία της διαδικτυακής βάσης

Η πρώτη ενέργεια που ήταν απαραίτητη να γίνει πριν προχωρήσει η διαδικασία της σχεδίασης του συστήματος, ήταν να δημιουργηθεί η διαδικτυακή βάση που θα φιλοξενεί πλέον τα δεδομένα που υπήρχαν στο access αρχείο. Για να γίνει αυτό έπρεπε να γίνει κατανοητή η δομή που είχε το access αρχείο καθώς και τι σχέσεις υπήρχαν μεταξύ των δεδομένων.

Το αρχείο αυτό που διατηρούσε ο καθηγητής αποτελούνταν από μόλις έναν πίνακα με τις εξής στήλες:

Αα	Ο αύξων αριθμός της εγγραφής
Τίτλος	Ο τίτλος της πτυχιακής
Εισηγητής	Ο εισηγητής της πτυχιακής
Ημ_Υποβ	Η ημερομηνία υποβολής της συγκεκριμένης πτυχιακής
Επ_Φοιτητή	Το επώνυμο του Φοιτητή στον οποίο ανατέθηκε η συγκεκριμένη πτυχιακή
Ον_Φοιτητή	Το όνομα του Φοιτητή στον οποίο ανατέθηκε η

	συγκεκριμένη πτυχιακή
ΑΜ_Φοιτητή	Ο αριθμός μητρώου του Φοιτητή στον οποίο ανατέθηκε η συγκεκριμένη πτυχιακή
Εξ_ανάθεσης	Το εξάμηνο που έγινε η ανάθεση
Ημ_έναρ	Η ημερομηνία έναρξης της ανάθεσης
Vs	Αν η πτυχιακή προοριζόταν για διαγωνισμό ή για απευθείας ανάθεση
Ομαδ	Αν η πτυχιακή ήταν ομαδική ή ατομική
Ολο	Αν η πτυχιακή είχε ολοκληρωθεί από το φοιτητή
ΠΣ_ολο	Το πρακτικό συνέλευσης ολοκλήρωσης της πτυχιακής
Ημ_εξέτ	Η ημερομηνία εξέτασης της πτυχιακής
Ακύρ	Αν η πτυχιακή ακυρώθηκε
ΠΣ_ακ	Το πρακτικό συνέλευσης ακύρωσης της πτυχιακής
Παρ	Αν ο φοιτητής πήρε παράταση
ΠΣ_παρ	Το πρακτικό συνέλευσης παράτασης της πτυχιακής
Ζητ	Αν η πτυχιακή ζητήθηκε κατά τη διάρκεια διαγωνισμού αναθέσεων
Δοθ	Αν η πτυχιακή έχει δοθεί σε κάποιον φοιτητή
ΠΣ_αλ_επ	Το πρακτικό συνέλευσης αλλαγής επιβλέποντα της πτυχιακής
Νέος_Επιβ	Ο νέος επιβλέπωντας της πτυχιακής
Σχόλια	Κάποια σχόλια

Εφόσον μελετήθηκε η δομή του αρχείου και έγιναν κατανοητές οι στήλες που είχε, έπρεπε να αποφασιστεί, πόσους πίνακες θα έχει η νέα βάση. Ακολούθησε στη συνέχεια μία συζήτηση με τον καθηγητή και μετά από

κάποιες αναλύσεις και κάποιες κανονικοποιήσεις, αποφασίστηκε πως τέσσερις πίνακες ήταν το ιδανικό για την καλύτερη οργάνωση των πληροφοριών. Αυτοί είναι οι εξής:

1. Foititis
2. Eisigitis
3. Ptyxiaki
4. Anathesi

Ακολουθεί η αναλυτική περιγραφή του κάθε πίνακα, ώστε να γίνει πιο κατανοητή η δομή της διαδικτυακής βάσης.

Foititis

Στον πίνακα αυτόν, όπως λέει και το όνομά του, βρίσκονται όλοι οι φοιτητές του τμήματος πληροφορικής (ενεργοί και απόφοιτοι).

AM	Ο αριθμός μητρώου του φοιτητή
Surname	Το επώνυμο του φοιτητή
Name	Το όνομα του φοιτητή
Username	Το username του φοιτητή
DM	Οι διδακτικές μονάδες του φοιτητή
SP	Ο συντελεστής προόδου του φοιτητή
MO	Ο μέσος όρος του φοιτητή
Examino	Το τρέχων εξάμηνο του φοιτητή
Energos	Αν ο φοιτητής είναι ενεργός ή όχι

Κύριο κλειδί είναι προφανώς ο αριθμός μητρώου (AM), καθώς είναι μοναδικό για τον κάθε φοιτητή. Δεν γίνεται δύο φοιτητές να έχουν τον ίδιο αριθμό μητρώου.

Eisigitis

ID	Ένας κωδικός, ξεχωριστός για κάθε εισιγή
Surname	Το επώνυμο του εισιγητή
Name	Το όνομα του εισιγητή
Username	Το username του εισιγητή

Monimos	Αν ο εισηγητής είναι μόνιμος ή όχι
Energos	Αν ο εισηγητής είναι ενεργός ή όχι

Κύριο κλειδί είναι η στήλη ID, και επομένως κάθε εγγραφή θα έχει διαφορετική τιμή σε αυτή τη στήλη.

Ptyxiaki

Εδώ βρίσκονται οι πτυχιακές που είτε δημοσιεύονται από τους καθηγητές και προορίζονται για τον διαγωνισμό ανάθεσης πτυχιακών, είτε έχουν δοθεί με απευθείας ανάθεση σε κάποιον-κάποιους φοιτητές.

ID	Ένας κωδικός, ξεχωριστός για κάθε πτυχιακή
Id_eisigiti	Ο κωδικός του εισηγητή της πτυχιακής
Title	Ο τίτλος της πτυχιακής
Date_submit	Η ημερομηνία υποβολής της πτυχιακής
Semester_start	Το εξάμηνο ανάθεσης της πτυχιακής
Team	Αν η πτυχιακή είναι ομαδική ή ατομική
VS	Αν η πτυχιακή προορίζεται για διαγωνισμό ή για απευθείας ανάθεση
Asked	Αν η πτυχιακή ζητήθηκε από κάποιον φοιτητή κατά τη διάρκεια του διαγωνισμού
Given	Αν η πτυχιακή έχει δοθεί σε κάποιον φοιτητή

Κύριο κλειδί είναι προφανώς το πεδίο ID. Το πεδίο Id_eisigiti είναι ξένο κλειδί και αντιστοιχεί στο ID στον πίνακα Eisigitis.

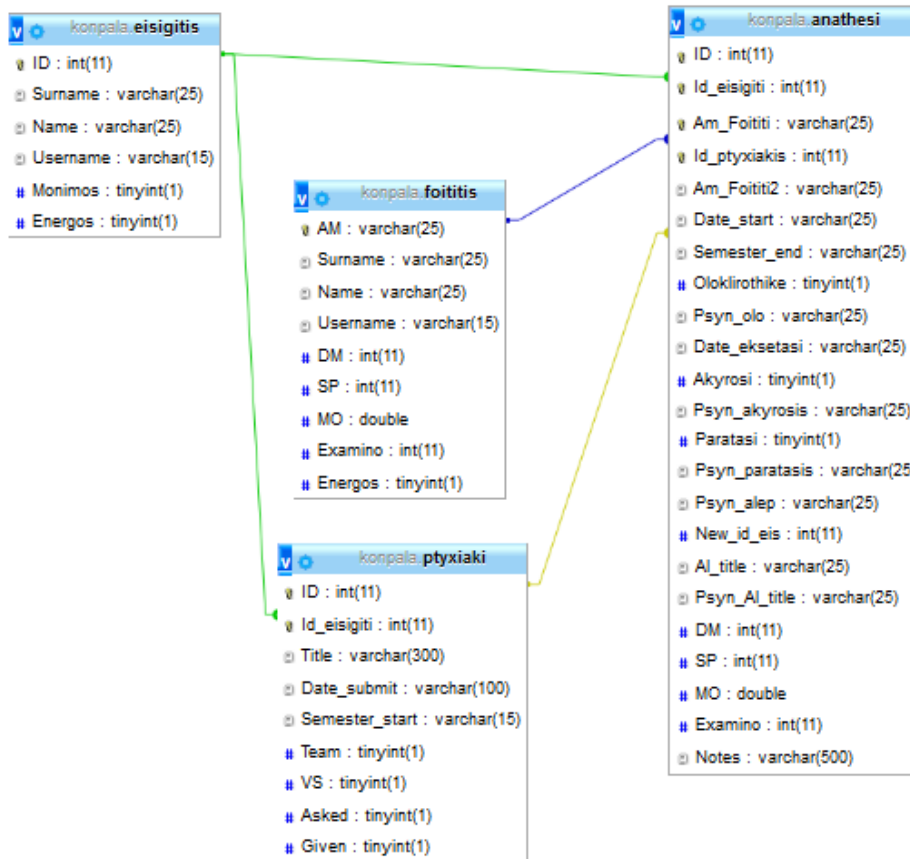
Anthesi

ID	Ένας κωδικός, ξεχωριστός για κάθε ανάθεση
Id_eisigiti	Ο κωδικός του εισηγητή της πτυχιακής
Am_Foititi	Ο αριθμός μητρώου του φοιτητή, στον οποίο ανατέθηκε η πτυχιακή
Id_ptyxiakis	Ο κωδικός της πτυχιακής

Am_Foititi2	Ο αριθμός μητρώου του δεύτερου φοιτητή σε περίπτωση ομαδικής πτυχιακής
Date_start	Η ημερομηνία έναρξης της ανάθεσης
Semester_end	Το εξάμηνο ολοκλήρωσης της πτυχιακής
Oloklirothike	Αν η πτυχιακή έχει ολοκληρωθεί
Psyn_olo	Το πρακτικό συνέλευσης ολοκλήρωσης
Date_eksetasi	Η ημερομηνία εξέτασης της πτυχιακής
Akyrosi	Αν η πτυχιακή έχει ακυρωθεί
Psyn_akyrosis	Το πρακτικό συνέλευσης ακύρωσης
Paratasi	Αν ο φοιτητής πήρε παράταση
Psyn_paratasis	Το πρακτικό συνέλευσης παράτασης
Psyn_alep	Το πρακτικό αλλαγής επιβλέποντα
New_id_eis	Ο κωδικός του νέου εισηγητή
AI_title	Ο νέος τίτλος της πτυχιακής
Psyn_AI_title	Το πρακτικό συνέλευσης αλλαγής τίτλου
DM	Οι διδακτικές μονάδες που είχε ο φοιτητής όταν του ανατέθηκε η πτυχιακή
SP	Ο συντελεστής προόδου που είχε ο φοιτητής όταν του ανατέθηκε η πτυχιακή
MO	Ο μέσος όρος που είχε ο φοιτητής όταν του ανατέθηκε η πτυχιακή
Examino	Το εξάμηνο του φοιτητή όταν του ανατέθηκε η πτυχιακή
Notes	Κάποια σχόλια

Όπως και πριν κύριο κλειδί είναι η στήλη ID για τους προφανείς λόγους. Η στήλη Id_eisigiti είναι ξένο κλειδί και αντιστοιχεί στο ID του πίνακα Eisigitis. Επιπλέον ξένο κλειδί είναι και η στήλη New_id_eis που αντιστοιχεί και αυτή στην ίδια στήλη. Επιπλέον ξένα κλειδιά αποτελούν οι στήλες Am_Foititi και Am_Foititi2, που αντιστοιχούν στη στήλη AM στον πίνακα Foititis, και η στήλη Id_ptyx που αντιστοιχεί στη στήλη ID στον πίνακα Ptyxiaki.

Ακολουθεί επίσης και ένα σχεδιάγραμμα της βάσης για να γίνει ακόμα πιο κατανοητό το πως συνδέονται οι παραπάνω πίνακες μεταξύ τους.



Εικόνα 3: Σχεδιάγραμμα της νέας διαδικτυακής βάσης

Να σημειωθεί πως χρησιμοποιήθηκε ένας ακόμη πίνακας που περιέχει τους χρήστες του συστήματος. Ο πίνακας αυτός δημιουργήθηκε από το φοιτητή και αφορά αποκλειστικά και μόνο την ταυτοποίηση των στοιχείων κατά την είσοδο του χρήστη στο σύστημα (log in).

5.3 Ενημέρωση της διαδικτυακής βάσης

Από τη στιγμή που είχε πλέον δημιουργηθεί η διαδικτυακή βάση, σειρά είχε η μεταφορά των δεδομένων από το αρχείο access στη νέα βάση. Το γεγονός όμως ότι η δομή της νέας βάσης είχε αρκετές διαφορές με τη δομή της βάσης που υπήρχε στο αρχείο, πρόσθεσε ένα ακόμη επίπεδο δυσκολίας στην συλλογή των πληροφοριών.

Αυτό που έπρεπε να γίνει για την ενημέρωση της διαδικτυακής βάσης, ήταν η “απομόνωση” των κατάλληλων εγγραφών από το αρχείο, για κάθε πίνακα της νέας βάσης ξεχωριστά. Χρειαζόταν για κάθε πίνακα να εκτελεστεί ένα sql ερώτημα στη βάση του αρχείου, το οποίο θα έδινε το επιθυμητό αποτέλεσμα. Στη συνέχεια έγινε εξαγωγή των εγγραφών σε ένα αρχείο ανά πίνακα (δηλαδή συνολικά 4 αρχεία), στο οποίο οι πληροφορίες ήταν δομημένες με συγκεκριμένο τρόπο. Τέλος μέσω rhp κώδικα έγινε η ενημέρωση του κάθε πίνακα ξεχωριστά χρησιμοποιώντας το αντίστοιχο αρχείο. Ο sql κώδικας που χρησιμοποιήθηκε για κάθε πίνακα, ώστε να ολοκληρωθεί η παραπάνω ενέργεια βρίσκεται στο παράρτημα.

Εφόσον ολοκληρώθηκε με επιτυχία η παραπάνω ενέργεια, σειρά είχε η εξαγωγή τους σε αρχείο, που χρησιμοποιήθηκε για την ενημέρωση των πινάκων της διαδικτυακής βάσης. Η λειτουργία αυτή έγινε με τη βοήθεια του προγράμματος Microsoft Access. Τα αρχεία έπρεπε είχαν την κατάληξη txt και είχαν την εξής δομή. Οι στήλες ήταν χωρισμένες σε το ελληνικό ερωτηματικό (semicolon separated) και όπου υπήρχε αλφαριθμητική πληροφορία ήταν μέσα σε “”. Επιπλέον, αν σε μία στήλη οι πληροφορίες ήταν αληθές ή ψευδές, τότε στο αρχείο στη συγκεκριμένη στήλη υπήρχε 1 ή 0 αντίστοιχα.

Έπειτα εφόσον υπήρχαν τα δεδομένα για κάθε πίνακα, έγινε η ενημέρωσή τους χρησιμοποιώντας το κάθε αρχείο. Η ενημέρωση αυτή έγινε μέσω rhp κώδικα. Συγκεκριμένα δημιουργήθηκε ένα εργαλείο μέσω του οποίου ανέβαινε το αρχείο αυτό σε μία συγκεκριμένη τοποθεσία όπου εκεί, ο κώδικας διάβαζε μία μία γραμμή από το αρχείο και ενημέρωνε τον αντίστοιχο πίνακα. Η διαδικασία αυτή είναι η ίδια και για τους 4 πίνακες. Ελάχιστες διαφορές υπάρχουν στον τρόπο ενημέρωσης των πινάκων που περιέχουν τις Πτυχιακές και τις αναθέσεις. Οι διαφορές αυτές παρουσιάζονται παρακάτω σε επόμενο κεφάλαιο.

5.4 Σχεδίαση της διαδικτυακής διεπαφής

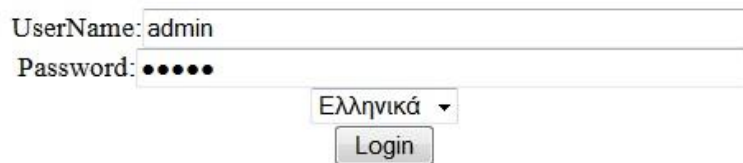
Όπως έχει ήδη αναφερθεί, ο τίτλος της πτυχιακής του φοιτητή ήταν “Σύστημα επίβλεψης Πτυχιακών Εργασιών”. Από τη στιγμή που γίνεται λόγος για ένα σύστημα στο οποίο θα υπάρχει μία λίστα πτυχιακών, μαζί με εισηγητές, φοιτητές, ημερομηνίες και άλλες πληροφορίες άμεσα

συνδεδεμένες, τότε προφανώς ο καλύτερος και πιο κατανοητός τρόπος για να παρουσιαστούν οι παραπάνω πληροφορίες στο χρήστη είναι ο πίνακας. Έτσι σε όλες τις σελίδες του συστήματος χρησιμοποιήθηκε ένας πίνακας για την εμφάνιση των αντίστοιχων πληροφοριών.

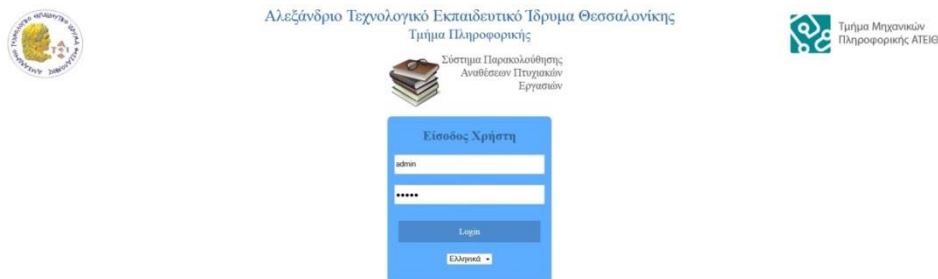
Παρακάτω παρουσιάζονται όλες οι διεπαφές που υπάρχουν στο σύστημα.

Σχεδίαση αρχικής φόρμας και μενού πλοήγησης

Ο φοιτητής είχε ήδη σχεδιάσει μία αρχική φόρμα εισόδου, όπου ο χρήστης, μέσω της αυθεντικοποίησής του, θα έχει πρόσβαση στο σύστημα. Η λειτουργικότητα αυτής της φόρμας φυσικά διατηρήθηκε. Οι αλλαγές που προστέθηκαν αφορούν αποκλειστικά την εμφάνιση.

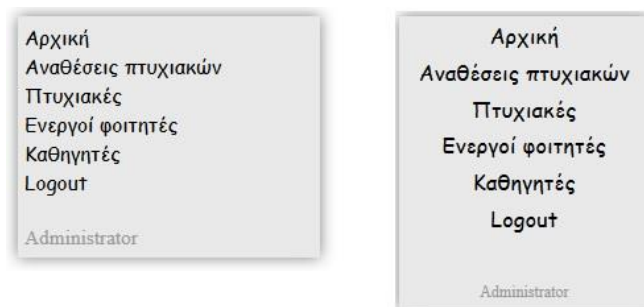


Εικόνα 4 : Αρχική φόρμα εισόδου σχεδιασμένη από το φοιτητή



Εικόνα 5 : Τελική φόρμα εισόδου

Επίσης είχε σχεδιάσει και ένα μενού πλοήγησης στο αριστερό μέρος της οθόνης, το οποίο επίσης διατηρήθηκε με ελάχιστες αλλαγές.



Εικόνα 6 : Στα αριστερά βρίσκεται το μενού πλοήγησης όπως ήταν σχεδιασμένο από τον φοιτητή, ενώ στα δεξιά βρίσκεται το τελικό μενού.

Σχεδίαση διεπαφής φοιτητών

Από τη στιγμή που υπήρχε στη βάση πίνακας που περιείχε όλους τους φοιτητές του τμήματος, τότε προφανώς θα έπρεπε να υπάρχει και μια διεπαφή όπου ο χρήστης θα μπορούσε να δει όλους αυτούς τους φοιτητές μαζί με τις αντίστοιχες πληροφορίες, καθώς και να μπορεί να επεξεργαστεί κάποιον φοιτητή.

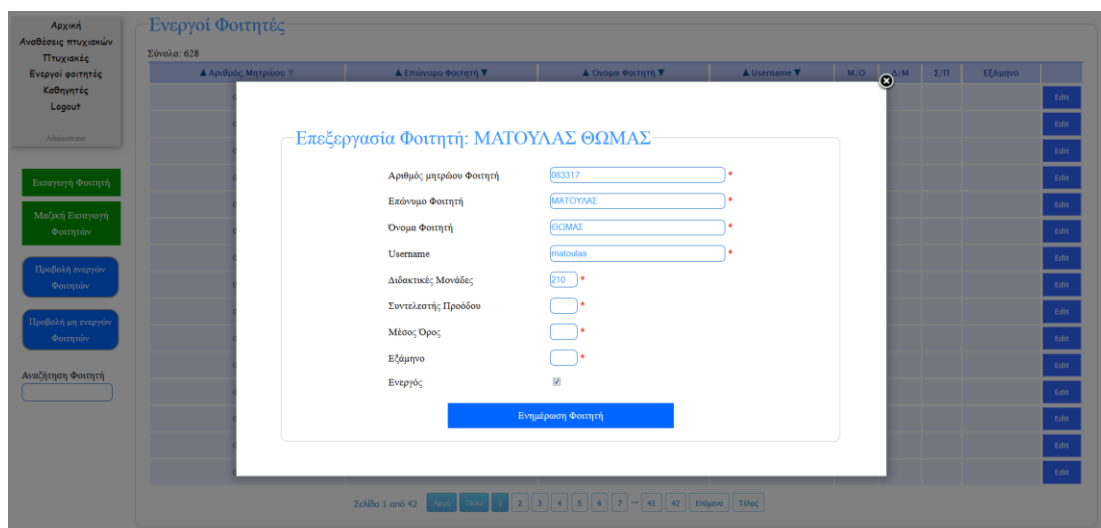
Ενεργοί Φοιτητές
Σύνολο: 628

▲ Αριθμός Μητρώου ▼	▲ Επώνυμο Φοιτητή ▼	▲ Όνομα Φοιτητή ▼	▲ Username ▼	M/O	Δ/Μ	Σ/Π	Εξάμηνο	
001444	ΠΕΡΙΔΟΥ	ΜΑΡΙΟΣ						Edit
001462	ΜΟΥΤΣΟΥΡΛΗΣ	ΓΕΩΡΓΙΟΣ						Edit
001469	ΠΑΝΝΑΚΟΠΟΥΛΟΣ	ΘΕΟΚΛΗΣ						Edit
001481	ΠΑΠΑΔΗΜΗΤΡΙΟΥ	ΓΕΩΡΓΙΟΣ						Edit
001483	ΤΥΧΑΛΑΣ	ΚΩΝ/ΝΟΣ						Edit
001489	ΤΡΙΑΝΤΑΦΥΛΛΟΥ	ΚΩΝ/ΝΟΣ						Edit
001498	ΘΕΟΔΩΡΗΣ	ΚΩΝΣΤΑΝΤΙΝΟΣ						Edit
001533	ΑΣΤΡΟΠΕΚΑΚΗ	ΔΕΣΠΟΙΝΑ						Edit
001547	ΜΙΧΟΣ	ΑΛΕΞΑΝΔΡΟΣ						Edit
001556	ΝΑΣΙΟΥ	ΑΣΤΑΣΙΑ						Edit
001559	ΣΚΑΦΙΔΑΣ	ΑΛΕΞΑΝΔΡΟΣ						Edit
001574	ΜΠΑΜΠΑΤΣΗΣ	ΧΡΗΣΤΟΒΑΛΑΝΤΗΣ						Edit
001577	ΜΑΡΚΟΥ	ΧΡΗΣΤΟΣ						Edit
001582	ΜΑΤΑΜΗΣ	ΠΑΝΑΓΙΩΤΗΣ						Edit
001607	ΤΣΙΚΑΣ	ΣΤΑΥΡΟΣ						Edit

Σελίδα 1 από 42 [Αρχή] [Πίσω] [2] [3] [4] [5] [6] [7] [41] [42] [Επόμενο] [Τέλος]

Εικόνα 7 : Παρουσίαση όλων των ενεργών φοιτητών.

Στη σελίδα όπου παρουσιάζονται όλοι οι φοιτητές, ο χρήστης έχει τη δυνατότητα να ταξινομήσει τους φοιτητές ανάλογα με τον αριθμό μητρώου, το επώνυμο, το όνομα και το username του. Στο αριστερό μέρος κάτω από το μενού πλοήγησης, ο χρήστης έχει τη δυνατότητα να δει είτε τους ενεργούς φοιτητές, είτε τους απόφοιτους και μπορεί να αναζητήσει κάποιον φοιτητή με βάση αριθμό μητρώου, όνομα, επώνυμο και username. Μπορεί να κάνει εισαγωγή κάποιου φοιτητή, είτε να κάνει μαζική εισαγωγή φοιτητών από εξωτερικό αρχείο. Μπορεί επίσης μέσω της επιλογής Edit να επεξεργαστεί κάποιον φοιτητή. Αν κατά την επεξεργασία ή κατά την νέα εισαγωγή δεν συμπληρωθούν όλα τα απαραίτητα πεδία εμφανίζεται κατάλληλο μήνυμα.



Εικόνα 8: Επεξεργασία φοιτητή

Το εργαλείο “Μαζική εισαγωγή” χρησιμοποιήθηκε για την ενημέρωση όλων των πινάκων της διαδικτυακής βάσης μέσω του αρχείου sql που προαναφέρθηκε. Αν δεν επιλεγεί κάποιο αρχείο ή το αρχείο δεν έχει την κατάληξη txt, τότε εμφανίζεται κατάλληλο μήνυμα. Επίσης σε όλες τις σελίδες που παρουσιάζονται δεδομένα υπάρχει στο κάτω μέρος σελιδοποίηση για καλύτερη διευκόλυνση του χρήστη.

Σχεδίαση διεπαφής καθηγητών

Όπως και πριν στην παρουσίαση των φοιτητών, έτσι και στους καθηγητές ο χρήστης μπορεί να ταξινομεί τα αποτελέσματα που βλέπει ανάλογα με το επώνυμο, το όνομα και το username των καθηγητών. Έχει τη δυνατότητα επεξεργασίας κάποιου καθηγητή, μπορεί να πραγματοποιήσει εισαγωγή ενός καθηγητή ή εισαγωγή από αρχείο.

Να σημειωθεί πως σε κάποιες πτυχιακές εργασίες εισηγητές ήταν κάποιοι καθηγητές που δε διδάσκουν πλέον στο τμήμα, είτε επειδή ήταν έκτακτοι, είτε λόγω συνταξιοδότησης. Επομένως έπρεπε να προστεθου δύο επιλογές (κάτω από το μενού πλοήγησης), όπου ο χρήστης θα μπορεί να βλέπει είτε μόνιμους καθηγητές, είτε έκτακτους και στο πεδίο ενεργός θα φαίνεται ποιοι καθηγητές διδάσκουν και ποιοι όχι.

Μόνιμοι Καθηγητές

Σύνολο: 26

▲ Επίσημο ?	▲ Όνομα Καθηγητή ▼	▲ Username ▼	Ενεργός	
athata	athata	athata	-	Επίτ.
dia	dia	aaaaaaaaaaaa	✓	Επίτ.
fid	fid	fid	✓	Επίτ.
Αδαμίδης	Παναγιώτης	adaidid	✓	Επίτ.
Αμεινιλάκης	Δημήτριος	dimia	✓	Επίτ.
Αρτυνίου	Ευστάθιος	artotiu	✓	Επίτ.
Βασιλάκης	Απόστολος	vas	-	Επίτ.
Βίτσας	Βασίλειος	vitvas	✓	Επίτ.
Γιακουμίδης	Κωνσταντίνος	giakia	✓	Επίτ.
Γουλιάνης	Κωνσταντίνος	gouliana	✓	Επίτ.
Δελαγιάννης	Ιωάννης	idelianis	✓	Επίτ.
Δούβος	Δημήτριος	duv	✓	Επίτ.
Διαμαντίδης	Κωνσταντίνος	diamanit	✓	Επίτ.
Ηλιοβίδης	Χρήστος	iliovidis	✓	Επίτ.
Καραγεωργίου	Αθανάσιος	karageorgi	✓	Επίτ.

Σελίδα 1 από 2

Εικόνα 8: Διεπαφή καθηγητών

Σχεδίαση διεπαφής πτυχιακών

Το παραπάνω πρότυπο σχεδίασης εφαρμόστηκε και στη σχεδίαση της διεπαφής, όπου ο χρήστης μπορεί να βλέπει όλες τις πτυχιακές του κάθε καθηγητή, είτε αυτές προορίζονται για τον διαγωνισμό ανάθεσης, είτε δίνονται σε απ' ευθείας ανάθεση σε κάποιον/κάποιους φοιτητές. Υπάρχει η δυνατότητα ταξινόμησης των αποτελεσμάτων ανάλογα με τον τίτλο, το όνομα του εισηγητή καθώς και την ημερομηνία υποβολής της συγκεκριμένης πτυχιακής.

Ο χρήστης μπορεί επίσης να δει ποιες πτυχιακές είναι ομαδικές και ποιες ατομικές, ποιες έχουν ζητηθεί σε κάποιον διαγωνισμό ανάθεσης αλλά και ποιες έχουν ήδη δοθεί σε κάποιον φοιτητή. Τέλος έχει τη δυνατότητα να βλέπει πτυχιακές τις οποίες τις έχει δημοσιεύσει ένας συγκεκριμένος καθηγητής, μέσω της δεύτερης επιλογής από τα δεξιά, να κάνει αναζήτηση με βάση τον τίτλο ή το όνομα/επώνυμο/username εισηγητή (κάτω από την επιλογή Μαζική Εισαγωγή Πτυχιακών), και να επεξεργαστεί κάποια πτυχιακή εάν το επιθυμεί. Επιπλέον μπορεί να κάνει εισαγωγή νέας πτυχιακής στο σύστημα.

Πτυχιακή Εργασία του φοιτητή Ματούλα Θωμά

Τίτλος	Εισηγητής	Ημ/Τυποθ	ΕΣ/Ανά	Απομ	Απευθ	Ζητ	Δοθ		
Αυτομακά Διαδικαστικά Robot (SelfBots)	Σταμάτης	4/12/2013	χρη1314	✓	✓	-	✓	Πτυχιακές δοθεί	Εδώ
Ανάπτυξη λογισμικού για την Ομοσπονδία Σαλλώνων Εκπαιδευτικού Προσωπικού (ΟΣΕΠ) των ΤΕΙ	Σταμάτης	4/12/2013	χρη1314	✓	✓	-	✓	Πτυχιακές δοθεί	Εδώ
Πρόλογος και Java με GUI: Αντικειμενοστρεφής Προγραμματισμός με δέσμες ενοφείας	Σταμάτης	4/12/2013	χρη1314	✓	✓	-	✓	Πτυχιακές δοθεί	Εδώ
3D Πλοήγηση εντός κτηρίων: Τεχνικές και μελέτη περίπτωσης	Σταμάτης	4/12/2013	χρη1314	✓	✓	-	✓	Πτυχιακές δοθεί	Εδώ
Παραμετροποίηση - Επισκευαστικής Εφαρμογής Αυτόματης Επεξεργασίας Ερευνηματολογίων Πρακτικής Άσκησης Τελεφοίτων και Αποσπώμενης Αποφοίτων	Γουλιάνος	3/12/2013	χρη1314	✓	✓	-	✓	Πτυχιακές δοθεί	Εδώ
Μελέτη της Ποιότητας Εμπειρίας (Quality of Experience) στα Δίκτυα Long Term Evolution (LTE)	Χιτζογιάννης	15/11/2013	χρη1314	✓	✓	-	✓	Πτυχιακές δοθεί	Εδώ
Μελέτη απόδοσης κώδικων δικτύων Long Term Evolution (LTE) με το λογισμικό προσομοίωσης LTE-sim	Χιτζογιάννης	15/11/2013	χρη1314	✓	✓	-	✓	Πτυχιακές δοθεί	Εδώ
Ανάλυση, σχεδίαση και ανάπτυξη συστήματος διαχείρισης περιφερειακών διαδικτυακών υπηρεσιών	Χιτζογιάννης	15/11/2013	χρη1314	✓	✓	-	✓	Πτυχιακές δοθεί	Εδώ
Ανάλυση, σχεδίαση και ανάπτυξη συστήματος διαχείρισης καθηκόντων και διανομημένων	Χιτζογιάννης	15/11/2013	χρη1314	✓	✓	-	✓	Πτυχιακές δοθεί	Εδώ
Study of the access network selection mechanisms for mobile devices in heterogeneous wireless environments	Χιτζογιάννης	15/11/2013	χρη1314	✓	✓	-	✓	Πτυχιακές δοθεί	Εδώ

Εικόνα 9: Διεπαφή πτυχιακών

Εάν κάποια πτυχιακή δεν έχει δοθεί σε κάποιον φοιτητή τότε εμφανίζεται η επιλογή ανάθεση, μέσω της οποίας ο χρήστης θα μπορεί να ορίζει χειροκίνητα την ανάθεση της πτυχιακής σε κάποιον φοιτητή.

Νέα ανάθεση Πτυχιακής

Τίτλος Πτυχιακής:

Εισηγητής:

Φοιτητής 1:

Ημερομηνία έναρξης:

Εικόνα 10: Νέα ανάθεση πτυχιακής

Σχεδίαση διεπαφής αναθέσεων

Η διεπαφή όπου παρουσιάζονται οι αναθέσεις πτυχιακών είναι και η τελευταία που σχεδιάστηκε στο σύστημα. Εδώ ο χρήστης μπορεί να δει όλες τις αναθέσεις πτυχιακών που υπάρχουν (συγκεκριμένα αυτές που υπήρχαν στο access αρχείο) είτε πρόκειται για ολοκληρωμένες, είτε για ακυρωμένες, είτε για τρέχουσες αναθέσεις. Μπορεί επίσης να δει και επιπλέον πληροφορίες που σχετίζονται με την κάθε ανάθεση, όπως ημερομηνία εξέτασης και πρακτικό συνέλευσης ολοκλήρωσης εάν επρόκειτο για

ολοκληρωμένη ανάθεση, ή πρακτικό συνέλευσης ακύρωσης εάν επρόκειτο για ακυρωμένη ανάθεση κ.ά. Στο αριστερό μέρος της διεπαφής ο χρήστης μπορεί να επιλέξει ποιες αναθέσεις θέλει να δει

Αρχική Αναθέσεις Πτυχιακών Ενεργό φοιτητές Καθηγητές Logout

Μαζική Εισαγωγή Αναθέσεων

Ολοκληρωμένες Πτυχιακές

Ακυρωμένες Πτυχιακές

Τρέχουσες Αναθέσεις

Αναζήτηση Αναθέσεων με βάση Εισηγητή

Ακυρωμένες Αναθέσεις Πτυχιακών

Σύνολο: 267

Τίτλος	Εισηγητής	ΑΜ Φοιτητή	ΑΜ 2ου Φοιτητή	Η. Ημ/Εν	Πρ/Ακυρ	Παράταση	Νέος Εισ	Ακυρωμένες Αναθέσεις	Είδος
Ανάπτυξη συστήματος αρχείωσης και επεξεργασίας ερωτηματολογίων στο διαδίκτυο	Αδαμίδης	063046 ΚΑΡΓΚΑΣ	-	26/4/2013	-	-	-	Ακυρωμένες Αναθέσεις	αδελφία
Διασύνδεση Web εφαρμογής για τη μεταγραφή μιας Ανταγωνιστικούπαικτικού Βάσης Δεδομένων σε ...	Καραβίτσουλης	063048 ΔΙΑΤΗΣ	-	26/4/2013	214/27-11-13	-	-	Ακυρωμένες Αναθέσεις	επίδομα
Ευέλικτη ανάπτυξη Java εφαρμογών με τις Spring, Hibernate και Eclipse	Σάββας	063008 ΠΟΥΡΝΟΥ	-	26/4/2013	214/27-11-13	-	-	Ακυρωμένες Αναθέσεις	επίδομα
Ανάπτυξη Dedicated iPhone app με χρήση web services	Σαλαμπίδης	063147 ΠΑΓΚΟΥ	-	26/4/2013	32/6-2-14	-	-	Ακυρωμένες Αναθέσεις	επίδομα
Ανάπτυξη Ολοκληρωμένου Πληροφορικού Συστήματος Διαχείρισης Ηλεκτρονικού Συνεδρίου	Συλκα	063054 ΚΑΣΣΑΒΕΤΗ	-	26/4/2013	7/22-5-13	-	-	Ακυρωμένες Αναθέσεις	επίδομα
Παρουσίαση και μελέτη των εργασιών προεπισθεών για τα πρωτόκολλα IEEE 802.11a	Χανδρινάκης	052892 ΤΟΠΑΛΙΔΗΣ	-	26/4/2013	11/30-8-13	-	-	Ακυρωμένες Αναθέσεις	επίδομα
Αποστολικά Συστήμα Ισός	Γεωργαντίδης	052945 ΚΟΥΡΟΥ	063134 ΧΑΥΗ	15/2/2013	211/30-9-13	-	-	Ακυρωμένες Αναθέσεις	επίδομα
Διασύνδεση Web εφαρμογής για τη μεταγραφή μιας Ανταγωνιστικούπαικτικού Βάσης Δεδομένων σε ...	Καραβίτσουλης	063008 ΒΕΡΡΑΦΟΣ	-	30/11/2012	6/24-4-13	-	-	Ακυρωμένες Αναθέσεις	επίδομα
Ανάπτυξη Java εφαρμογών με την χρήση του εργαλείου Apache Maven	Σάββας	073250 ΑΝΔΡΕΑΔΗΣ	-	30/11/2012	1/21-1-13	-	-	Ακυρωμένες Αναθέσεις	επίδομα
Σύστημα Ηλεκτρονικής Ψηφοφορίας	Αδαμίδης	032226 ΜΕΤΣΙΔΗΣ	-	30/11/2012	214/27-11-13	-	-	Ακυρωμένες Αναθέσεις	επίδομα

Σελίδα 1 από 27

Εικόνα 11: Διεπαφή Αναθέσεων

Όσον αφορά τις τρέχουσες αναθέσεις ο χρήστης έχει δύο επιπλέον επιλογές:

1. Να ορίσει μία ανάθεση ως ολοκληρωμένη δίνοντας τις απαραίτητες πληροφορίες (ημερομηνία εξέτασης, πρακτικό συνέλευσης ολοκλήρωσης, εξάμηνο ολοκλήρωσης)
2. Να ορίσει μία ανάθεση ως ακυρωμένη δίνοντας το πρακτικό συνέλευσης ακύρωσης

Αρχική Αναθέσεις Πτυχιακών Ενεργό φοιτητές Καθηγητές Logout

Μαζική Εισαγωγή Αναθέσεων

Ολοκληρωμένες Πτυχιακές

Ακυρωμένες Πτυχιακές

Τρέχουσες Αναθέσεις

Αναζήτηση Αναθέσεων με βάση Εισηγητή

Τρέχουσες Αναθέσεις Πτυχιακών

Σύνολο: 149

Τίτλος	Εισηγητής	ΑΜ Φοιτητή	ΑΜ 2ου Φοιτητή	Η. Ημ/Εν	Νέος Εισ	Εξόφλη	Παράταση	Ολοκλήρωση	Ακύρωση	Τρέχουσες Αναθέσεις	Είδος
Ανάπτυξη εφαρμογών διαδικτυακής αλληλεπίδρασης με χρήση του CMS	Αδαμίδης	063046 ΚΑΡΓΚΑΣ	-	26/4/2013	-	-	-	✓	✗	Τρέχουσες Αναθέσεις	επίδομα
Ανάπτυξη web- εφαρμογών και πλατφόρμα Ruby on Rails	Αδαμίδης	063046 ΚΑΡΓΚΑΣ	-	26/4/2013	-	-	-	✓	✗	Τρέχουσες Αναθέσεις	επίδομα
Ανάπτυξη web- εφαρμογών και πλατφόρμα Java Enterprise Edition	Αδαμίδης	063046 ΚΑΡΓΚΑΣ	-	26/4/2013	-	-	-	✓	✗	Τρέχουσες Αναθέσεις	επίδομα
Ανάπτυξη mobile- εφαρμογών και πλατφόρμα Android	Αδαμίδης	063046 ΚΑΡΓΚΑΣ	-	26/4/2013	-	-	-	✓	✗	Τρέχουσες Αναθέσεις	επίδομα
Εξέλιξη και μελλοντικές εφαρμογές Διαφορετικών Διατάξεων	Αδαμίδης	063046 ΚΑΡΓΚΑΣ	-	26/4/2013	-	-	-	✓	✗	Τρέχουσες Αναθέσεις	επίδομα
Μεταφόρτωση Πτυχιακών Εργασιών	Αδαμίδης	063046 ΚΑΡΓΚΑΣ	-	26/4/2013	-	-	-	✓	✗	Τρέχουσες Αναθέσεις	επίδομα
Παρουσίαση και Στατιστικά των φοιτητών	Αδαμίδης	063046 ΚΑΡΓΚΑΣ	-	26/4/2013	-	-	-	✓	✗	Τρέχουσες Αναθέσεις	επίδομα
Σύστημα Αναθέσεων Πτυχιακών	Αδαμίδης	063046 ΚΑΡΓΚΑΣ	-	26/4/2013	-	-	-	✓	✗	Τρέχουσες Αναθέσεις	επίδομα
Σύστημα Ψηφιακής Διδακτικής Αντίστροφης Αναλογίας Πτυχιακών	Αδαμίδης	063046 ΚΑΡΓΚΑΣ	-	26/4/2013	-	-	-	✓	✗	Τρέχουσες Αναθέσεις	επίδομα
Σύστημα Παρακολούθησης των Εργασιών με χρήση CMS	Αδαμίδης	063046 ΚΑΡΓΚΑΣ	-	26/4/2013	-	-	-	✓	✗	Τρέχουσες Αναθέσεις	επίδομα

Ολοκλήρωση Πτυχιακής

Μεταφόρτωση Πτυχιακών Εργασιών

Τίτλος Πτυχιακής:

Εισηγητής:

Φοιτητής 1:

Ημερομηνία έναρξης:

Νέος Εισηγητής:

Παράταση:

Ημερομηνία Εξέτασης:

Εξάμηνο Ολοκλήρωσης:

Παραστατικό:

Σελίδα 1 από 15

Εικόνα 12: Ολοκλήρωση Πτυχιακής

Να σημειωθεί πως όπως και πριν στην διεπαφή των πτυχιακών, ο χρήστης έχει τη δυνατότητα να επιλέξει να δει τις αναθέσεις ενός συγκεκριμένου καθηγητή, είτε μέσω της επιλογής στο δεξιό μέρος, είτε μέσω αναζήτησης. Ακόμα έχει τη δυνατότητα ταξινόμησης των αποτελεσμάτων ανάλογα με τον τίτλο της πτυχιακής, το όνομα του εισηγητή, τον αριθμό μητρώου του φοιτητή και την ημερομηνία έναρξης. Τέλος μπορεί να επεξεργαστεί ανά πάσα στιγμή οποιαδήποτε εγγραφή και να αλλάξει όποια πληροφορία θέλει μέσω της τελευταίας επιλογής μέσω της οποίας μπορεί επίσης να δει αν η συγκεκριμένη ανάθεση ήταν απευθείας ή όχι.

5.5 Υλοποίηση Συστήματος

Για την υλοποίηση του συστήματος χρησιμοποιήθηκε αρχικά HTML και CSS για τη σωστή και ωραία εμφάνιση των αποτελεσμάτων. Προκειμένου να προστεθεί κάποια αλληλεπίδραση με το χρήστη χρησιμοποιήθηκε Javascript και JQuery, όπως για παράδειγμα στον έλεγχο ότι έχουν συμπληρωθεί όλα τα απαραίτητα πεδία στην εισαγωγή φοιτητή, καθηγητή, πτυχιακής. Επίσης σε συγκεκριμένα σημεία του συστήματος χρησιμοποιήθηκε τεχνολογία Ajax σε συνδυασμό με JQuery (λειτουργία αναζήτησης). Επιπλέον η λειτουργικότητα εμφάνισης του παραθύρου στη μεμονωμένη επεξεργασία μιας εγγραφής ή στην μεμονωμένη εισαγωγή, είναι ένα έτοιμο εργαλείο της JQuery, είχε χρησιμοποιηθεί από το φοιτητή και διατηρήθηκε και στην υλοποίηση του συστήματος.

Οι παραπάνω τεχνολογίες αφορούν κυρίως το αποτέλεσμα που βλέπει ο χρήστης (**Front End**). Ωστόσο έχει χρησιμοποιηθεί σε μεγάλο βαθμό η PHP. Η επιλογή των κατάλληλων δεδομένων για εμφάνιση, η εισαγωγή νέων εγγραφών, μεμονωμένων και μαζικών (από εξωτερικό αρχείο), αλλά και οι λειτουργίες ταξινόμησης, έγιναν μέσω PHP κώδικα (**Back End**). Από τη στιγμή που ήταν έτοιμο το σύστημα έπρεπε να μεταφορτωθεί στον προσωπικό λογαριασμό του καθηγητή.

5.5.1 Ολοκλήρωση Συστήματος

Οι τελευταίες ενέργειες που έγιναν για την ολοκλήρωση του συστήματος, αφορούσαν την εισαγωγή φοιτητών πτυχιακών και αναθέσεων από εξωτερικό αρχείο.

Εισαγωγή φοιτητών

Ο καθηγητής ήθελε στο σύστημα να υπάρχουν όλοι οι ενεργοί φοιτητές μαζί με κάποιες επιπλέον πληροφορίες όπως διδακτικές μονάδες, συντελεστής προόδου, μέσος όρος, εξάμηνο, username. Όλες αυτές οι πληροφορίες βρίσκονται σε μια διαφορετική τοποθεσία από αυτήν του συστήματος και μάλιστα οι πληροφορίες αυτές ενημερώνονται κάθε εξάμηνο.

Έτσι αποφασίστηκε πως κάθε εξάμηνο που αλλάζουν αυτές οι πληροφορίες, θα γίνεται μία ενημέρωση των φοιτητών στο σύστημα με ένα εξωτερικό αρχείο το οποίο θα το παίρνει ο καθηγητής από την κατάλληλη τοποθεσία. Αυτό που χρειαζόταν να γίνει ήταν οι απαραίτητες διορθώσεις στον κώδικα εισαγωγής φοιτητών από εξωτερικό αρχείο, ώστε να είναι δυνατή αυτή η λειτουργία. Μόλις έγινε γνωστή η δομή του αρχείου αυτού, πραγματοποιήθηκαν οι αλλαγές.

Εισαγωγή αναθέσεων – πτυχιακών

Όσον αφορά την ενημέρωση αναθέσεων και πτυχιακών, ο καθηγητής ήθελε να πραγματοποιεί αυτήν τη ενέργεια μέσω δύο αρχείων τα οποία προέρχονται από μια άλλη πτυχιακή κάθε φορά που γίνεται ένας διαγωνισμός αναθέσεων πτυχιακών. Συνεπώς η γνώση της δομής αυτών των αρχείων ήταν απαραίτητη ώστε να τροποποιηθεί κατάλληλα ο σχετικός κώδικας, προκειμένου πλέον η ενημέρωση να γίνεται με βάση την δομή αυτών των αρχείων και όχι με βάση των αρχείων που προήλθαν από το access

Τα αρχικά αρχεία χρησιμοποιήθηκαν μόνο για την μεταφορά των δεδομένων από το access στην διαδικτυακή βάση. Μόλις έγινε γνωστή η δομή των νέων αρχείων έγιναν οι σχετικές αλλαγές. Από δω και στο εξής ο καθηγητής θα μπορεί να ενημερώνει την βάση του συστήματος μέσω των αρχείων αυτών, χωρίς την εξάρτησή του από το τοπικό αρχείο access που διατηρούσε.

Τέλος ο καθηγητής ζήτησε να προστεθεί μία ακόμη λειτουργικότητα που υπήρχε στο αρχείο access. Αυτή ήταν η εξής:

Κάθε φορά που γίνεται μία νέα ανάθεση πτυχιακής (ατομική ή ομαδική), πρέπει να υπάρχουν σαν πληροφορίες ο μέσος όρος, το εξάμηνο, οι διδακτικές μονάδες και ο συντελεστής προόδου του φοιτητή ή των φοιτητών της συγκεκριμένης ανάθεσης, έτσι ώστε όταν ανανεωθούν αυτές οι πληροφορίες στον πίνακα με τους φοιτητές να έχουν διατηρηθεί οι τιμές τους τη στιγμή που έγινε η ανάθεση.

Για να είναι αυτό δυνατό έπρεπε να προστεθούν οι τέσσερις παραπάνω στήλες από δύο φορές και στον πίνακα με τις αναθέσεις, σύνολο οκτώ δηλαδή. Κάθε φορά που θα γίνεται νέα ανάθεση, είτε χειροκίνητα είτε μέσω ενημέρωσης από εξωτερικό αρχείο, θα ενημερώνονται αυτές οι στήλες με τις κατάλληλες τιμές. Με τον τρόπο αυτό θα δημιουργείται το λεγόμενο “στιγμιότυπο ανάθεσης”.

5.6 Μεταφόρτωση πτυχιακής

Από τη στιγμή που η επέκταση και βελτίωση της πτυχιακής του φοιτητή είχε ολοκληρωθεί με επιτυχία, σειρά είχε η μεταφόρτωση του τελικού συστήματος από τον προσωπικό λογαριασμό του γράφοντα στο λογαριασμό του επιβλέποντα καθηγητή. Τα βήματα που ακολουθήθηκαν ήταν τα ίδια όπως και στη μεταφόρτωση της πρώτης πτυχιακής.

Δημιουργήθηκε από τον καθηγητή ένας φάκελος με τα κατάλληλα δικαιώματα εγγραφής (777). Στη συνέχεια μέσω της παρακάτω εντολής έγινε η αντιγραφή όλων των αρχείων, διατηρώντας φυσικά όλα τα δικαιώματα ίδια, στον φάκελο που δημιουργήθηκε.

```
cp -r -p public_html/konpala/* /home/staff/tech/dima/public_html/ptyxiakes/konpala
```

Στη συνέχεια χρειαζόταν να γίνει αντιγραφή της βάσης που βρισκόταν στον προσωπικό του λογαριασμό. Όπως και στην πρώτη πτυχιακή, έτσι και σε αυτήν την περίπτωση έγινε εξαγωγή της βάσης σε sql αρχείο, το οποίο χρησιμοποιήθηκε για την δημιουργία της βάσης-αντίγραφο στον λογαριασμό του καθηγητή.

```
mysqldump -h hostname -u user -pPassword database > konpala.sql
```

```
mysql -h hostname -u user -pPassword new_database < konpala.sql
```

Έχοντας πλέον τα νέα αντεγγραμμένα αρχεία μαζί με τη νέα αντεγγραμμένη βάση, το μόνο που έμεινε ήταν η τελική σύνδεσή τους. Σε αυτήν την περίπτωση έπρεπε να δοθούν τα νέα στοιχεία για τη σύνδεση με τη βάση στο κατάλληλο αρχείο. Το αρχείο αυτό είχε δημιουργηθεί από το φοιτητή, βρισκόταν στον φάκελο db και είχε όνομα *db_connection.php*.

Κάνοντας και τις απαραίτητες αλλαγές όσον αφορά τους ιδιοκτήτες των νέων αρχείων, έχει ολοκληρωθεί με επιτυχία και η μεταφόρτωση της βελτιωμένης πτυχιακής του φοιτητή και είναι πλέον προσβάσιμη από τον λογαριασμό του καθηγητή.

5.7 Προβλήματα που αντιμετωπίστηκαν

Όπως και στην μεταφόρτωση της πρώτης πτυχιακής, έτσι και σε αυτήν παρουσιάστηκαν κάποια προβλήματα. Συγκεκριμένα παρουσιάστηκαν τα παρακάτω προβλήματα:

1. Συντακτικά λάθη στο αρχείο
2. Ενημέρωση Καθηγητών
3. Διπλοεγγραφές στις ομαδικές πτυχιακές-αναθέσεις

Συντακτικά Λάθη στο αρχείο

Στο αρχείο του καθηγητή, υπήρχαν, όπως έχει προαναφερθεί, πληροφορίες για παλαιότερες αναθέσεις πτυχιακών (ολοκληρωμένες, ακυρωμένες, αλλά και πτυχιακές που δεν είχαν ανατεθεί σε κάποιον φοιτητή). Οι εγγραφές σε αυτό το αρχείο ήταν πάρα πολλές (πάνω από χίλιες) και όπως είναι φυσικό σε κάποιες υπήρχαν κάποια λάθη, κυρίως συντακτικά.

Όπως έχει αναφερθεί, προκειμένου να γίνει ενημέρωση της διαδικτυακής βάσης χρειαζόταν να γίνει εξαγωγή ενός αρχείου με συγκεκριμένη δομή. Οι στήλες στο αρχείο έπρεπε να είναι χωρισμένες με το ελληνικό ερωτηματικό, συνεπώς αν υπήρχε σε κάποιο πεδίο ο χαρακτήρας αυτός, πχ στα σχόλια, τότε υπήρχε πρόβλημα στην εισαγωγή των δεδομένων.

Κατά την εξαγωγή του αρχείου για την εισαγωγή των αναθέσεων, υπήρχε η εξής εγγραφή στο αρχείο txt

961;"Κλεφτούρης Δ."; "ΑΝΑΠΤΥΞΗ ΗΛΕΚΤΡΟΝΙΚΗΣ ΠΛΑΤΦΟΡΜΑΣ ΓΙΑ ΒΙΒΛΙΟΠΩΛΕΙΟ ΜΕΣΩ ΣΥΣΤΗΜΑΤΟΣ ΔΙΑΧΕΙΡΙΣΗΣ ΠΕΡΙΕΧΟΜΕΝΟΥ (CMS)"; 23/10/2013;"χειμ1314"; 0;0;0;1;"ej;etash 3/12/13"

Σύμφωνα με το αντίστοιχο sql ερώτημα, ο αριθμός των στηλών που έπρεπε να εμφανιστούν ήταν δέκα, με χαρακτήρα διαχωρισμού το ελληνικό ερωτηματικό. Αν παρατηρήσει κάποιος πιο προσεκτικά την παραπάνω εγγραφή θα διαπιστώσει πως οι στήλες δεν είναι δέκα αλλά έντεκα, και αυτό διότι στη λέξη 'ej;etasi' υπήρχε ο συγκεκριμένος χαρακτήρας διαχωρισμού. Αυτό είχε σαν αποτέλεσμα να μη δουλεύει σωστά ο κώδικας rhp που είχε γραφτεί καθώς απαιτούσε σαν είσοδο δέκα στήλες και όχι έντεκα. Η λύση του προβλήματος ήταν η εξής:

- Πριν προχωρήσει η διαδικασία ενημέρωσης της βάσης από το αρχείο, γίνεται έλεγχος για τον αριθμό των στηλών της κάθε γραμμής, με διαχωριστή να συνεχίζει να παραμένει το ελληνικό ερωτηματικό. Αν ο αριθμός των στηλών σε κάθε γραμμή είναι δέκα τότε συνεχίζεται η ενημέρωση της βάσης, ενώ σε διαφορετική περίπτωση ο κώδικας τερματίζεται εμφανίζοντας στο χρήστη σε ποια γραμμή του αρχείου εντοπίστηκε το λάθος ώστε να μπορεί να το λύσει εύκολα και γρήγορα.

Ενημέρωση Καθηγητών

Ένα άλλο πρόβλημα που παρουσιάστηκε αφορούσε την ενημέρωση των καθηγητών. Οι πληροφορίες όσον αφορά τους καθηγητές που υπήρχαν στο αρχείο, δεν επαρκούσαν για την πλήρη ενημέρωση του αντίστοιχου πίνακα στη διαδικτυακή βάση.

Στο αρχείο υπήρχε μόνο επώνυμο του καθηγητή και το πρώτο γράμμα του μικρού ονόματος. Αντίθετα ο πίνακας στη διαδικτυακή βάση είχε πέρα από τη στήλη Επώνυμο και άλλες, με αποτέλεσμα μόνο η ενημέρωση αυτής της στήλης να είναι εφικτή. Όλες οι άλλες πληροφορίες έπρεπε να δοθούν χειροκίνητα. Το γεγονός όμως ότι το επώνυμο δεν ήταν απομονωμένο σαν εγγραφή αλλά μαζί με το πρώτο γράμμα του μικρού ονόματος του καθηγητή, πρόσθεσε μία ακόμα, μικρή βέβαια, δυσκολία στην ενημέρωση των καθηγητών.

Παρατηρώντας προσεκτικά τη δομή που είχαν οι εγγραφές στη στήλη επώνυμο του αρχείου, διαπιστώθηκε πως το επώνυμο του κάθε καθηγητή και το πρώτο γράμμα του μικρού ονόματος, ήταν χωρισμένα με κενό.

Αμανατιάδης Δ.

Εφόσον υπήρχε αυτή η λογική σε όλες τις εγγραφές, τότε μπορούσε εύκολα να γίνει ο διαχωρισμός αν ληφθεί σαν χαρακτήρας διαχωρισμού ο κενό. Με τον τρόπο αυτό ενημερώθηκαν τα επώνυμα των καθηγητών στη νέα βάση και οι υπόλοιπες πληροφορίες ενημερώθηκαν χειροκίνητα.

Η παραπάνω ενέργεια πραγματοποιήθηκε όχι μόνο στην ενημέρωση των καθηγητών, αλλά και στην ενημέρωση των πτυχιακών και αναθέσεων, αφού και στις δύο αυτές περιπτώσεις υπάρχει στήλη εισηγητής. Συγκεκριμένα διαβαζόταν η κάθε γραμμή από τα δύο αρχεία, γινόταν απομόνωση του επωνύμου με τον παραπάνω τρόπο, και στη συνέχεια γινόταν αναζήτηση του επωνύμου στον πίνακα των εισηγητών ώστε να βρεθεί ο αντίστοιχο id, το οποίο χρησιμοποιούνταν σαν ξένο κλειδί στον πίνακα των πτυχιακών και των αναθέσεων. Ο κώδικας PHP που χρησιμοποιήθηκε για τις παραπάνω ενέργειες βρίσκεται στο παράρτημα.

Διπλοεγγραφές στις ομαδικές πτυχιακές-αναθέσεις

Όταν μία πτυχιακή ήταν ομαδική και είχε ανατεθεί σε δύο φοιτητές, τότε υπήρχαν δύο εγγραφές στο αρχείο access, με τα παρακάτω πεδία να είναι ίδια:

- Τίτλος πτυχιακής
- Ημερομηνία υποβολής πτυχιακής
- Εξάμηνο ανάθεσης

Στην μία εγγραφή υπήρχαν τα στοιχεία του πρώτου φοιτητή και στην αμέσως επόμενη υπήρχαν τα στοιχεία του δεύτερου φοιτητή. Επομένως έπρεπε να βρεθεί ένας τρόπος ώστε στην τελική βάση να υπάρχουν μόνο μία φορά οι ομαδικές αναθέσεις, με σωστά ενημερωμένες τις στήλες των αντίστοιχων φοιτητών. Λαμβάνοντας υπόψιν τις τρεις στήλες που παρέμεναν ίδιες, δημιουργήθηκε ένας αλγόριθμος ο οποίος έκανε αυτήν την παραπάνω

ενημέρωση. Ακολουθεί η περιγραφή του αλγορίθμου, ενώ ο κώδικας βρίσκεται στο παράρτημα.

1. Πτυχιακές

Για κάθε γραμμή του εξωτερικού αρχείου, γίνεται έλεγχος αν υπάρχει ήδη εγγραφή στον πίνακα των πτυχιακών με ίδιο τίτλο, ίδια ημερομηνία υποβολής και ίδιο εξάμηνο ανάθεσης. Αν υπάρχει τότε η συγκεκριμένη γραμμή αγνοείται και προχωράμε στην επόμενη γραμμή του αρχείου. Αν δεν υπάρχει τότε γίνεται νέα εγγραφή στη βάση με τα στοιχεία της γραμμής.

2. Αναθέσεις

Για κάθε γραμμή του εξωτερικού αρχείου γίνεται έλεγχος αν υπάρχει ήδη εγγραφή στον πίνακα των αναθέσεων με ίδιο τίτλο, ίδια ημερομηνία υποβολής και ίδιο εξάμηνο ανάθεσης (για λόγους ευκολίας προστέθηκαν στον πίνακα αναθέσεις οι συγκεκριμένες στήλες οι οποίες μετά αφαιρέθηκαν). Αν υπάρχει ήδη μια εγγραφή τότε γίνεται ενημέρωση της εγγραφής προσθέτοντας μόνο τον αριθμό μητρώου του δεύτερου φοιτητή. Σε διαφορετική περίπτωση γίνεται νέα εγγραφή με τις πληροφορίες της συγκεκριμένης γραμμής.

Ο αντίστοιχος κώδικας βρίσκεται στο παράρτημα.

ΜΕΡΟΣ Γ

Παράρτημα

6.1 Δημιουργία προσωπικής ιστοσελίδας στον Server (aeto)

Όλοι φοιτητές του τμήματος πληροφορικής έχουν έναν λογαριασμό στον server της σχολής, τον aeto. Τον χώρο αυτόν μπορούν να τον χρησιμοποιήσουν είτε για δημιουργία προσωπικής ιστοσελίδας ή ενός ολοκληρωμένου διαδικτυακού πληροφοριακού συστήματος (για λόγους πτυχιακής), είτε απλά για αποθηκευτικό χώρο.

Για να δημιουργήσει κάποιος μία απλή HTML σελίδα στον aeto προφανώς πρέπει αρχικά να ξέρει τα στοιχεία σύνδεσής του στον server, username και password, ώστε να μπορεί να συνδεθεί και να τοποθετήσει εκεί τα αρχεία του. Μόλις δημιουργήσει τα αρχεία της σελίδας του (HTML, CSS, Javascript) πρέπει στη συνέχεια να δημιουργήσει έναν φάκελο στον προσωπικό του λογαριασμό. Ο φάκελος αυτός πρέπει υποχρεωτικά να έχει το όνομα 'public_html' και κωδικό δικαιωμάτων 711. Μέσα σε αυτόν τον φάκελο θα πρέπει ο φοιτητής να τοποθετήσει όλα τα αρχεία της προσωπικής του ιστοσελίδας, με την αρχική σελίδα να έχει όνομα index.html ή index.php αν χρησιμοποιεί php.

Όλη την παραπάνω διαδικασία μπορεί να την πραγματοποιήσει με δύο τρόπους:

1. Με απομακρυσμένη σύνδεση στο server
2. Με τη χρήση κάποιου προγράμματος μεταφοράς αρχείων (ftp) πχ Filezilla ή WinSCP.

Και στις 2 περιπτώσεις πρέπει να χρησιμοποιήσει τα εξής στοιχεία:

- Username: το username που έχει στον aeto
- Password: το password που έχει στον aeto
- Hostname: aetos.it.teithe.gr
- Port: 22

Ο σύνδεσμος όπου μπορεί να δει την ιστοσελίδα του ο φοιτητής είναι:

<http://aetos.it.teithe.gr/~username/>

6.2 Χρήση Mysql

Ένας φοιτητής σαφώς έχει τη δυνατότητα να χρησιμοποιήσει και Mysql ώστε να υποστηρίξει είτε κάποια πτυχιακή του ή κάποια εργασία στα πλαίσια ενός μαθήματος, είτε για άλλους δικούς του λόγους (εκμάθηση). Σε κάθε περίπτωση έχει δύο επιλογές.

1. Χρήση κεντρικής Mysql
2. Χρήση προσωπικής Mysql

Κεντρική Mysql

Η κεντρική Mysql που υπάρχει στον aeto διαχειρίζεται κεντρικά και σε αυτήν μπορούν να έχουν πρόσβαση όλοι όσοι έχουν λογαριασμό στον aeto. Ο κάθε χρήστης μπορεί να έχει μόνο μία βάση που θα έχει το ίδιο όνομα με το όνομα του χρήστη (username). Είναι ο εύκολος-προτεινόμενος τρόπος χρήσης καθώς το μόνο που χρειάζεται κάποιος για να αποκτήσει πρόσβαση σε αυτήν είναι να στείλει ένα email στον υπεύθυνο καθηγητή αναφέροντας απλά το όνομά του καθώς και το σκοπό χρήσης της βάσης.

Για να συνδεθεί κάποιος στην κεντρική Mysql πρέπει να τρέξει την παρακάτω εντολή:

```
mysql -h 127.0.0.1 -P 3306 -u username -pPassword
```

Το password αν δεν είναι κολλητά γραμμένο με το -p τότε θα ζητηθεί η εισαγωγή του με το που ο χρήστης πατήσει το enter.

Προσωπική Mysql

Η χρήση της προσωπικής Mysql είναι λίγο πιο πολύπλοκη, έχει όμως περισσότερες δυνατότητες σε σχέση με την κεντρική.

Η προσωπική Mysql είναι ουσιαστικά η εκτέλεση της Mysql, ώστε να αποθηκεύει κάποιος τους πίνακες και όλα τα άλλα δεδομένα στον προσωπικό του χώρο. Τα πλεονεκτήματα σε σχέση με την κεντρική Mysql είναι:

- Απεριόριστος αριθμός βάσεων
- Δυνατότητα απομακρυσμένης πρόσβασης
- Πρόσβαση ως Διαχειριστής της βάσης (Database Administrator)

Για διευκόλυνση έχει δημιουργηθεί ένα script το οποίο αυτοματοποιεί την εγκατάσταση της Mysql. Παρόλα αυτά κάποιος που θέλει να χρησιμοποιήσει την προσωπική Mysql θα πρέπει να γνωρίζει πως λειτουργεί η Mysql γενικά. Πληροφορίες υπάρχουν στην επίσημη ιστοσελίδα (<http://www.mysql.com/>).

Επομένως για να γίνει η εγκατάσταση της Mysql στον προσωπικό χώρο του χρήστη αρκεί να τρέξει την παρακάτω εντολή:

```
create_mysql
```

Πρέπει να σημειωθεί πως παρόλο που η Mysql είναι έτοιμη για χρήση, εάν γίνει επανεκκίνηση στον αετο ο δαίμονας δεν θα ξεκινάει αυτόματα. Για να γίνει κάτι τέτοιο χρειάζεται να κατασκευαστεί κάποιο κατάλληλο keep alive script το οποίο και να καλείται από το crontab. Παρακάτω γίνεται αναφορά για το συγκεκριμένο script.

Για να συνδεθεί κάποιος στην προσωπική Mysql πρέπει να τρέξει την παρακάτω εντολή:

```
mysql -u root -pPassword
```

Keep alive script

Το συγκεκριμένο script έπρεπε να δημιουργηθεί ώστε να ξεκινάει την Mysql σε περίπτωση που δεν έτρεχε. Το script αυτό είναι:

```
#!/bin/bash
```

```
mysqladmin -u root -pPassword ping 2>/dev/null 1>/dev/null
```

```
if [ $? -ne 0 ]; then
```

```
    mysqld_safe --skip-syslog &
```

```
fi
```

Έπειτα πρέπει να γίνει μία ρύθμιση ώστε το παραπάνω script να καλείται από το crontab κάθε 30 λεπτά. Αυτό γίνεται με τις παρακάτω εντολές.

```
crontab -e
```

```
πατάμε το i για να μπορούμε να επεξεργαστούμε
```

```
MAILTO="matoulas@it.teithe.gr"
```

```
*/30 * * * * sh -x mysql.sh
```

Η εντολή MAILTO είναι προαιρετική. Τη βάζουμε αν θέλουμε να ενημερωνόμαστε με email όποτε καλείται το script. Επομένως για να υπάρχει καλύτερη εικόνα για το πότε ακριβώς ξεκινάει η Mysql αν δεν τρέχει, στο script πρέπει να τροποποιηθεί ως εξής:

```
#!/bin/bash
```

```
mysqladmin -u root -pPassword ping 2>/dev/null 1>/dev/null
```

```
if [ $? -ne 0 ]; then
```

```
mysql_safe --skip-syslog &
```

```
echo "Starting" | mail -s "keep_alive_script" matoulas@it.teithe.gr
```

```
fi
```

Έτσι κάθε φορά που θα ξεκινάει η Mysql θα στέλνε ένα email στην επιθυμητή διεύθυνση.

6.3 Πτυχιακή Χατζή

Εδώ παρουσιάζεται ο κώδικας που χρησιμοποιήθηκε για την αντιγραφή της βάσης στην πτυχιακή του Χατζή.

Εξαγωγή sql αρχείο για το Joomla site

Παρατηρώντας τη βάση του φοιτητή, βγήκε το συμπέρασμα πως οι πίνακες που αφορούσαν το Joomla site είχαν πρόθεμα 'dxuno_'. Άρα η κατάλληλη εντολή για την εξαγωγή του sql αρχείου ήταν:

```
mysql -h 127.0.0.1 -P 3306 -u Username -pPassword Database -e 'show tables like  
"dxuno_%" | grep -v Tables_in | xargs mysqldump -h 127.0.0.1 -P 3306 -u Username  
-pPassword Database > thomas_joomla.sql
```

Εξαγωγή sql αρχείο για το Drupal site

Οι πίνακες που αφορούσαν το Drupal site δεν είχαν κανένα πρόθεμα. Άρα έπρεπε να εξαιρεθούν όλοι οι πίνακες που είχαν πρόθεμα. Η εντολή για αυτήν την ενέργεια ήταν:

```
mysql -h 127.0.0.1 -P 3306 -u Username -pPassword Database -e "show tables where tables_in_antcha not like 'dxuvo_%' and tables_in_antcha not like 'phpbb_%' and tables_in_antcha not like 'tsvmj_%' and tables_in_antcha not like 'r3s7u_%' and tables_in_antcha not like 'c182n_%' and tables_in_antcha not like 'bak_c182n_%;'" | grep -v Tables_in | xargs mysqldump -h 127.0.0.1 -P 3306 -u Username -pPassword Database > thomas_drupal.sql
```

Εξαγωγή sql αρχείο για το PHPBB site

Λόγω του μεγάλου μέγεθος της βάσης δεν ήταν δυνατόν η αντιγραφή των πινάκων που αφορούσαν το PHPBB. Ωστόσο θεωρητικά η εντολή για την εξαγωγή των πινάκων, που αφορούσαν το PhpBB, σε sql αρχείο θα ήταν η εξής:

```
mysql -h 127.0.0.1 -P 3306 -u Username -pPassword Database -e 'show tables like "phpbb_%"' | grep -v Tables_in | xargs mysqldump -h 127.0.0.1 -P 3306 -u Username -pPassword Database > thomas_phpbb.sql
```

6.4 Πτυχιακή Παλάσχα

Σε αυτό το κεφάλαιο υπάρχει ο κώδικας (SQL-PHP) που χρησιμοποιήθηκε κατά την επέκταση-βελτίωση της πτυχιακής του Παλάσχα.

Απομόνωση Φοιτητών από το access αρχείο

```
SELECT Distinct Πτυχιακές.AM_Φοιτ, Πτυχιακές.Επ_Φοιτητή, Ον_Φοιτητή  
FROM Πτυχιακές;
```

Απομόνωση Καθηγητών από το access αρχείο

```
SELECT Distinct Πτυχιακές.Εισηγητής  
FROM Πτυχιακές;
```

Απομόνωση Πτυχιακών από το access αρχείο

```
SELECT Πτυχιακές.[αα], Πτυχιακές.[Τίτλος], Πτυχιακές.[Εισηγητής],  
Πτυχιακές.[Ημ_Υποβ], Πτυχιακές.[AM_Φοιτ], Πτυχιακές.[Εξ_ανάθεσης]  
FROM Πτυχιακές;
```

Απομόνωση Αναθέσεων από το access αρχείο

```
SELECT Πτυχιακές.[αα], Πτυχιακές.[Τίτλος], Πτυχιακές.[Εισηγητής],
Πτυχιακές.[Ημ_Υποβ], Πτυχιακές.[ΑΜ_Φοιτ], Πτυχιακές.[Εξ_ανάθεσης],
Πτυχιακές.[Ημ_έναρ], Πτυχιακές.[Ολο], Πτυχιακές.[ΠΣ ολο], Πτυχιακές.[Ημ_εξέτ],
Πτυχιακές.[Ακύρ], Πτυχιακές.[ΠΣ ακ], Πτυχιακές.[Παρ], Πτυχιακές.[ΠΣ παρ],
Πτυχιακές.[ΠΣ αλ επ], Πτυχιακές.[Νέος Επιβ], Πτυχιακές.[Σχόλια]

FROM Πτυχιακές

WHERE Πτυχιακές.Δοθ=True;
```

Ανέβασμα εξωτερικού αρχείου

```
if(!isset($_SESSION['role'])) //έλεγχος αν υπάρχει το session, δηλαδή αν ο χρήστης είναι συνδεδεμένος
```

```
header('Location: ../'); //αν δεν υπάρχει τότε γίνεται ανακατεύθυνση στην αρχική σελίδα
```

```
if( $_FILES['myFile']['name'] != ""){
    include("../header.php");
    include("../menu.php");
    include("../functions/myFunctions.php");
    move_uploaded_file($_FILES['myFile']['tmp_name'], "../".$_FILES['myFile']['name']);

    $file = fopen($_FILES['myFile']['name'], "r");
    $db->db_open();

    while ($line = fgets($file)) {
        list($field1, $field2, $field4 ..... ) = explode(':', str_replace("'", "$line));
        //Εισαγωγή των πεδίων στον κατάλληλο πίνακα.
    }
    fclose($file);
    $db->db_close();
    unlink($_FILES['myFile']['name']);
}
```

Ενημέρωση Καθηγητών

Αρχείο: "Αμανατιάδης Δ."

```
list($eisigitis) = explode(':', str_replace("'", "$line)); // Η τιμή του $eisigitis είναι 'Αμανατιάδης Δ'
```

```
$seisig = explode(" ", $eisigitis); // Η τιμή του $eisigitis είναι 'Αμανατιάδης'.
```

Η παραπάνω τεχνική εφαρμόστηκε όπου υπήρχε σαν ξένο κλειδί το id του εισηγητή (Αναθέσεις-Πτυχιακές). Γινόταν απομόνωση του Επωνύμου από

το αρχείου και στη συνέχεια γινόταν αναζήτηση στον πίνακα των Εισηγητών για την εύρεση του αντίστοιχου Id.

Διπλοεγγραφές Πτυχιακών

```
$check_ptyxiaki = mysql_query("SELECT * FROM Ptyxiaki WHERE Title = '{$title}'");
if(!$check_ptyxiaki){
    echo(mysql_error());
    exit();
}
if(mysql_num_rows($check_ptyxiaki)!=0){
    $flag=true;
    while ($row_ptyxiaki = mysql_fetch_array($check_ptyxiaki)){
        if($row_ptyxiaki['Date_submit']==$date_submit && $row_ptyxiaki['Semester_start']==$semester_start){
            $flag=false;
            break 1;
        }
    }
}
if($flag==true){
    $result_ptyxiaki = mysql_query("INSERT INTO Ptyxiaki ".
        "(ID, Id_eisigiti, Title, Date_submit, Semester_start, Team, VS,
        Asked, Given)
        VALUES ($aa, $eisigitis, '$title', '$date_submit', '$semester_start', $team,
        $vs, $asked, $given);");
    if(!$result_ptyxiaki){
        echo(mysql_error());
        exit();
    }
}
}else{
    $result_ptyxiaki = mysql_query("INSERT INTO Ptyxiaki (ID, Id_eisigiti, Title, Date_submit, Semester_start, Team, VS,
        Asked, Given)
        VALUES ($aa, $eisigitis, '$title', '$date_submit', '$semester_start', $team, $vs, $asked,
        $given);");
    if(!$result_ptyxiaki){
        echo(mysql_error());
        exit();
    }
}
}
```

Διπλοεγγραφές Αναθέσεων

```
$check_anathesi_result = mysql_query("SELECT * FROM Anathesi WHERE Title='{title}' AND
                                     Date_submit='{date_submit}' AND Semester_start='{semester_start}");

if(!$check_anathesi_result){
    echo (mysql_error());
    exit();
}

if(mysql_num_rows($check_anathesi_result)!=0){
    $check_anathesi_ptyxiaki_result= mysql_query("SELECT * FROM Ptyxiaki WHERE Title='{title}' AND
                                                Date_submit='{date_submit}' AND
                                                Semester_start='{semester_start}");

    $check_anathesi_ptyxiaki_row = mysql_fetch_array($check_anathesi_ptyxiaki_result);
    $check_anathesi_ptyxiaki_id = $check_anathesi_ptyxiaki_row['ID'];
    $update_query = mysql_query("UPDATE Anathesi SET Am_Foititi2='{sam_foititi}' WHERE
                                Id_ptyxiakis='{check_anathesi_ptyxiaki_id}");
}

else{
    $insert_anathesi = mysql_query("INSERT INTO Anathesi (Title, Id_eisigiti, Am_foititi, Date_submit,
                                                Am_Foititi2, Semester_start, Id_ptyxiakis, Date_start, Olokirothike,
                                                Psyn_olo,
                                                Date_eksetasi, Akyrosi, Psyn_akyrosis, Paratasi, Psyn_paratasis,
                                                New_id_eis,
                                                Psyn_alep, Notes)
                                VALUES ('$title', $eisigitis, '$sam_foititi', '$date_submit', '-', '$semester_start', $saa,
                                '$date_start',
                                $oloklirothike, '$ps_olo', '$date_test', $akyrosi, '$ps_akyrosi', $paratasi,
                                '$ps_paratasi', $id_eisigiti_new, '$ps_al_eis', '$notes')");
}
}
```

Παράβλεψη ειδικών χαρακτήρων

Ο κώδικας για την παράβλεψη ειδικών χαρακτήρων είναι ο εξής:

```
$variable = htmlspecialchars($variable_from_file, ENT_QUOTES);
```

Ουσιαστικά φιλτράρεται η τιμή της μεταβλητής που προέρχεται από το εξωτερικό αρχείο και αποθηκεύεται το αποτέλεσμα στην μεταβλητή **variable**.

6.5 Αξιοσημείωτα *Css* και *Jquery-PHP* κώδικας που χρησιμοποιήθηκε

Focus on Textbox

Κάθε φορά που χρήστης εστιάζει σε ένα textbox προκειμένου να επεξεργαστεί την τιμή που έχει, το textbox αποκτά μία σκιά. Αυτό έγινε εφικτό με χρήση CSS3:

```
.myTextbox {
    border-radius: 0.5em;
    color:#3399FF;
    border:1px solid #297ACC;
    -webkit-transition: box-shadow .3s;
    transition: box-shadow .3s;
    height:25px;
    width: 275px;
}

.myTextbox:focus {
    -moz-box-shadow:0 0 5px 5px #317A7A;
    -webkit-box-shadow:0 0 5px 5px #317A7A;
    box-shadow: 0px 0px 7px #0066FF;
}
```

Έλεγχος για το σωστό τύπου αρχείου

Πριν γίνει η εισαγωγή δεδομένων από εξωτερικό αρχείο, πρέπει να γίνει ένας έλεγχος για το σωστό τύπο του αρχείου. Η λειτουργία αυτή έγινε με Javascript:

```
function checkFile() {
    var file = document.getElementById("myFile").value;

    if (file == ""){
        document.getElementById("errorMsg").innerHTML = "<font color='red'>Δεν
                                                    επιλέχθηκε κάποιο αρχείο</font>";
    }
}
```

```
        return false;
    }

    if (file.split('.').pop() != "txt"){
        document.getElementById("errorMsg").innerHTML = "<font color='red'>Το αρχείο
                                                    πρέπει να είναι τύπου txt (.txt)";
        return false;
    }
}
```

Έλεγχος για συμπλήρωση των απαραίτητων πεδίων

Κάθε φορά που γίνεται η μεμονωμένη εισαγωγή ενός φοιτητή, καθηγητή, πτυχιακής, ανάθεση πτυχιακής, ή η ολοκλήρωση-ακύρωση μιας τρέχουσας ανάθεσης, πρέπει να συμπληρωθούν υποχρεωτικά κάποια πεδία. Αν ο χρήστης κατά λάθος πατήσει ενημέρωση-εισαγωγή χωρίς να συμπληρώσει αυτά τα πεδία τότε το σύστημα δεν θα πρέπει να πραγματοποιεί την ενέργεια αλλά να του εμφανίζει κατάλληλο μήνυμα. Η λειτουργία αυτή έγινε με χρήση Javascript:

```
function validateInput(){
if(document.getElementById("surname").value==null ||
        document.getElementById("surname").value==""){
    document.getElementById("message").style.color = "#ff0000";
    document.getElementById("message").innerHTML = "***Πρέπει να συμπληρώσετε όλα τα
                                                    υποχρεωτικά πεδία";
    return false;
}
```

Έτσι γίνεται έλεγχος για τη συμπλήρωση όλων πληροφοριών που είναι απαραίτητες.

Χρήση ajax με JQuery

Σε αρκετά σημεία του συστήματος υπάρχει ασύγχρονη επικοινωνία (ajax) με τον server για καλύτερη διαδραστικότητα με το χρήστη. Η λειτουργία αυτή πραγματοποιήθηκε εύκολα με τη χρήση JQuery:

```
$.ajax({  
  type: 'POST',  
  url: 'searchStudent.php',  
  data: {'search':this.value},  
  complete: function(r){  
    document.getElementById("searchResult").innerHTML = r.responseText;  
  }  
}
```

Βιβλιογραφία

- [1] Douglas E. Comer - Διαδίκτυα με TCP/IP, Αρχές, πρωτόκολλα και αρχιτεκτονικές, τέταρτη αμερικάνικη έκδοση, 2000
- [2] Στέφανος Γκρίτζαλης, Σωκράτης Κ. Κάσικας, Δημήτρης Γκρίτζαλης, Ασφάλεια Δικτύων Υπολογιστών, Αθήνα 2003
- [3] Μιχάλης Σαλαμπάσης, Εισαγωγή στον Προγραμματισμό Διαδικτυακών Εφαρμογών, 2008
- [4] Luke Welling, Laura Thomson, Ανάπτυξη Web Εφαρμογών με PHP και MySQL, Τέταρτη έκδοση, 2009

Ηλεκτρονική Βιβλιογραφία

- [5] <http://kristiannissen.wordpress.com/2009/03/13/mysqldump-selected-tables/>
- [6] <https://drupal.org/node/2141795>, November 22 2013
- [7] <http://www.toptechietimes.com/2012/05/script-to-check-if-mysql-is-alive-and.html>
- [8] W3C, <http://www.w3.org/>
- [9] Wikipedia, <http://www.wikipedia.org/>
- [10] Εισαγωγή στη JQuery
<http://www.idesigner.gr/%CE%B5%CE%B9%CF%83%CE%B1%CE%B3%CF%89%CE%B3%CE%AE-%CF%83%CF%84%CE%B7%CE%BD-jquery/>, Αύγουστος 4 2011
- [11] About Postgress <http://postgresql.gr/node/3>, Οκτώριος 23 2007
- [12] Εισαγωγή στο Json <http://json.org/json-el.html>