

Πτυχιακή εργασία του φοιτητή Γεωργακόπουλου Βασίλη



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



ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Ανάπτυξη συστήματος διαχείρισης μάθησης (learning management system)



Του φοιτητή

Γεωργακόπουλου Βασίλη

Αρ. Μητρώου: 05/2752

Επιβλέπων καθηγητής

Ψαρράς Νικόλαος

Θεσσαλονίκη 2013

ΠΡΟΛΟΓΟΣ

Μείζον αντικείμενο συζήτησης, αποτελεί σήμερα η προσθήκη νέων τεχνολογιών στην εκπαιδευτική διαδικασία, τεχνολογιών δηλαδή που δύνανται να προσφέρουν μία πιο εμπλουτισμένη εμπειρία μάθησης. Ο μέχρι τώρα παραδοσιακός τρόπος διδασκαλίας και εξέτασης, επεκτείνεται με στοιχεία πολυμεσικής φύσης, τα οποία συντελούν σε μία πιο γρήγορη, εύκολη και άμεση διαδικασία απόκτησης γνώσεων. Χάρη λοιπόν των προαναφερθέντων επεκτάσεων, στο εν λόγω σύγγραμμα θα καταγραφεί η απόπειρα δημιουργίας ενός ηλεκτρονικού συστήματος μάθησης ώστε να μελετηθούν, αφενός τα εργαλεία μέσω των οποίων αυτό μπορεί να υλοποιηθεί και αφετέρου να αναδειχθούν οι δυνατότητες που ένα τέτοιο σύστημα μπορεί να προσφέρει στην εκπαιδευτική διαδικασία.

ΠΕΡΙΛΗΨΗ

Η ευρεία εξάπλωση του διαδικτύου καθώς και των υπηρεσιών που παρέχονται μέσω αυτού, έχει γίνει η αφορμή για την ανάπτυξη συστημάτων εκπαιδευτικού περιεχομένου, συστήματα που με την πάροδο του χρόνου τυγχάνουν ολοένα μεγαλύτερης αναγνώρισης από τα Ακαδημαϊκά Ιδρύματα, ώστε αυτά αποφασίζουν να τα ενσωματώσουν στην εκπαιδευτική τους διαδικασία. Πρόκειται για τα συστήματα που είναι γνωστά με τον όρο Συστήματα Διαχείρισης Μάθησης. Η εκπόνηση αυτής της διπλωματικής εργασίας έχει σκοπό αφενός να αναλύσει έννοιες και στοιχεία που σχετίζονται με αυτά τα ΣΔΜ, αφετέρου να περιγράψει σε πρακτικό επίπεδο την αρχιτεκτονική δομή και τις τεχνολογίες πάνω στις οποίες στηρίζονται και τέλος να αναπτύξει ένα τέτοιο ΣΔΜ για το οποίο θα πραγματοποιηθεί πλήρης και περιεκτική τεκμηρίωση.

ABSTRACT

The wide spread of the Internet and the services provided through it, has become the reason for the development of educational aim systems which over time are becoming increasingly recognized by academic institutions which integrate these systems in their educational process. These systems are known as Learning Management Systems. The preparation of this thesis aims to analyze theoretical concepts and data related to these systems, provide a practical level description of the architectural structure and the technologies upon which these systems are built and finally developing such a learning management system for which there will be a complete and comprehensive documentation.

ΕΥΧΑΡΙΣΤΙΕΣ

Η συγγραφή αυτής της πτυχιακής εργασίας δε θα ήταν δυνατή χωρίς την πολύτιμη βοήθεια κάποιων ανθρώπων. Καταρχάς, θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή μου, κ. Ψαρρά Νίκο για την βοήθειά του κατά την διάρκεια της πορείας της εργασίας

Δεύτερον, θα ήθελα να ευχαριστήσω όλους τους φίλους μου για την αμέριστη συμπαράσταση την οποία έδειξαν σε όποιο πρόβλημα και αν αντιμετώπισα όλο αυτό το διάστημα, δίνοντας μου δύναμη να το αντιμετωπίσω.

Τέλος θα ήθελα να ευχαριστήσω τους γονείς μου για την πολύτιμη υλική, ηθική και πνευματική στήριξή τους.

Κατάλογος περιεχομένων

ΠΡΟΛΟΓΟΣ.....	2
ΠΕΡΙΛΗΨΗ.....	3
ABSTRACT.....	4
ΕΥΧΑΡΙΣΤΙΕΣ	5
ΕΙΣΑΓΩΓΗ.....	10
ΚΕΦΑΛΑΙΟ 1 : Μια εισαγωγική προσέγγιση στα Συστήματα Διαχείρισης Μάθησης (LMS).....	11
1.1 LMS: Οι Βασικές Έννοιες.....	11
1.1.1 Τι είναι το LMS.....	11
1.1.2 Χαρακτηριστικά ενός LMS συστήματος	12
1.1.3 Χρήστες.....	15
1.1.4 Πρότυπα.....	17
1.1.5 Επίλογος.....	18
ΚΕΦΑΛΑΙΟ 2: Οι τεχνολογίες και η αρχιτεκτονική του συστήματος.....	19
2.1 Το Internet.....	19
2.2 Η αρχιτεκτονική client-server.....	20
2.3 Η αρχιτεκτονική client-server σε web-based εφαρμογές.....	20
2.3.1 Ο πελάτης (client).....	21
2.3.2 Ο εξυπηρετητής (server).....	22
2.3.3 HTTP.....	23
2.4 Η αρχιτεκτονική client-server τριών στρωμάτων (3 tier).....	24
2.5 Ο Database Server.....	25
2.6 Συστήματα διαχείρισης βάσεων δεδομένων (DBMS).....	25
2.7 SQL.....	28
2.8 Server Side Scripting και PHP.....	29

2.8.1 Server Side Scripting	29
2.8.2 PHP.....	31
ΚΕΦΑΛΑΙΟ 3: Η Υλοποίηση του Συστήματος.....	32
3.1 Το λειτουργικό σύστημα και η εγκατάσταση των εργαλείων λογισμικού	32
3.1.1 Ο εξυπηρετητής και το λειτουργικό σύστημα.....	32
3.1.2 Σύνδεση με τον εξυπηρετητή.....	33
3.1.3 Εργαλεία λογισμικού.....	37
3.2 Υλοποίηση του βασικού συστήματος.....	38
3.2.1 Η βάση δεδομένων.....	38
3.2.1.1 Οι πίνακες.....	39
3.2.1.2 Οι συσχετίσεις.....	42
3.2.2 Η ανάπτυξη του συστήματος με τη γλώσσα PHP	45
3.2.2.1 Sessions.....	45
3.2.2.2 Σύνδεση με την βάση.....	46
3.2.2.3 Η σελίδα login.....	47
3.2.2.3 Header.php.....	51
3.2.2.4 Η σελίδα του Διαχειριστή (Administrator).....	52
3.2.2.5 Η σελίδα του Καθηγητή.....	60
3.2.2.6 Η σελίδα του Φοιτητή.....	70
3.2.2.7 Βελτιώσεις του συστήματος.....	72
ΣΥΜΠΕΡΑΣΜΑΤΑ.....	76
ΑΝΑΦΟΡΕΣ.....	78
ΠΑΡΑΡΤΗΜΑ ΚΩΔΙΚΑ.....	79

Κατάλογος εικόνων

Εικόνα 1: Client- server αρχιτεκτονική.....	21
Εικόνα 2: Server Side Scripting	31
Εικόνα 3: Επιτυχία σύνδεσης μέσω VPN.....	35
Εικόνα 4: Interface σύνδεσης, μέσω VPN.....	35
Εικόνα 5: Σύνδεση μέσω προγράμματος Putty.....	36
Εικόνα 6: Ταυτοποίηση χρήστη μέσω Putty.....	37
Εικόνα 7: Πίνακες της Βάσης Δεδομένων.....	46
Εικόνα 8: Σελίδα Login.....	49
Εικόνα 9: Σύνδεση με δικαιώματα διαχειριστή.....	52
Εικόνα 10: Σύνδεση με δικαιώματα καθηγητή.....	52
Εικόνα 11: Σύνδεση με δικαιώματα φοιτητή.....	52
Εικόνα 12: Interface σελίδας διαχειριστή.....	53
Εικόνα 13: Προσθήκη μαθήματος	55
Εικόνα 14: Ανάδραση προσθήκης μαθήματος.....	55
Εικόνα 15: Διαθέσιμα μαθήματα.....	56
Εικόνα 16: Ανάδραση ελέγχου username καθηγητή.....	56
Εικόνα 17: Εργαλείο επεξεργασίας Διαθέσιμων Μαθημάτων.....	56
Εικόνα 18: Τα διαθέσιμα μαθήματα μαζί με τα εργαλεία επεξεργασίας.....	57
Εικόνα 19: Αλλαγή κωδικού πρόσβασης.....	58
Εικόνα 20: Συγχρονισμός με LDAP server.....	59
Εικόνα 21: Ανάδραση ελέγχου φοιτητών εκτός του LDAP.....	59
Εικόνα 22: Διαγραφή φοιτητών.....	60
Εικόνα 23: Φόρμα Διαθέσιμων Μαθημάτων.....	61
Εικόνα 24: Το περιβάλλον διαχείρισης ενός τμήματος.....	61

Εικόνα 25: Η φόρμα δημιουργίας τμήματος.....	62
Εικόνα 26: Προσθήκη Φοιτητή σε τμήμα.....	63
Εικόνα 27: Ανάδραση ελέγχου εγγραφής φοιτητή.....	64
Εικόνα 28: Ανάδραση ελέγχου εγγραφής φοιτητή σε Μάθημα.....	64
Εικόνα 29: Εγγεγραμμένοι Φοιτητές σε μάθημα.....	64
Εικόνα 30: Το interface των Αποτελεσμάτων Φοιτητών.....	65
Εικόνα 31: Αποτελέσματα φοιτητών μετά από εξέταση.....	65
Εικόνα 32: Η φόρμα για την προσθήκη νέας ερώτησης.....	66
Εικόνα 33: Ερωτήσεις ερωτηματολογίου και επεξεργασία.....	67
Εικόνα 34: Η φόρμα δημιουργίας ερωτηματολογίου.....	68
Εικόνα 35: Καταχωρημένα ερωτηματολόγια.....	69
Εικόνα 36: Ερωτηματολόγιο προς εξέταση.....	70
Εικόνα 37: Διεξαγωγή εξέτασης.....	70
Εικόνα 38: Φόρμα Εξέτασης.....	71
Εικόνα 39: Φόρμα Βαθμολογιών.....	71
Εικόνα 40: Βελτιστοποίηση σελίδας Login.....	73
Εικόνα 41: Είσοδος ως καθηγητής με επιλογή μετάβασης στο περιβάλλον εργασίας του φοιτητή	73
Εικόνα 42: Είσοδος ως φοιτητής με επιλογή μετάβασης στο περιβάλλον εργασίας του καθηγητή	74
Εικόνα 43: Το σύνολο των καταχωρημένων τμημάτων.....	75
Εικόνα 44: Καταχωρημένα τμήματα του καθηγητή με username psarnik.....	75
Εικόνα 45: Καταχωρημένα τμήματα καθηγητή με username peris.....	75

ΕΙΣΑΓΩΓΗ

Ο σκοπός της συγκεκριμένης πτυχιακής εργασίας είναι η ανάπτυξη ενός συστήματος διαχείρισης μάθησης σε περιβάλλον UNIX που θα στοχεύει στο να παρέχει την δυνατότητα αξιολόγησης των φοιτητών του τμήματος Πληροφορικής του Αλεξάνδρειου Τεχνολογικού Εκπαιδευτικού Ιδρύματος Θεσσαλονίκης, με τη χρήση on-line ερωτηματολογίων πολλαπλής επιλογής.

Στο πρώτο κεφάλαιο θα πραγματοποιηθεί μία θεωρητική προσέγγιση βασικών εννοιών που σχετίζονται άμεσα με τα συστήματα διαχείρισης μάθησης ενώ παράλληλα θα αναφερθούν και τα βασικά τους χαρακτηριστικά.

Στο δεύτερο κεφάλαιο θα περιγραφούν οι τεχνολογίες και η αρχιτεκτονική δομή πάνω στην οποία στηρίζεται η ανάπτυξη αυτών των συστημάτων με ταυτόχρονη αναφορά στα εργαλεία λογισμικού που τα υλοποιούν.

Στο τρίτο κεφάλαιο θα παρατεθούν οι οδηγίες εγκατάστασης των εργαλείων λογισμικού τα οποία αναφέρθηκαν στην προηγούμενη ενότητα ενώ στη συνέχεια θα πραγματοποιηθεί μία λεπτομερής αναφορά των βημάτων που ακολουθήθηκαν για την ανάπτυξη του συστήματος με τη χρήση των εργαλείων αυτών.

ΚΕΦΑΛΑΙΟ 1 : Μια εισαγωγική προσέγγιση στα Συστήματα Διαχείρισης Μάθησης (LMS)

ΕΙΣΑΓΩΓΗ

Στο κεφάλαιο αυτό θα πραγματοποιηθεί μία θεωρητική προσέγγιση στα συστήματα διαχείρισης μάθησης κατά την οποία θα μελετηθούν βασικές έννοιες που σχετίζονται με αυτά, ενώ παράλληλα θα γίνει προσπάθεια αναγνώρισης των βασικών τους χαρακτηριστικών. Στη συνέχεια, θα παρατεθούν οι κατηγορίες χρηστών που θεσπίζονται σε αυτά τα συστήματα και ο ρόλος που η κάθε μία από αυτές κατέχει, ενώ τέλος θα αναφερθούν μοντέλα προτυποποίησης που έχουν αναπτυχθεί για τα συστήματα αυτά.

1.1 LMS: Οι Βασικές Έννοιες

1.1.1 Τι είναι το LMS

Πριν γίνει οποιαδήποτε απόπειρα ορισμού των συστημάτων LMS, κρίνεται σκόπιμο να παρουσιαστούν ορισμένες έννοιες οι οποίες σχετίζονται άμεσα με την φιλοσοφία αυτών των συστημάτων οι οποίες αναφέρονται παρακάτω.

Η Εκπαιδευτική Τεχνολογία : Η εκπαιδευτική τεχνολογία είναι ένας συστηματικός τρόπος σύλληψης, πραγματοποίησης και αξιολόγησης του συνόλου της διαδικασίας μάθησης και διδασκαλίας σε σχέση με τους παιδαγωγικούς στόχους που απορρέουν από την έρευνα στους τομείς της ανθρώπινης μάθησης και της επικοινωνίας. Χρησιμοποιεί ένα συνδυασμό υλών-πηγών ανθρωπίνων και μη προκειμένου να προκαλέσει αποτελεσματική διδασκαλία[1].

Η Ηλεκτρονική Μάθηση : Ως ηλεκτρονική μάθηση ορίζεται ο τρόπος διδασκαλίας που διατυπώνεται μέσω ψηφιακών μηχανισμών όπως είναι οι ηλεκτρονικοί υπολογιστές ή άλλες ηλεκτρονικές συσκευές, οι οποίες αποσκοπούν να ενισχύσουν την διαδικασία μάθησης[2].

Έχοντας λοιπόν αποδώσει αυτές τις χρήσιμες έννοιες στον αναγνώστη, στη συνέχεια θα γίνει μία προσπάθεια αναγνώρισης και ορισμού των συστημάτων LMS.

Σύμφωνα με τους *Szabo και Flesher* [3], ένα σύστημα LMS αποτελεί μια υποδομή η οποία παρέχει και διαχειρίζεται το εκπαιδευτικό περιεχόμενο, προσδιορίζει και αποτιμά τους στόχους εκπαίδευσης και κατάρτισης σε μεμονωμένο επίπεδο αλλά και επίπεδο ενός οργανισμού, παρακολουθεί την πρόοδο που επιτυγχάνεται κατά την πορεία επίτευξης των στόχων αυτών και τέλος, συλλέγει και παρουσιάζει στοιχεία τα οποία βοηθούν στην εποπτεία της διαδικασίας μάθησης ενός οργανισμού ως σύνολο.

Σύμφωνα πάλι με έναν άλλο ορισμό ο οποίος δίνεται από τους Hamish Coates, Richard James και Gabrielle Baldwin[4], τα LMS είναι συστήματα Ηλεκτρονικών Υπολογιστών τα οποία βασίζονται στο Διαδίκτυο (Internet) και ενσωματώνουν ένα ευρύ φάσμα παιδαγωγικών εργαλείων καθώς και εργαλείων διαχείρισης μαθημάτων. Τα συστήματα αυτά έχουν την ικανότητα να δημιουργούν εικονικά περιβάλλοντα μάθησης λαμβάνοντας ως κριτήριο τις απαιτήσεις των ατόμων που συμμετέχουν στην εκπαιδευτική διαδικασία (καθηγητές και μαθητές).

Χρησιμοποιώντας μία πιο τεχνική προσέγγιση οι Baumgartner, Häfele, και Maier-Häfele[5] υποστηρίζουν ότι το LMS (ή όπως οι ίδιοι το αποκαλούν e-learning platform), είναι ένα κομμάτι λογισμικού το οποίο εγκαθίσταται από την πλευρά ενός υπολογιστή-εξυπηρετητή (server-side installed software), το οποίο παρέχει βοήθεια στην διδασκαλία οποιουδήποτε μαθησιακού υλικού μέσω του Διαδικτύου καθώς και την οργάνωση των απαραίτητων διαδικασιών.

1.1.2 Χαρακτηριστικά ενός LMS συστήματος

Σε αυτήν την ενότητα θα γίνει μια προσπάθεια καταγραφής των στοιχείων τα οποία αποτελούν τα πιο σημαντικά και απαραίτητα χαρακτηριστικά ενός συστήματος LMS. Στην προσπάθεια της αναγνώρισης αυτών των χαρακτηριστικών λοιπόν, θα χρησιμοποιηθούν τοποθετήσεις που προκύπτουν από επιστημονική έρευνα ατόμων και οργανισμών που πρόσκεινται στον τομέα των τεχνολογιών των συστημάτων LMS και της ηλεκτρονικής μάθησης γενικότερα.

Σύμφωνα με τους McCormack και Jones[6], τα χαρακτηριστικά αυτά συνοψίζονται σε τέσσερις βασικές κατηγορίες λειτουργιών τις οποίες πρέπει να είναι σε θέση να προσφέρουν τα συστήματα LMS ώστε, μέσω αυτών, να ικανοποιούνται οι απαραίτητες ανάγκες για την σωστή διεξαγωγή της εκπαιδευτικής διαδικασίας.

Οι τέσσερις αυτές βασικές λειτουργίες αναφέρονται παρακάτω:

➤ *Διανομή Πληροφοριών:*

Περιλαμβάνει τις λειτουργίες του συστήματος με τις οποίες είναι δυνατή η κοινοποίηση και η παράθεση πληροφοριών, των οποίων η γνώση είναι απαραίτητη στα άτομα που συμμετέχουν στην εκπαιδευτική διαδικασία. Παραδείγματα αυτής της κατηγορίας αποτελούν το ημερολόγιο, το γλωσσάρι, η χρήσιμη πληροφορία της ημέρας (tip of the day) καθώς και ο πίνακας (περιοχή) ανακοινώσεων.

➤ *Διαχείριση του διδακτικού υλικού:*

Εμπεριέχει τις λειτουργίες οι οποίες επιτρέπουν σ' ένα χρήστη του συστήματος να διαχειριστεί, να προσαρμόσει καθώς και να τροποποιήσει στοιχεία τα οποία σχετίζονται με το περιεχόμενο και τον τρόπο που το σύστημα παρουσιάζει τα δεδομένα στον χρήστη. Παραδείγματα αυτής της κατηγορίας αποτελούν οι λειτουργίες όπως:

- Παραμετροποίηση της διεπαφής χρήστη ως προς τις ανάγκες του.
- Ενημέρωση καθώς και αναβάθμιση των μαθησιακών πόρων.

➤ *Πολλαπλές διευκολύνσεις επικοινωνίας:*

Αποτελεί το σύνολο των διαφορετικών μεθόδων τις οποίες ενσωματώνει ένα σύστημα LMS ώστε να επιτευχθεί η επικοινωνία μεταξύ των χρηστών του. Χαρακτηριστικό παράδειγμα αυτής της κατηγορίας αποτελούν οι λειτουργίες σύγχρονης και ασύγχρονης επικοινωνίας.

- Σύγχρονη επικοινωνία: Η επικοινωνία πραγματοποιείται σε πραγματικό χρόνο (π.χ. Online chat, instant messaging).
- Ασύγχρονη επικοινωνία: Η επικοινωνία δεν πραγματοποιείται σε πραγματικό χρόνο καθώς η αποστολή και η λήψη του μηνύματος γίνονται ετεροχρονισμένα (π.χ. E-mail, forums συζητήσεων).

➤ *Διαχείριση της (εικονικής) τάξης:*

Στην κατηγορία αυτή τοποθετούνται όλες οι λειτουργίες οι οποίες σχετίζονται με την διαχείριση των δραστηριοτήτων της (εικονικής) τάξης, με τις οποίες εμπλέκονται τα άτομα που φέρουν τον ρόλο των μαθητών και γενικά των ατόμων που λειτουργούν ως δέκτες στην εκπαιδευτική διαδικασία μάθησης

μέσω των συστημάτων LMS. Μερικά παραδείγματα αυτών των λειτουργιών είναι τα εξής:

- Online βαθμολόγηση σε αξιολόγηση των μαθητών.
- Παρακολούθηση της συμμετοχής των μαθητών στο μάθημα.
- Διαχείριση των προφίλ των μαθητών που συμμετέχουν στο μάθημα.

Μια πιο πρόσφατη και τεχνική προσέγγιση από το American Society for Training & Development[7] συνιστά ως χαρακτηριστικά ενός LMS συστήματος τις εξής λειτουργικές απαιτήσεις κατά τις οποίες ένα σύστημα LMS πρέπει να:

- Επιτρέπει την ενοποίηση με το ανθρώπινο δυναμικό του συστήματος
- Ενσωματώνει εργαλεία τα οποία να επιτρέπουν στην αρχή διαχείρισης του συστήματος (π.χ. Administrator) να:
 - Διαχειρίζεται την εγγραφή των χρηστών και την επεξεργασία των προφίλ τους.
 - Ορίζει προγράμματα σπουδών και διαδικασίες πιστοποίησης.
 - Εκχωρεί διδάσκοντες και εκπαιδευτικό υλικό.
 - Διαχειρίζεται τους προϋπολογισμούς.
 - Προετοιμάζει χρονοδιαγράμματα για τους διδάσκοντες, τους μαθητές και τις αίθουσες.
- Παρέχει πρόσβαση στην παράδοση των μαθημάτων από τις πτυχές του μέσου παράδοσης (π.χ. Αίθουσα ή online), της μεθόδου (καθοδηγούμενη από κάποιον διδάσκοντα ή self-paced) και των διδασκομένων.
- Είναι σε θέση να αναπτύσσει μαθησιακό περιεχόμενο συμπεριλαμβανομένων και των λειτουργιών συγγραφής, συντήρησης (maintaining) και αποθήκευσης.
- Παρέχει την δυνατότητα ενοποίησης του μαθησιακού περιεχομένου του LMS με το παραδοσιακό υλικό μάθησης του εκπαιδευτικού οργανισμού στο οποίο εγκαθίσταται το σύστημα.
- Αξιολογεί πιθανόν ελλείψεις ικανοτήτων σε μαθητές και να τους υποβοηθά στην απόκτηση δεξιοτήτων.
- Προσφέρει καθώς και να υποστηρίζει την δυνατότητα δημιουργίας δραστηριοτήτων αξιολόγησης (π.χ. Τεστ πολλαπλής επιλογής).
- Συμμορφώνεται με πρότυπα όπως το IMS και το AICC ώστε να επιτρέπεται η ενσωμάτωση περιεχομένου και μαθησιακού υλικού τα οποία υπακούν στα

πρότυπα αυτά ανεξάρτητα από το σύστημα το οποίο τα δημιούργησε (π.χ. ένα άλλο σύστημα LMS).

- Υποστηρίζει την κατάλληλη διαμόρφωσή του ώστε να μπορεί να συνεργάζεται με ήδη υπάρχοντα συστήματα του ακαδημαϊκού φορέα.
- Παρέχει ασφάλεια με λειτουργίες δημιουργίας κωδικών (password) και κρυπτογράφησης.

1.1.3 Χρήστες

Παρατηρώντας προηγούμενες ενότητες αυτού του κεφαλαίου, γίνεται αντιληπτό ότι το σύνολο το ατόμων τα οποία σχετίζονται με το σύστημα, διαχωρίζονται σε κατηγορίες ανάλογα με τον τρόπο διάδρασής τους με το σύστημα καθώς και τις απαιτήσεις τους από αυτό.

Οι κατηγορίες αυτές ονομάζονται *ρόλοι των χρηστών* και αποτελούν ένα συστατικό στοιχείο των συστημάτων LMS καθώς, μέσω αυτών, αναγνωρίζονται και προσδίδονται αρμοδιότητες στους χρήστες, ανάλογα με τον ρόλο τους, ενώ παράλληλα καθορίζονται και τα όρια του συστήματος μέσα στα οποία ο κάθε χρήστης μπορεί να ενεργεί.

Στην ενότητα αυτή θα γίνει μία προσπάθεια αναφοράς των ρόλων των χρηστών που υπάρχουν σε ένα σύστημα LMS, ενώ παράλληλα θα δοθεί και μια σύντομη και γενικευμένη περιγραφή του καθενός από αυτούς τους ρόλους.

Μαθητής:

Αποτελεί ίσως την πιο σημαντική κατηγορία χρηστών καθώς, ένας από τους σημαντικότερους στόχους ενός συστήματος LMS κατά την σχεδίαση του, είναι να μπορεί να ικανοποιήσει τις ανάγκες και τις απαιτήσεις των μαθητών υπό την οπτική ότι, η επίτευξη όλων των παραπάνω οδηγεί αφενός σε βελτιστοποίηση της αποτελεσματικότητας της διδασκαλίας και αφετέρου στη δημιουργία ενός εύχρηστου συστήματος, στοιχεία τα οποία αποτελούν σημαντικούς παράγοντες για την δημιουργία ενός επιτυχημένου συστήματος LMS.

Ο μαθητής δημιουργεί μία αλληλεπίδραση με το σύστημα υπό την οπτική του συστήματος ως ενός ή περισσότερων μαθημάτων με τα οποία ο μαθητής σχετίζεται και μίας ή περισσότερων τάξεων (class) στις οποίες ανήκει. Στη συνέχεια, μέσω της παραπάνω αλληλεπίδρασης, ο μαθητής αποκτά πρόσβαση σε διδακτικό υλικό, μέσω του οποίου αποκτά γνώση για ένα συγκεκριμένο εκπαιδευτικό τομέα, με χρήση εργαλείων και τεχνολογιών τα οποία του παρέχονται. Τέλος, με βάση τις γνώσεις που έχει αποκτήσει από τα παραπάνω, ο

μαθητής διέρχεται από μία διαδικασία αποτίμησης της προόδου του κατά την οποία, οι δεξιότητες και η κατάρτιση του δοκιμάζονται μέσω διαδικασιών αξιολόγησης όπως, παραδείγματος χάριν, τεστ πολλαπλών επιλογών και ομαδικές εργασίες.

Καθηγητής:

Στην κατηγορία αυτή ανήκει η ομάδα των ατόμων η οποία είναι υπεύθυνη για την εκπαίδευση και την κατάρτιση της ομάδας των μαθητών που αναφέρθηκε παραπάνω. Διδάσκοντες καθηγητές και βοηθοί καθηγητών αποτελούν τα βασικότερα παραδείγματα αυτής της ομάδας.

Ο καθηγητής δημιουργεί, όπως και ο μαθητής παραπάνω, μια αλληλεπίδραση με το σύστημα κατά τη οποία, του αποδίδεται η ευθύνη διδασκαλίας ενός ή περισσότερων μαθημάτων στα οποία είναι εγγεγραμμένη μια ομάδα μαθητών.

Στα πλαίσια λοιπόν όλων των προαναφερθέντων ο καθηγητής, χρησιμοποιώντας εργαλεία και λειτουργίες τα οποία του παρέχει το σύστημα, επωμίζεται την επίτευξη δύο βασικών στόχων:

➤ *Διαχείριση του μαθησιακού περιεχομένου:*

Ο καθηγητής αναλαμβάνει να δημιουργήσει ένα αρχικό περιεχόμενο γνώσης το οποίο θα παρέχεται στους μαθητές, το οποίο στη συνέχεια έχει την ευθύνη να ανανεώνει και να εμπλουτίζει ώστε να συμβαδίζει με τυχόν νέα δεδομένα τα οποία μπορεί να προκύψουν για ένα εκπαιδευτικό τομέα.

➤ *Αξιολόγηση των μαθητών:*

Ο καθηγητής είναι επίσης υπεύθυνος για την παρακολούθηση αλλά και για την εκτίμηση της προόδου την οποία έχουν σημειώσει οι μαθητές του. Για τον λόγο αυτό λοιπόν, πρέπει να δημιουργήσει μια σειρά από εργασίες και εξετάσεις τις οποίες θα αναθέσει στους μαθητές του ώστε, μέσω αυτών να ενημερώνεται για την κατάστασή τους.

Διαχειριστής (Administrator):

Στην κατηγορία αυτή ανήκουν τα άτομα τα οποία αναλαμβάνουν την ευθύνη της συνολικής διαχείρισης του συστήματος ώστε να εξασφαλίζεται η σωστή λειτουργία του καθώς και η εξυπηρέτηση των αναγκών όλων των χρηστών.

Αν και αποτελεί μία από τις ομάδες χρηστών, ο ίδιος δεν μπορεί να αποκαλεστεί “χρήστης”. Η βασική διαφορά του από τις υπόλοιπες ομάδες αποτελεί το γεγονός ότι ο διαχειριστής έχει πρόσβαση σε μια εσωτερική και ολοκληρωμένη εικόνα του συστήματος στην οποία οι υπόλοιποι χρήστες δεν έχουν πρόσβαση. Μέσω αυτής,

επενεργεί ώστε να καθορίσει τις ρυθμίσεις και τις παραμέτρους του συστήματος από τα πιο βασικά συστατικά του (π.χ. πρόγραμμα σπουδών, μαθήματα, τάξεις, χρήστες και τις μεταξύ τους σχέσεις) μέχρι και τις πιο μικρές του λεπτομέρειες (π.χ. το UI ενός χρήστη).

Αντιλαμβάνεται λοιπόν κανείς ότι ο διαχειριστής δεν είναι ένας απλός χρήστης ο οποίος “χειρίζεται” μία όψη του συστήματος αλλά ένας *υπερχρήστης (superuser)* ο οποίος έχει την ικανότητα “διαχείρισης” του συστήματος στην ολότητά του.

Επισκέπτης (Guest):

Ορισμένα συστήματα ορίζουν την ύπαρξη μιας τέταρτης κατηγορίας χρηστών, αυτής των επισκεπτών.

Οι επισκέπτες αποτελούν άτομα τα οποία δεν ανήκουν σε καμία από τις παραπάνω κατηγορίες και στους οποίους παρέχεται ένα περιορισμένο σε σχέση με τις υπόλοιπες ομάδες επίπεδο διάδρασης με το σύστημα, ώστε να αποκτήσουν μία πρώτη επαφή με αυτό και να εξοικειωθούν με τον τρόπο λειτουργίας του.

Κλείνοντας αυτήν την ενότητα, αξίζει να σημειωθεί ότι τα παραπάνω αποτελούν μια γενική προσέγγιση για τις αρμοδιότητες και τα όρια της κάθε μία ομάδας καθώς, κάθε ένα σύστημα LMS μπορεί να παράγεται από διαφορετική εταιρία με διαφορετικές προτεραιότητες και απόψεις για την συμπεριφορά του συστήματός τους. Οπότε, έχοντας υπ' όψιν όλα τα προαναφερθέντα, οι αρμοδιότητες καθώς και τα όρια της κάθε ομάδας ενδέχεται να είναι διαφορετικά από σύστημα σε σύστημα.

1.1.4 Πρότυπα

Σε προηγούμενη ενότητα αναγνωρίστηκε ως χαρακτηριστικό ενός LMS συστήματος, η ικανότητα του να προσαρμόζεται και να παρέχει υποστήριξη σε πρότυπα μοντέλα . Σε αυτήν την ενότητα θα αναφερθούν μερικά από αυτά τα πρότυπα ενώ, παράλληλα, θα δοθεί μία σύντομη περιγραφή του σκοπού τον οποίο εξυπηρετούν.

AICC [8]

Τα αρχικά AICC αν και παρεξηγημένα θεωρούνται ως η ονομασία του προτύπου αποτελούν, στην ουσία, την ονομασία του οργανισμού ο οποίος το ανέπτυξε (Aviation Industry Computer-Based Training Committee).

Ο οργανισμός αυτός απαρτίζεται από ομάδες ειδικών οι οποίοι ασχολούνται με δημιουργία κατευθυντηρίων γραμμών για την δημιουργία, παράδοση και αξιολόγηση εργαλείων κατάρτισης και εκπαίδευσης στον τομέα της βιομηχανίας αερομεταφορών με χρήση τεχνολογιών ηλεκτρονικής μάθησης.

Αν και ο οργανισμός δημιουργήθηκε ώστε να λύσει θέματα του τομέα εκπαίδευσης στις αερομεταφορές, η χρήση αφαιρετικών και γενικευμένων προδιαγραφών, έκανε ευρέως αποδεκτή την χρήση των προτύπων του από μη σχετιζόμενες με αερομεταφορές εταιρίες.

Το πρότυπο αναπτύσσει μία σειρά από κατευθυντήριες οδηγίες με τις οποίες, ορίζεται ο τρόπος επικοινωνίας και ανταλλαγής πληροφοριών μεταξύ των στοιχείων μαθησιακού περιεχομένου, τα οποία ορίζει ως αντικείμενα μάθησης (learning objects), και του συστήματος μάθησης το οποίο διαχειρίζεται το εν λόγω περιεχόμενο.

Η πρώτη μορφή του προτύπου αυτού, εκδόθηκε το 1989 και αφορούσε διαδικασίες εκπαίδευσης με χρήση υπολογιστών συστημάτων. Αργότερα όμως το 1998 το πρότυπο εμπλουτίστηκε με μία διεπιφάνεια διαδικτυακής επικοινωνίας, βασισμένη στο πρωτόκολλο HTTP (HACP), ενώ ένα χρόνο αργότερα προστέθηκε και μία διεπιφάνεια η οποία βασιζόταν στην γλώσσα διαδικτυακού προγραμματισμού JavaScript, κάνοντας έτσι το πρότυπο συμβατό με τα σύγχρονα συστήματα LMS τα οποία λειτουργούν σε διαδικτυακή βάση (web-based)[9].

IMS Learning Design

Το εν λόγω πρότυπο δημιουργήθηκε από τον οργανισμό IMS Global Learning Consortium το 2003 και βασίζεται πάνω στο Educational Modeling Language (EML) το οποίο αναπτύχθηκε από το ανοικτό πανεπιστήμιο της Ολλανδίας (OUNL) [10].

Το πρότυπο ορίζει την προδιαγραφή μιας συνοπτικής περιγραφικής γλώσσας με την χρήση της οποίας να είναι δυνατή η μοντελοποίηση σεναρίων μάθησης, ώστε να μπορούν να ενσωματωθούν και να επαναχρησιμοποιηθούν από συστήματα ηλεκτρονικής μάθησης, όπως είναι παραδείγματος χάριν τα συστήματα LMS. Επιπρόσθετα, η μεθοδολογία περιγραφής των εν λόγω σεναρίων μάθησης δεν είναι δεσμευμένη με βάση κάποιο συγκεκριμένο παιδαγωγικό μοντέλο καθώς, η προδιαγραφή ορίζει μια αφαιρετική γλώσσα υψηλού επιπέδου ή αλλιώς ένα (μοναδικό) μετά-μοντέλο το οποίο να μπορεί να περιγράψει όλα τα υπόλοιπα διαφορετικά παιδαγωγικά μοντέλα[A1].

Τέλος η χρήση της γλώσσας XML ως βάση για την γλώσσα περιγραφής των μαθησιακών σεναρίων, καθιστά δυνατή την ανάγνωσή τους από τα υπολογιστικά συστήματα και τα συστήματα LMS[10].

1.1.5 Επίλογος

Σε αυτό το σημείο ολοκληρώνεται το κομμάτι του θεωρητικού υποβάθρου το οποίο αφορά στα συστήματα LMS. Στην επόμενη ενότητα θα πραγματοποιηθεί μία ανάλυση του αρχιτεκτονικού και τεχνολογικού υποβάθρου με βάση τα οποία αυτά τα συστήματα δημιουργούνται.

ΚΕΦΑΛΑΙΟ 2: Οι τεχνολογίες και η αρχιτεκτονική του συστήματος

ΕΙΣΑΓΩΓΗ

Στο κεφάλαιο αυτό θα γίνει μία αναφορά στο αρχιτεκτονικό μοντέλο client-server καθώς και τον τρόπο που αυτό εφαρμόζεται σε επίπεδο διαδικτυακής εφαρμογής. Παράλληλα θα γίνει και μία σύντομη αναφορά στο θεμελιώδες στοιχείο των διαδικτυακών εφαρμογών, το διαδίκτυο. Τέλος θα συζητηθούν τα θέματα της αρχιτεκτονικής client-server τριών στρωμάτων καθώς και ο λόγος που την επιλέγουμε, τα δομικά στοιχεία από τα οποία αποτελείται καθώς και τα εργαλεία λογισμικού που υλοποιούν τα στοιχεία αυτά, ενώ επιπρόσθετα θα αναφερθεί ποια από τα εργαλεία θα επιλεγθούν για την ανάπτυξη του συστήματος της πτυχιακής εργασίας.

2.1 Το Internet

Το διαδίκτυο ή το Internet όπως είναι πιο γνωστό, αποτελεί ένα πολύπλοκο σύστημα επικοινωνίας στο οποίο ορίζεται μια ιεραρχική δομή δικτύων ηλεκτρονικών υπολογιστών παγκόσμιας κλίμακας, τα οποία διασυνδέονται μεταξύ τους με πληθώρα τεχνολογιών δικτύωσης (χάλκινα καλώδια, οπτικές ίνες, δορυφορικές συνδέσεις). Έτσι, δημιουργείται ένα μεγαλύτερο δίκτυο ή αλλιώς ένα υπερδίκτυο (*supernet*) το οποίο, επιτρέπει την μεταφορά δεδομένων μεταξύ αυτών των δικτύων υπολογιστών.

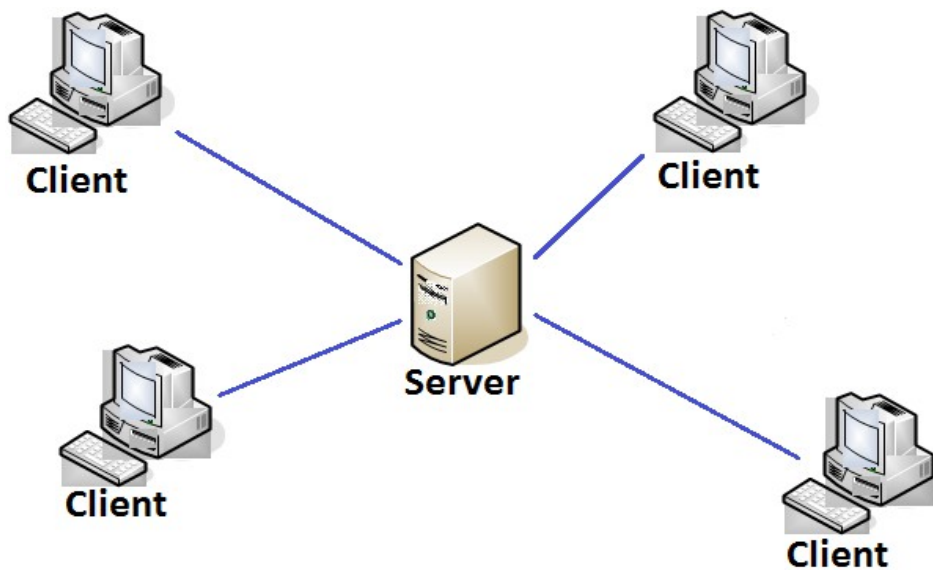
Αν και η ανάπτυξη του ξεκίνησε το 1960 από τις ΗΠΑ για στρατιωτικούς σκοπούς, πολύ σύντομα υποστηρίχθηκε και υιοθετήθηκε από τα ακαδημαϊκά ιδρύματα, γεγονός το οποίο αποτέλεσε την απαρχή γνωστοποίησης του προς το κοινό. Έτσι λοιπόν, οδηγήθηκε στη σημερινή μορφή του που πλέον προσφέρει τις υπηρεσίες του σε δισεκατομμύρια ανθρώπους.

Θεωρείται μία από τις σημαντικότερες, αν όχι η σημαντικότερη, τεχνολογία της επιστήμης της πληροφορικής καθώς αυτή τη στιγμή το internet αποτελεί το μεγαλύτερο σύστημα διακίνησης και εύρεσης πληροφοριών στον κόσμο, γεγονός που το καθιστά ένα απαραίτητο στοιχείο της καθημερινής ζωής.

Ο λόγος για τον οποίο αναφέρεται στο συγκεκριμένο κεφάλαιο αποτελεί το γεγονός ότι πέρα όλων των προαναφερθέντων, το internet αποτελεί και αναπόσπαστο συστατικό στοιχείο των σύγχρονων συστημάτων LMS και γενικότερα οποιασδήποτε web-based εφαρμογής, καθώς αποτελεί το μέσο επικοινωνίας (medium) μέσω του οποίου πραγματοποιείται η ανταλλαγή των πληροφοριών μεταξύ των συμμετεχόντων του συστήματος, καθιστώντας έτσι εφικτή την διαδικασία μάθησης.

2.2 Η αρχιτεκτονική client-server

Η αρχιτεκτονική client-server περιγράφει ένα καταναμημένο σύστημα επικοινωνίας και διαμοιρασμού πόρων στο οποίο, εφαρμογές λογισμικού ενός ή περισσότερων υπολογιστικών συστημάτων (συνήθως προσωπικοί υπολογιστές) τα οποία ονομάζονται *clients*, αποστέλλουν μέσω ενός κοινού καναλιού επικοινωνίας αιτήσεις για την απόκτηση ενός συγκεκριμένου περιεχομένου προς ένα απομακρυσμένο υπολογιστικό σύστημα, το οποίο ονομάζεται *server*. Ο *server* από την πλευρά του συντηρεί μια πληθώρα εφαρμογών (ανάλογα με το πλήθος των υπηρεσιών που εξυπηρετεί) οι οποίες επεξεργάζονται τις αιτήσεις των αντίστοιχων εφαρμογών πελάτη και επιστρέφουν το αιτούμενο περιεχόμενο σε αυτές.



Εικόνα 1: Client- server αρχιτεκτονική

2.3 Η αρχιτεκτονική client-server σε web-based εφαρμογές

Στην ενότητα αυτή θα περιγραφεί ο τρόπος με τον οποίο η αρχιτεκτονική client-server η οποία αναλύθηκε στην προηγούμενη ενότητα, μπορεί να χρησιμοποιηθεί για να οριστεί ο τρόπος επικοινωνίας μεταξύ των στοιχείων που συμμετέχουν στην υλοποίηση μιας διαδικτυακής(web-based) εφαρμογής. Στη συνέχεια θα πραγματοποιηθεί η ανάλυση των δομικών συστατικών της αρχιτεκτονικής στα πλαίσια της διαδικτυακής εφαρμογής καθώς και τα εργαλεία λογισμικού τα οποία υλοποιούν το κάθε ένα από τα συστατικά αυτά.

2.3.1 Ο πελάτης (client)

Σε μια διαδικτυακή εφαρμογή όπως είναι παραδείγματος χάριν τα συστήματα ηλεκτρονικής μάθησης και κατ' επέκταση τα συστήματα LMS, το κομμάτι της εφαρμογής πελάτη αντιπροσωπεύει η χρήση ενός γραφικού αλληλεπιδραστικού περιβάλλοντος (GUI), μέσω του οποίου οι χρήστες μπορούν να αλληλεπιδρούν με την εφαρμογή, να εισάγουν τα επιθυμητά δεδομένα καθώς και να παρατηρούν το οπτικό αποτέλεσμα των δεδομένων που εισήγαγαν.

Όλες τις προαναφερθείσες λειτουργίες υλοποιεί το πρόγραμμα περιήγησης στον παγκόσμιο ιστό (world wide web) ή αλλιώς ο web browser. Αυτός βρίσκεται εγκατεστημένος σε κάποιο υπολογιστικό σύστημα (συνήθως προσωπικός υπολογιστής) το οποίο διαθέτει επικοινωνία με το διαδίκτυο, (Internet) μέσω του οποίου μπορεί να αποστέλλει τις αιτήσεις προς τον απομακρυσμένο εξυπηρετητή.

Η εφαρμογή του web browser ως εφαρμογή πελάτη διαθέτει μια πολύ περιορισμένη λειτουργικότητα καθώς, οι μόνες αρμοδιότητές της είναι η αποστολή αιτήσεων για την προσπέλαση κάποιας συγκεκριμένης πληροφορίας προς τον εξυπηρετητή και η εμφάνιση της πληροφορίας που αυτός επιστρέφει στον χρήστη. Για το λόγο αυτό, πολλές φορές αποκαλείται και ως “λεπτός” πελάτης (thin client). Αυτός ο περιορισμός όμως, ταυτόχρονα αποτελεί ένα σημαντικό θετικό στοιχείο για την αρχιτεκτονική αυτή καθώς, όλη η διαδικασία της επεξεργασίας της πληροφορίας και γενικά η λογική του προγραμματισμού πραγματοποιείται στην πλευρά του εξυπηρετητή. Έτσι, απαλλάσσει την εφαρμογή πελάτη από οποιαδήποτε παραμετροποίηση, ρύθμιση και γενικά φόρτο εργασίας. Αυτό μας επιτρέπει λοιπόν να επιλέξουμε την εφαρμογή πελάτη χωρίς να μας απασχολούν στοιχεία όπως ποιο πρόγραμμα web-browser θα επιλέξουμε και σε ποια πλατφόρμα ή λειτουργικό σύστημα αυτό θα εκτελείται.

Για την επικοινωνία μεταξύ web browser και του εξυπηρετητή, χρησιμοποιείται το πρωτόκολλο *HTTP* για το οποίο θα πραγματοποιηθεί αναλυτική περιγραφή σε επόμενη ενότητα.

Για την επιλογή ενός web browser, διατίθεται μια μεγάλη γκάμα επιλογών με γνωστότερες στο κοινό τον *Mozilla Firefox*, τον *Opera*, τον *Google Chrome*, τον *Internet Explorer* και τον *Safari*. Για τις ανάγκες της συγκεκριμένης πτυχιακής εργασίας προτείνεται η χρήση του Google Chrome.

2.3.2 Ο εξυπηρετητής (server)

Για την υλοποίηση της λειτουργικότητας από την πλευρά του εξυπηρετητή, χρησιμοποιείται μία εφαρμογή λογισμικού η οποία είναι γενικότερα γνωστή ως Εξυπηρετητής δικτύου, HTTP server ή αλλιώς web server.

Ο web server αναλαμβάνει την διαχείριση όλων των αιτήσεων των πελατών για την απόκτηση μιας πληροφορίας. Στη συνέχεια, επεξεργάζεται τις αιτήσεις αυτές και επιστρέφει το κατάλληλο περιεχόμενο, με βάση την αίτηση την οποία εξέδωσε το κάθε πρόγραμμα πελάτη, προς αυτούς. Οι αιτήσεις αυτές αφορούν συνήθως την προσπέλαση μιας σελίδας *HTML* ή κάποιας υπηρεσίας, στοιχεία τα οποία συντηρεί ο εξυπηρετητής στον προσωπικό του αποθηκευτικό χώρο.

Για την επικοινωνία και την απόκριση μηνυμάτων με τις αιτούμενες πληροφορίες, ο web server, όπως και οι clients, χρησιμοποιεί το πρωτόκολλο *HTTP*.

Πολλές εταιρίες λογισμικού έχουν ασχοληθεί με την ανάπτυξη web server με αποτέλεσμα να διατίθεται πληθώρα επιλογών προς εγκατάσταση. Μερικές από τις πιο δημοφιλείς επιλογές για web server αποτελούν οι παρακάτω:

- Apache HTTP server από την εταιρεία Apache Software Foundation
- IIS από την εταιρεία Microsoft
- IBM HTTP server από την εταιρεία IBM
- lighttpd ο οποίος αναπτύχθηκε από τον Jan Kneschke
- nginx από την εταιρεία NGINX.

Για τις ανάγκες της εν λόγω πτυχιακής εργασίας, θα επιλεγεί η χρήση του web server Apache καθώς η επιλογή του προσφέρει τα εξής πλεονεκτήματα:

- Αποτελεί μία από τις παλαιότερες εφαρμογές για web server που αναπτύχθηκαν (συγκεκριμένα από το 1995) η οποία συντηρείται και εξελίσσεται μέχρι και σήμερα προσφέροντας έτσι ένα προϊόν πολυετούς εμπειρίας και γνώσης.
- Η χρήση του είναι δωρεάν καθώς διανέμεται με άδεια δωρεάν χρήσης λογισμικού σε αντίθεση με άλλες υλοποιήσεις (π.χ. IIS).
- Είναι συμβατός με μία μεγάλη γκάμα λειτουργικών συστημάτων μερικά από τα οποία είναι: Unix, FreeBSD, Linux, Microsoft Windows και OS-X
- Υποστηρίζει ένα μεγάλο εύρος γλωσσών προγραμματισμού server-side (π.χ. php, perl, python)
- Διαθέτει μία μεγάλη γκάμα από precompiled modules τα οποία επεκτείνουν την λειτουργικότητά του.

- Υποστηρίζεται από μία πολυπληθή και ενεργή κοινότητα χρηστών και προγραμματιστών, η οποία παρέχει βοήθεια στους νέους χρήστες και ταυτόχρονα εξασφαλίζει την συνεχή ενημέρωση και ανάπτυξη του.
- Αποτελεί μία δοκιμασμένη επιλογή καθώς έχει χρησιμοποιηθεί σε παραπάνω από 100 εκατομμύρια web sites.

2.3.3 HTTP

Το HTTP (Hyper Text Transfer Protocol), όπως αναφέρθηκε παραπάνω, αποτελεί το πρωτόκολλο το οποίο χρησιμοποιείται για την επικοινωνία μεταξύ του client(web-browser) και του server(web server) σε μια διαδικτυακή εφαρμογή. Είναι ένα πρωτόκολλο μεταφοράς υπερκειμένων. Ως υπερκείμενο ορίζεται η τεχνική κατά την οποία, κείμενο το οποίο βρίσκεται σε ηλεκτρονική μορφή έχει την ικανότητα να αναφερθεί σε ένα άλλο κείμενο, γεγονός το οποίο συνδυάζεται συνήθως με άμεση μετάβαση στο αναφερθέν κείμενο. Επειδή η τεχνική αυτή αποτελεί ένα από τα βασικότερα γνωρίσματα της γλώσσας HTML, πολλές φορές το HTTP χαρακτηρίζεται και ως πρωτόκολλο μεταφοράς σελίδων HTML.

Η λειτουργία του χαρακτηρίζεται από τα εξής βήματα:

- Μία σύνδεση προαπαιτείται μεταξύ του πελάτη και του εξυπηρετητή ώστε να είναι δυνατή η επικοινωνία μεταξύ τους.
- Ο πελάτης αποστέλλει ένα μήνυμα προς τον εξυπηρετητή το οποίο περιέχει μία αίτηση για προσπέλαση ενός πόρου, η θέση του οποίου προσδιορίζεται μέσω του μηχανισμού URL.
- Ο εξυπηρετητής αποστέλλει με τη σειρά του ένα μήνυμα προς τον πελάτη το οποίο περιέχει τον ζητούμενο πόρο ή μια ειδοποίηση για την αποτυχία εύρεσης του πόρου.

Αν και αποτελεί το de facto πρωτόκολλο επικοινωνίας μεταξύ web browser και web server, το HTTP διαθέτει ένα σημαντικό μειονέκτημα στον τρόπο λειτουργίας του το οποίο μπορεί να περιγραφεί πλήρως από τον χαρακτηρισμό του ως *stateless*. Με τον όρο *stateless* εννοούμε ότι, ένας εξυπηρετητής ο οποίος λειτουργεί με HTTP επεξεργάζεται τις αιτήσεις που του αποστέλλονται χωρίς να διατηρεί γνώση για τις αιτήσεις οι οποίες προηγήθηκαν, ακόμα και αν αυτές οι αιτήσεις προήλθαν από τον ίδιο πελάτη.

Αυτό δημιουργεί ένα μεγάλο πρόβλημα στις διαδικτυακές εφαρμογές καθώς, υπάρχουν πολλές περιπτώσεις στις οποίες η προηγούμενη κατάσταση της αλληλεπίδρασης ενός χρήστη με το σύστημα πρέπει να είναι διαθέσιμη και σε καμία περίπτωση να μην χάνεται. Το παραπάνω πρόβλημα μπορεί να γίνει εύκολα κατανοητό αν, παραδείγματος χάριν, αναλογιστούμε την πλέον γνώριμη διαδικασία κατά την οποία ένας χρήστης εισέρχεται στο σύστημα (login). Είναι λογικό και

αναμενόμενο από το σύστημα να διατηρεί τα στοιχεία σύνδεσης του χρήστη από τη στιγμή και μετά που αυτός έχει συνδεθεί στο σύστημα και να μην αναγκάζεται να εισάγει τα διαπιστευτήριά του ξανά και ξανά. Δυστυχώς, η παραπάνω ανάγκη είναι κάτι το οποίο το πρωτόκολλο HTTP δεν μπορεί να αντιμετωπίσει.

Για την αντιμετώπιση του παραπάνω προβλήματος έχουν δημιουργηθεί ορισμένες τεχνικές διατήρησης τέτοιων πληροφοριών μία από τις οποίες είναι αυτή της δημιουργίας συνεδρίων (sessions). Περισσότερες πληροφορίες για τις συνεδρίες θα υπάρξουν στο τρίτο κεφάλαιο στην ενότητα υλοποίησης του συστήματος.

2.4 Η αρχιτεκτονική client-server τριών στρωμάτων (3 tier)

Το πρότυπο client-server το οποίο περιγράφηκε μέχρι αυτή τη στιγμή χρησιμοποιεί δύο μόνο μέρη λογισμικού για να περιγράψει πλήρως την λειτουργικότητα την οποία υλοποιεί. Για το λόγο αυτό, το μοντέλο ονομάζεται αρχιτεκτονική δύο στρωμάτων (2-tier architecture) .

Η αρχιτεκτονική αυτή υπήρξε για αρκετό καιρό ικανοποιητική στις απαιτήσεις και τις ανάγκες των δικτυακών εφαρμογών που αναπτύσσονταν καθώς, το περιεχόμενό τους ήταν σχετικά στατικό με λίγες πληροφορίες ενώ παράλληλα ο αριθμός των αιτήσεων των χρηστών προς τους εξυπηρετητές ήταν σχετικά χαμηλός και ανεκτός από την επεξεργαστική ισχύ των εξυπηρετητών.

Στα χρόνια που ακολούθησαν όμως, τα παραπάνω δεδομένα άλλαξαν. Η ανάπτυξη και η διάδοση του internet συνετέλεσε στη δραματική αύξηση των χρηστών οι οποίοι είχαν πλέον την δυνατότητα πρόσβασης και αίτησης δεδομένων προς τις διαδικτυακές εφαρμογές. Παράλληλα οι εφαρμογές, με την σειρά τους, εμπλουτίστηκαν με πολύπλοκο, ογκώδες, και πιο δυναμικό περιεχόμενο, ώστε να είναι σε θέση να ανταποκρίνονται σε μεγαλύτερες απαιτήσεις χρηστών και κυρίως επιχειρήσεων, οι οποίες αποφάσιζαν να επενδύσουν στην χρήση του δημοφιλούς διαδικτύου και των υπηρεσιών που αυτό είχε να προσφέρει.

Δεδομένων λοιπόν όλων των παραπάνω, οι εξυπηρετητές αντιμετώπιζαν πλέον δυσκολία στο να διαχειριστούν και να επεξεργαστούν αυτό τον νέο όγκο δεδομένων καθώς και την υπερπληθώρα αιτήσεων από τους υπεράριθμους πλέον χρήστες, καθιστώντας έτσι την αρχιτεκτονική 2-tier ανήμπορη να αντεπεξέλθει στα νέα δεδομένα.

Για την αντιμετώπιση λοιπόν όλων των προαναφερθέντων, έγινε μια προσπάθεια προσθήκης νέων μερών λογισμικού στην αρχιτεκτονική των δύο στρωμάτων τα οποία θα διαχειρίζονται τον επιπρόσθετο φόρτο εργασίας. Το αποτέλεσμα αυτής της προσπάθειας ήταν η δημιουργία της 3-tier αρχιτεκτονικής πελάτη-εξυπηρετητή.

Σύμφωνα με την νέα αυτή αρχιτεκτονική, ορίζεται η προσθήκη ενός στρώματος λειτουργικότητας για την διαχείριση και την εξυπηρέτηση αιτήσεων των πολύπλοκων δεδομένων, ανακουφίζοντας έτσι τον web server από τον βαρύ αυτό φόρτο εργασίας.

Η κεντρική λογική πίσω από την αρχιτεκτονική αυτή είναι ότι, πλέον ο web server εξυπηρετεί τις εισερχόμενες διαδικτυακές αιτήσεις υλοποιώντας την επιχειρησιακή λογική του συστήματος, όπως και πριν, αλλά με την διαφορά ότι πλέον οποιαδήποτε αίτηση η οποία αφορά διαχείριση δεδομένων όπως παραδείγματος χάριν, αναζήτηση σε αρχεία δεδομένων μεγάλου όγκου καθώς και ενημέρωση και αποθήκευση αυτών, αποστέλλεται πλέον στο στρώμα δεδομένων το οποίο επεξεργάζεται αυτές τις αιτήσεις και επιστρέφει έτοιμο το επιθυμητό αποτέλεσμα στον web server ο οποίος με την σειρά του επιστρέφει το αποτέλεσμα αυτό στο πρόγραμμα πελάτη (web browser).

Επειδή η λογική του στρώματος δεδομένων είναι παρόμοια με αυτή του web server, το στρώμα είναι επίσης γνωστό και ως εξυπηρετητής βάσεων δεδομένων (database server). Περισσότερες λεπτομέρειες για τον εξυπηρετητή βάσεων δεδομένων θα αναφερθούν στην επόμενη ενότητα.

2.5 Ο Database Server

Ο database server, όπως βέβαια και ο web server, αποτελεί ένα πρόγραμμα λογισμικού ή πολλές φορές συλλογή προγραμμάτων λογισμικού, το οποίο έχει ως σκοπό την διαχείριση και την εξυπηρέτηση των αιτήσεων οι οποίες σχετίζονται με τα δεδομένα της διαδικτυακής εφαρμογής. Ταυτόχρονα, φροντίζει για την αποθήκευση και την συντήρηση των δεδομένων αυτών ώστε να είναι διαθέσιμα οποιαδήποτε χρονική στιγμή κριθεί απαραίτητη η προσπέλασή τους.

Το γεγονός ότι ο database server αποτελεί μια ξεχωριστή οντότητα λογισμικού, παρέχει πλεονεκτήματα στην λειτουργικότητα του συστήματος της διαδικτυακής εφαρμογής. Αναλυτικότερα:

- Παρέχεται η δυνατότητα εγκατάστασης και της λειτουργίας του σε ένα ξεχωριστό υπολογιστικό σύστημα από εκείνο του web server προσφέροντας έτσι καλύτερες επιδόσεις και μεγαλύτερη αποτελεσματικότητα.
- Η εφαρμογή του web server δε χρειάζεται να διαθέτει γνώση για τον τρόπο αποθήκευσης και προσπέλασης των δεδομένων καθώς πλέον γι αυτό είναι υπεύθυνος ο database server.
- Δίνεται η δυνατότητα αναβάθμισης ή και ακόμα αλλαγής του λογισμικού το ενός εξυπηρετητή χωρίς αυτό να επηρεάζει την συμπεριφορά και την λειτουργικότητα του άλλου.

Στις επόμενες ενότητες θα πραγματοποιηθεί μια εκτενής αναφορά στα συστατικά στοιχεία από τα οποία αποτελείται ένας εξυπηρετητής βάσεων δεδομένων.

2.6 Συστήματα διαχείρισης βάσεων δεδομένων (DBMS)

Τα συστήματα διαχείρισης βάσεων δεδομένων ή αλλιώς DBMS, είναι λογισμικό (ή συλλογή προγραμμάτων λογισμικού) το οποίο επιτρέπει την δημιουργία καθώς και την συντήρηση μίας ή περισσότερων βάσεων δεδομένων ενώ ταυτόχρονα, ο σχεδιασμός του είναι διαμορφωμένος κατάλληλα ώστε να διευκολύνει τη χρήση και τη διαχείριση μεγάλου όγκου πληροφοριών.

Ως βάση δεδομένων θεωρείται οποιαδήποτε συλλογή δεδομένων ο συνδυασμός των οποίων εμπεριέχει κάποια συλλογιστική συγγένεια και συντελεί στην περιγραφή μία συγκεκριμένης δραστηριότητας. Για παράδειγμα, η συλλογή της πληροφορίας για το σύνολο των φοιτητών ενός ακαδημαϊκού ιδρύματος και του συνόλου των μαθημάτων στα οποία ο κάθε φοιτητής είναι εγγεγραμμένος, αποτελούν περιπτώσεις βάσεων δεδομένων[A2].

Σε αυτό το σημείο αξίζει να αναφερθεί ότι η χρήση ενός συστήματος διαχείρισης βάσεων δεδομένων δεν αποτελεί μοναδική επιλογή καθώς, παρέχεται πάντα η επιλογή χρήσης μίας γλώσσας προγραμματισμού μέσω της οποίας μπορούμε να παράγουμε κώδικα λογισμικού παρόμοιας λειτουργικότητας με αυτά των συστημάτων DBMS. Παρόλα αυτά, για την υλοποίηση της συγκεκριμένης πτυχιακής εργασίας, επιλέγουμε την χρησιμοποίηση DBMS για την διαχείριση των δεδομένων καθώς η επιλογή αυτή διακρίνεται από τα εξής προτερήματα[A2]:

- Το DBMS χρησιμοποιεί μία αφηρημένη προσέγγιση οπτικής των δεδομένων με τέτοιο τρόπο ώστε να ανεξαρτητοποιεί εξωτερικές εφαρμογές λογισμικού, καθώς και τους προγραμματιστές αυτών των εφαρμογών, από την ανάγκη για γνώση του εσωτερικού (φυσικού) τρόπου με τα οποία τα δεδομένα δομούνται.
- Τα συστήματα DBMS διαθέτουν και χρησιμοποιούν πληθώρα τεχνικών ώστε η καταχώρηση καθώς και η ανάκτηση των δεδομένων να πραγματοποιείται με παραγωγικό τρόπο. Αυτές οι τεχνικές αποδεικνύονται ιδιαίτερα χρήσιμες σε περιπτώσεις όπου τα δεδομένα βρίσκονται αποθηκευμένα σε εξωτερικές περιφερειακές συσκευές.
- Στην διαδικασία ενημέρωσης ή αποθήκευσης των δεδομένων, τα συστήματα DBMS διαθέτουν εσωτερικούς μηχανισμούς με τους οποίους μπορούν να επιβάλλουν περιορισμούς ώστε να επιτυγχάνεται η ακεραιότητα των δεδομένων. Για παράδειγμα, στην καταχώρηση της βαθμολογίας ενός φοιτητή πρέπει να εφαρμόζεται ο περιορισμός ότι η βαθμολογία του δεν θα πρέπει να ξεπερνάει το μέγιστο όριο (π.χ. 10) αλλά ταυτόχρονα θα πρέπει να είναι μη-αρνητικός αριθμός (≥ 0).
- Τα συστήματα DBMS παρέχουν επίσης την δυνατότητα ορισμού δικαιωμάτων πρόσβασης με βάση τα οποία μπορεί να οριστεί, από το σύνολο της πληροφορίας που είναι αποθηκευμένο στη βάση, ποιο ποσοστό θα είναι διαθέσιμο στον εκάστοτε χρήστη ή εφαρμογή.

- Παράλληλα με τη χρήση συστημάτων DBMS επιτυγχάνεται καλύτερη επίδοση του συστήματος καθώς, η διαχείριση γίνεται με τέτοιο τρόπο ώστε τα δεδομένα να καταχωρούνται μόνο μία φορά χωρίς να δημιουργούνται αχρείαστοι πλεονασμοί.
- Τα συστήματα DBMS έχουν την δυνατότητα να ορίσουν ταυτόχρονη πρόσβαση σε δεδομένα σε παραπάνω από έναν χρήστες με τέτοιο τρόπο ώστε ο κάθε χρήστης να έχει την αίσθηση ότι είναι ο μόνος ο οποίος προσπελαύνει τα εν λόγω δεδομένα την συγκεκριμένη χρονική στιγμή.
- Επιπρόσθετα τα DBMS προσφέρουν ένα επίπεδο ασφαλούς λειτουργίας και αίσθηση αξιοπιστίας καθώς, διαθέτουν όλες τις απαραίτητες λειτουργίες επαναφοράς ενός σύνολου πληροφορίας ή ακόμα και ολόκληρου του συστήματος σε περίπτωση απώλειας της πληροφορίας αυτής ή βλάβης του συστήματος αντίστοιχα.
- Τέλος τα συστήματα DBMS προσφέρουν αφηρημένες, υψηλού επιπέδου διεπαφές προγραμματισμού (API's), μέσω των οποίων είναι δυνατή η επικοινωνία με τη βάση δεδομένων καθώς και η πρόσβαση στα δεδομένα της από ετερογενείς γλώσσες προγραμματισμού (π.χ. PHP).

Έχοντας λοιπόν καταλήξει στην χρήση του DBMS, επόμενο βήμα αποτελεί η επιλογή του κατάλληλου προγράμματος το οποίο υλοποιεί ένα τέτοιο σύστημα.

Η αγορά προσφέρει έναν μεγάλο πλήθος επιλογών για εφαρμογές λογισμικού DBMS εκ των οποίων τα πιο δημοφιλή είναι τα παρακάτω:

- Oracle Database
- Microsoft SQL Server
- MySQL
- IBM DB2
- SAP Sybase Adaptive Server Enterprise
- Teradata

Για τις ανάγκες της εν λόγω πτυχιακής εργασίας, θα γίνει η χρήση του MySQL καθώς η επιλογή του προσφέρει τα εξής πλεονεκτήματα:

- Χρησιμοποιείται από μεγάλες εταιρίες λογισμικού όπως είναι παραδείγματος χάριν η Google
- Είναι open source το οποίο σημαίνει ότι ο κώδικας διατίθεται ελεύθερα στην κοινότητα λογισμικού για περαιτέρω διόρθωση και ανάπτυξη
- Διατίθεται δωρεάν
- Υποστηρίζει ένα μεγάλο εύρος από πλατφόρμες (Linux, Microsoft Windows, Mac OS-X, Solaris, OS/2)

- Διαθέτει ένα μεγάλο αριθμό από API's με τις οποίες υποστηρίζει πληθώρα γλωσσών προγραμματισμού (PHP, Python, Java, C, C++, Perl)

2.7 SQL

Για να καταστεί δυνατή η λειτουργικότητα των συστημάτων DBMS η οποία αναφέρθηκε παραπάνω, είναι αναγκαία η χρήση μιας δομημένης γλώσσας μέσω της οποίας να είναι δυνατό να οριστεί ένα σύνολο ενεργειών για την διαχείριση των δεδομένων που αποθηκεύονται. Η πλειοψηφία των συστημάτων DBMS (μεταξύ άλλων και το MySQL) χρησιμοποιεί για το σκοπό αυτό τη γλώσσα SQL.

Η SQL (Structured Query Language), είναι μια γλώσσα προγραμματισμού ειδικού σκοπού η οποία χρησιμοποιείται για τον ορισμό, τον σχεδιασμό καθώς την διαχείριση δεδομένων τα οποία καταχωρούνται σε ένα σχεσιακό σύστημα διαχείρισης βάσεων δεδομένων (RDBMS), ενώ παράλληλα είναι υπεύθυνη για την εξυπηρέτηση των αιτημάτων τα οποία διατυπώνονται και αφορούν τα δεδομένα αυτά. Επιπρόσθετα, έχει την ικανότητα να ορίζει το εύρος των δικαιωμάτων πρόσβασης για κάθε χρήστη ο οποίος επιθυμεί να προσπελάσει την βάση.

Με τον όρο RDBMS αναφερόμαστε σε όλα εκείνα τα συστήματα διαχείρισης δεδομένων των οποίων τα δεδομένα δομούνται με βάση το *σχεσιακό μοντέλο*. Σύμφωνα με αυτό, σε μια βάση δεδομένων, όλη η αποθηκευμένη πληροφορία αναπαριστάται με την μορφή πινάκων, οι οποίοι αποτελούνται από γραμμές και στήλες, μεταξύ των οποίων δημιουργούνται συσχετίσεις οι οποίες ορίζουν μια κατάσταση αλληλεπίδρασης ανάμεσά τους. Οι συσχετίσεις αυτές ονομάζονται *σχέσεις*.

Η SQL χρησιμοποιεί ένα σύνολο εντολών μέσω των οποίων είναι δυνατή η διαχείριση των δεδομένων από τα συστήματα DBMS. Οι εντολές αυτές αναφέρονται παρακάτω:

- *CREATE*: Εντολή για την δημιουργία μίας νέας βάσης ή ενός νέου πίνακα μέσα σε μια ήδη υπάρχουσα βάση δεδομένων
- *DROP*: Εντολή για την διαγραφή μίας βάσης δεδομένων ή ενός νέου πίνακα ο οποίος ανήκει σε μια βάση δεδομένων
- *SELECT*: Εντολή η οποία χρησιμοποιείται για την αίτηση ερωτημάτων στην βάση δεδομένων. Η εντολή αυτή επιστρέφει καμία, μία ή περισσότερες εγγραφές από έναν ή περισσότερους πίνακες με βάση τα κριτήρια αναζήτησης του εκάστοτε ερωτήματος.
- *INSERT*: Εντολή για την εισαγωγή μίας ή περισσότερων γραμμών (εγγραφών) σ' ένα πίνακα.
- *DELETE*: Εντολή για την διαγραφή μίας ή περισσότερων γραμμών (εγγραφών) σ' ένα πίνακα.

- **UPDATE:** Εντολή για την ενημέρωση των δεδομένων που περιέχονται σε μία ή περισσότερες γραμμές ενός πίνακα.
- **GRANT:** Εντολή για τον ορισμό δικαιωμάτων πρόσβασης ενός χρήστη πάνω σε μια βάση δεδομένων.
- **REVOKE:** Εντολή για την απαλοιφή δικαιωμάτων πρόσβασης ενός χρήστη πάνω σε μια βάση δεδομένων.

2.8 Server Side Scripting και PHP

Στην ενότητα αυτή θα πραγματοποιηθεί μία αναφορά στην τεχνική του *server side scripting* καθώς και στον λόγο για τον οποίο χρησιμοποιείται. Επιπρόσθετα θα γίνει μια μικρή αναφορά σε γλώσσες προγραμματισμού οι οποίες χρησιμοποιούνται για *server side scripting* κατά την οποία θα αποδοθούν οι λόγοι για τους οποίους επιλέγουμε την χρήση της PHP έναντι των άλλων γλωσσών.

2.8.1 Server Side Scripting

Το *server side scripting*, όπως και το στρώμα βάσεων δεδομένων το οποίο αναφέρθηκε παραπάνω, αποτελεί αποτέλεσμα της ανάγκης για δημιουργία πιο ευέλικτων και δυναμικών διαδικτυακών εφαρμογών, ώστε να μπορούν να ανταποκρίνονται σ' ένα πιο απαιτητικό περιβάλλον λειτουργικότητας, το οποίο επιβάλλεται από τις νεότερες και εξελιγμένες επιχειρησιακές απαιτήσεις των οργανισμών για τους οποίους δημιουργούνται οι εφαρμογές αυτές.

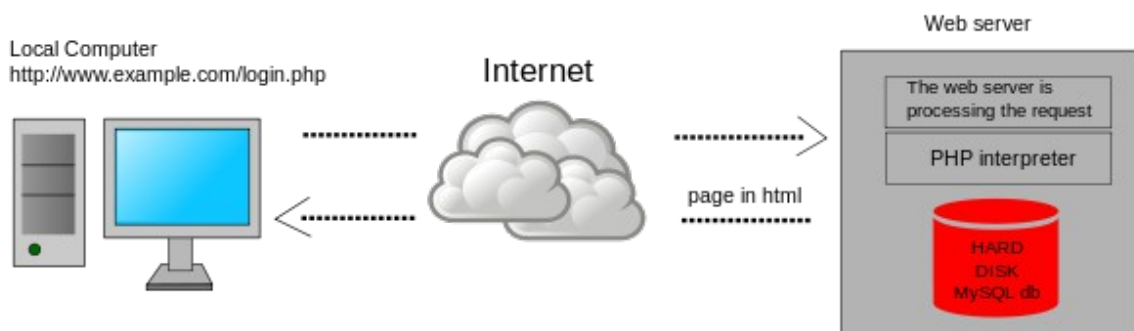
Αποτελεί μία τεχνική διαδικτυακού προγραμματισμού κατά την οποία, μέσω μιας γλώσσας προγραμματισμού, δημιουργούνται στοιχεία κώδικα (είναι γνωστά και ως *scripts*) τα οποία, είτε αποθηκεύονται σε ξεχωριστά αρχεία είτε ενσωματώνονται σε σελίδες HTML (ισχύει για μερικούς αλλά όχι για όλους τους τύπους *script*) και υλοποιούν μια συγκεκριμένη λειτουργικότητα εκτελώντας μία σειρά εντολών. Τα αρχεία αυτά, αποθηκεύονται στην πλευρά του *server* και είναι προσπελάσιμα μέσω του μηχανισμού URL, όπως δηλαδή πραγματοποιείται και η εύρεση μιας απλής σελίδας HTML.

Ο τρόπος με τον οποίο λειτουργεί το *server side scripting* μπορεί να γίνει κατανοητός από τα παρακάτω βήματα:

- Ο χρήστης, μέσω ενός *web browser*, αιτείται την προσπέλαση ενός αρχείου *script* το οποίο βρίσκεται στον αποθηκευτικό χώρο του *web server*. Η ακριβής θέση στην οποία βρίσκεται το αρχείο, διατυπώνεται από την τιμή του URL που εισάγει ο χρήστης στην περιοχή διεύθυνσης (*address bar*) του *web browser*

- Ο web browser επεξεργάζεται το αίτημα και αναγνωρίζει ότι πρόκειται για προσπέλαση ενός αρχείου script.
- Στην συνέχεια χρησιμοποιώντας το κατάλληλο module μεταφραστή για την γλώσσα στην οποία είναι γραμμένο το script, το εκτελεί και λαμβάνει τα δεδομένα που αυτό επιστρέφει. Στην περίπτωση που το script χρειαστεί πρόσβαση στη βάση δεδομένων, αποστέλλει ένα αίτημα στο DBMS το οποίο τη διαχειρίζεται ώστε να λάβει τα δεδομένα τα οποία χρειάζεται.
- Τέλος ο web server δημιουργεί μία σελίδα HTML η οποία περιέχει τα δεδομένα που προέκυψαν από την εκτέλεση του script και στην συνέχεια την επιστρέφει στον web browser.

Μία οπτική περιγραφή των προαναφερθέντων βημάτων μπορεί να δοθεί από την παρακάτω εικόνα:



Εικόνα 2: Server Side Scripting

Αυτό που αξίζει να σημειωθεί από την παραπάνω διαδικασία είναι ότι ο χρήστης, ενώ αποζητάει την προσπέλαση ενός αρχείου script, αντ' αυτού λαμβάνει ως αποτέλεσμα τα δεδομένα που προκύπτουν από την εκτέλεση του, στη μορφή μιας σελίδας HTML η οποία απεικονίζεται στην οθόνη του web browser.

Το παραπάνω αποτελεί ένα βασικό πλεονέκτημα που προσφέρει η τεχνική server side scripting καθώς, η δυνατότητα απόκρυψης του κώδικα των scripts από τους χρήστες, περιορίζει την ικανότητα τους να έχουν πρόσβαση σε πληροφορίες που αφορούν τον τρόπο λειτουργίας του συστήματος, προσφέροντας έτσι ένα επίπεδο αξιοπιστίας και ασφάλειας. Αυτό έρχεται σε αντίθεση με την τεχνική client side scripting στην οποία, η αίτηση μιας σελίδας HTML η οποία περιέχει κώδικα υλοποιημένο σε γλώσσα προγραμματισμού client side (π.χ. Javascript) θα επιστρέφει, εκτός τα στοιχεία της HTML σελίδας, όλο το περιεχόμενο κώδικα που αυτή περιέχει.

Επιπρόσθετα, προσφέρει ασφάλεια και στην περίπτωση πρόσβασης σε βάσεις δεδομένων καθώς η γλώσσα προγραμματισμού λειτουργεί ως ενδιάμεσος ανάμεσα στον χρήστη και τη βάση, αποκρύπτοντας έτσι ευαίσθητα δεδομένα ή δεδομένα στα οποία οι χρήστες δεν έχουν δικαίωμα πρόσβασης.

2.8.2 PHP

Η PHP (PHP Hypertext Preprocessor) ενώ μπορεί να θεωρηθεί ως μία γλώσσα προγραμματισμού γενικού σκοπού, πρακτικά χρησιμοποιείται κατά κόρον στον σχεδιασμό διαδικτυακών εφαρμογών για τη συγγραφή scripts. Η ανάπτυξη της ξεκίνησε το 1994 και έκτοτε εξελίσσεται συνεχώς με νέες εκδόσεις. Η τρέχουσα έκδοση είναι η 5.5.

Χρησιμοποιεί ένα σχετικά χαλαρό τρόπο ορισμού μεταβλητών καθώς, παρέχει την δυνατότητα να αποθηκεύσουμε περιεχόμενο σε μία μεταβλητή χωρίς να ορίσουμε κάποιο τύπο για αυτήν, προσφέροντας έτσι μεγαλύτερη ευελιξία στην συγγραφή κώδικα.

Ένα ακόμη χρήσιμο χαρακτηριστικό της γλώσσας είναι, η δυνατότητα χρήσης μεταβλητών γενικού σκοπού των οποίων το περιεχόμενο μπορεί να διαβαστεί από όλα τα αρχεία κώδικα (scripts) τα οποία δημιουργούνται. Οι μεταβλητές αυτές ονομάζονται *superglobal* μεταβλητές. Ενδεικτικά αναφέρονται οι παρακάτω:

- `$_GET`: Περιέχει μεταβλητές οι οποίες αποστέλλονται σε ένα αρχείο script μέσω της μεθόδου GET
- `$_POST`: Περιέχει μεταβλητές οι οποίες αποστέλλονται σε ένα αρχείο script μέσω της μεθόδου POST
- `$_SERVER`: Περιέχει πληροφορίες για θέσεις αρχείων και scripts σε μια διαδρομή καταλόγου
- `$_SESSION`: Περιέχει τις μεταβλητές που δημιουργήθηκαν κατά τη διάρκεια μιας συνεδρίας (Περισσότερες πληροφορίες για τις συνεδρίες θα δοθούν στο κεφάλαιο 3)

Μολονότι υπάρχει πληθώρα επιλογών γλωσσών για server-side scripting μεταξύ των οποίων οι: ASP, Ruby, Perl, και Python, για την υλοποίηση του συστήματος θα γίνει η χρήση της PHP για τους εξής λόγους:

- Παρέχεται δωρεάν
- Διαθέτει ενεργή κοινότητα χρηστών και forum συζητήσεων που βοηθούν τον νέο χρήστη να εξοικειωθεί με την γλώσσα
- Είναι cross-platform

- Συνεργάζεται πολύ καλά με τα components του APACHE server και του MySQL, γεγονός το οποίο επαληθεύεται από την πληθώρα των web εφαρμογών και ιστότοπων που έχουν δημιουργηθεί με τον συνδυασμό αυτό
- Είναι μια αρκετά δημοφιλής γλώσσα προγραμματισμού γεγονός το οποίο έχει οδηγήσει στην ανάπτυξη βιβλιοθηκών και framework τα οποία επεκτείνουν την λειτουργικότητα της.
- Έχει χρησιμοποιηθεί για την ανάπτυξη μεγάλων και σύνθετων συστημάτων όπως π.χ. το Facebook.

ΚΕΦΑΛΑΙΟ 3: Η Υλοποίηση του Συστήματος

ΕΙΣΑΓΩΓΗ

Στο κεφάλαιο αυτό θα δοθεί ένα σύνολο οδηγιών εγκατάστασης των εργαλείων λογισμικού τα οποία επιλέχθηκαν στο κεφάλαιο δύο. Στη συνέχεια, θα πραγματοποιηθεί μία αναλυτική περιγραφή των βημάτων που ακολουθήθηκαν για την δημιουργία του συστήματος με τη χρήση των εργαλείων που εγκαταστάθηκαν πρωτίτερα. Τέλος, θα δημιουργηθεί μία μικρή ενότητα στην οποία, θα περιγραφούν προβλήματα τα οποία προέκυψαν στην διαδικασία ανάπτυξης του συστήματος καθώς και τον τρόπο με τον οποίο αυτά επιλύθηκαν.

3.1 Το λειτουργικό σύστημα και η εγκατάσταση των εργαλείων λογισμικού

Στην ενότητα αυτή, θα περιγραφεί η διαδικασία εγκατάστασης αλλά και προετοιμασίας όλων των προαπαιτούμενων εργαλείων λογισμικού, τα οποία στη συνέχεια θα αποτελέσουν τη βάση πάνω στην οποία θα δομηθεί και θα αναπτυχθεί το σύστημα LMS.

3.1.1 Ο εξυπηρετητής και το λειτουργικό σύστημα

Η διαδικασία ξεκινάει με την επιλογή του υπολογιστικού συστήματος που θα διατελέσει χρέη εξυπηρετητή (server). Πάνω σε αυτόν θα γίνει, στη συνέχεια, η εγκατάσταση των εργαλείων λογισμικού και των υπηρεσιών με τα οποία θα είναι δυνατή η διαχείριση των αιτήσεων των χρηστών προς το σύστημα LMS.

Για τις ανάγκες της πτυχιακής εργασίας, με την βοήθεια του προσωπικού του τμήματος Πληροφορικής του Α.Τ.Ε.Ι.Θ και την χρήση εργαλείων λογισμικού virtual machine, παρέχεται ένας ιδεατός εξυπηρετητής (virtual server) ο οποίος είναι προσβάσιμος μέσω της διεύθυνσης IP 192.168.16.58. Επόμενο βήμα αποτελεί, στη συνέχεια, η εγκατάσταση του λειτουργικού συστήματος του εξυπηρετητή. Επειδή, όπως αναφέρθηκε και προηγουμένως, το εν λόγω σύστημα αποτελεί μία υλοποίηση η οποία θα αναπτυχθεί σε περιβάλλον UNIX, επιλέγεται ως λειτουργικό σύστημα για τον server το Linux και πιο συγκεκριμένα η διανομή Debian 6.0.

3.1.2 Σύνδεση με τον εξυπηρετητή.

Το επόμενο βήμα της διαδικασίας απαιτεί, την σύνδεση στον εξυπηρετητή ώστε να πραγματοποιηθεί η εγκατάσταση των προαναφερθέντων εργαλείων λογισμικού.

Η σύνδεση μπορεί επιτευχθεί με δύο τρόπους:

- Μέσω ηλεκτρονικού υπολογιστή που ανήκει στο δίκτυο υπολογιστών του τμήματος Πληροφορικής
- Απομακρυσμένα με χρήση υπολογιστή εκτός των εγκαταστάσεων του τμήματος

Με την πρώτη επιλογή η σύνδεση μπορεί να επιτευχθεί χωρίς κάποιο ιδιαίτερο πρόβλημα καθώς, ο εξυπηρετητής ανήκει και αυτός με την σειρά του σε αυτό το δίκτυο οπότε και είναι προσβάσιμος από οποιοδήποτε μηχάνημα ανήκει στο ίδιο δίκτυο με αυτόν και δεν έχει περιορισμό πρόσβασης από μηχανισμούς ασφαλείας (π.χ. Firewall).

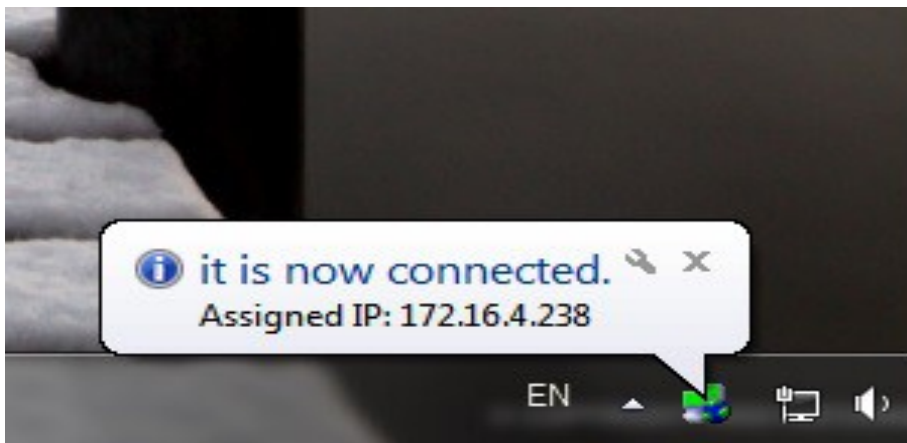
Αντίθετα, η απομακρυσμένη σύνδεση δημιουργεί ορισμένες δυσκολίες καθώς ο εξυπηρετητής στον οποίο απαιτείται πρόσβαση, χρησιμοποιεί μία ιδιωτική διεύθυνση IP το οποίο συνεπάγεται ότι, η πρόσβαση σε αυτό τον εξυπηρετητή μπορεί να πραγματοποιηθεί μόνο μέσω του δικτύου υπολογιστών του τμήματος Πληροφορικής.

Για να αντιμετωπιστεί το παραπάνω πρόβλημα, θα γίνει χρήση λογισμικού τύπου VPN (Virtual Private Network) το οποίο θα παρέχει την δυνατότητα σε υπολογιστές οι οποίοι δεν ανήκουν στο ιδιωτικό δίκτυο του τμήματος, να συνδέονται σε αυτό μέσω δημόσιων δικτύων όπως είναι το Διαδίκτυο. Οι υπηρεσίες του τμήματος Πληροφορικής Α.Τ.Ε.Ι.Θ, υποστηρίζουν την λειτουργία VPN για την οποία οδηγίες χρήσης παρέχονται στον ιστότοπο <http://hydra.it.teithe.gr/vpn/>. Οι οδηγίες επιπρόσθετα προσφέρουν και ένα σύνδεσμο (link) για την απόκτηση της

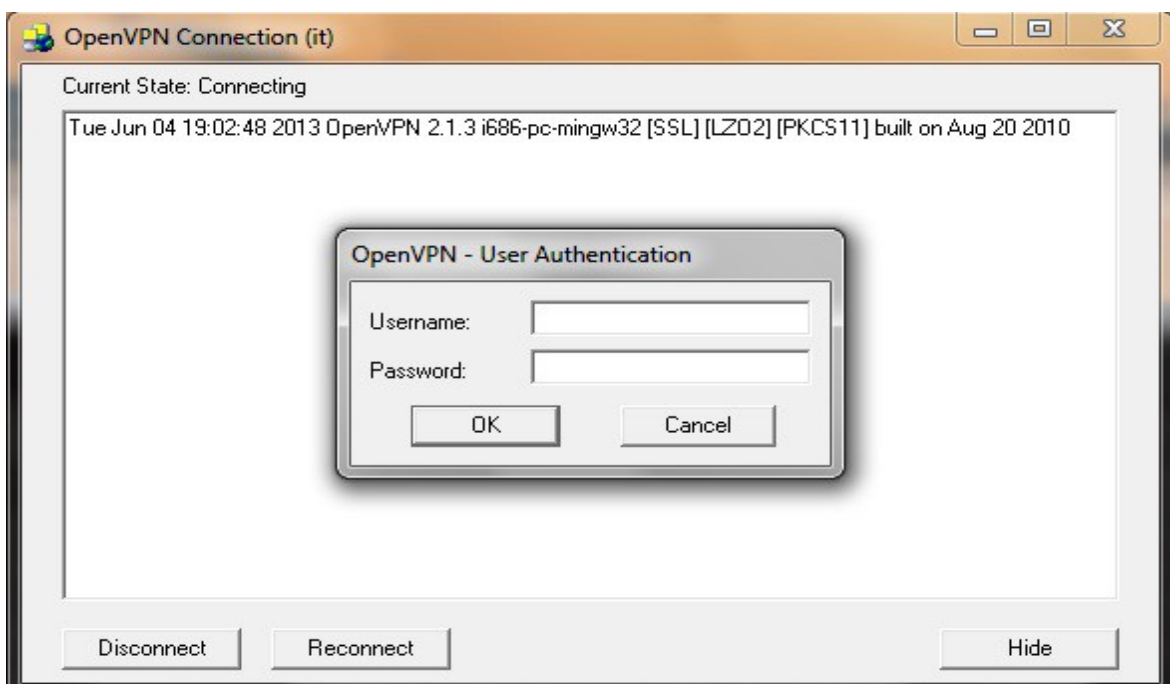
εφαρμογής OpenVPN η οποία είναι προρυθμισμένη κατάλληλα για την επικοινωνία με το τμήμα Πληροφορικής. Η εφαρμογή διατίθεται για λειτουργικά συστήματα Windows και Linux. Επιλέγεται για εγκατάσταση η έκδοση της εφαρμογής για Windows μια και ο απομακρυσμένος υπολογιστής χρησιμοποιεί το εν λόγω λειτουργικό σύστημα. Έχοντας λοιπόν εγκαταστήσει το OpenVPN ακολουθώντας προσεκτικά της οδηγίες, στην συνέχεια εκτελούμε το πρόγραμμα φροντίζοντας πάντα να έχουμε αποκτήσει τα κατάλληλα δικαιώματα διαχειριστή (run as administrator), ώστε να εκτελέσει σωστά όλες τις απαραίτητες ενέργειες που χρειάζονται για την σύνδεση με τον εξυπηρετητή.

Στο παράθυρο User Authentication ο χρήστης συμπληρώνει τα απαραίτητα στοιχεία σύνδεσης τα οποία διαθέτει κάθε άτομο που ανήκει στην ακαδημαϊκή κοινότητα του τμήματος Πληροφορικής και είναι μοναδικά για τον κάθε ένα (Εικόνα 4).

Αν η σύνδεση ολοκληρωθεί με επιτυχία τότε στο δεξί κάτω άκρο της επιφάνειας εργασίας θα εμφανιστεί ένα μήνυμα επιτυχούς σύνδεσης σαν το παρακάτω.



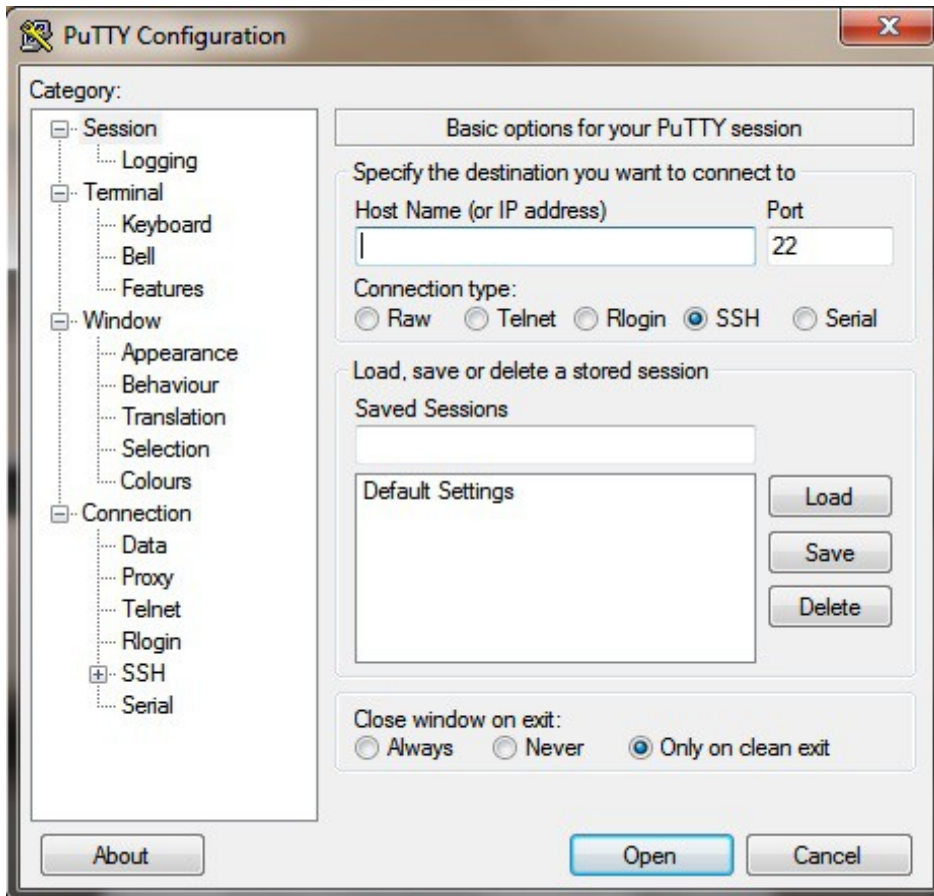
Εικόνα 3: Επιτυχία σύνδεσης μέσω VPN



Εικόνα 4: Interface σύνδεσης, μέσω VPN

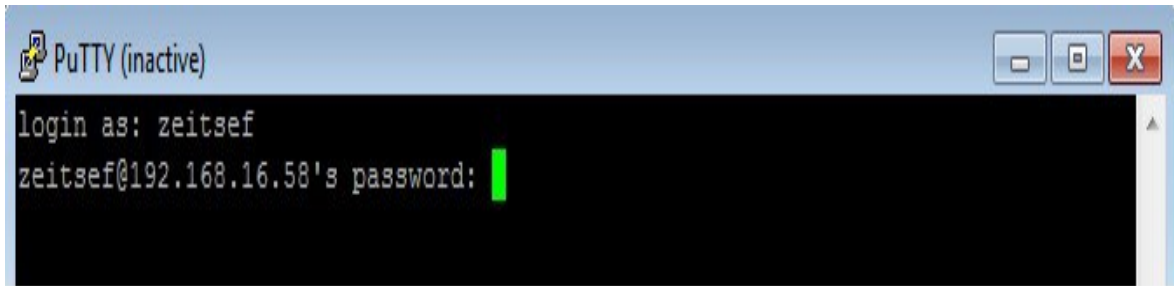
Με την ολοκλήρωση όλων των παραπάνω, το επόμενο βήμα είναι η επιλογή της εφαρμογής με την οποία θα πραγματοποιηθεί η σύνδεση στον εξυπηρετητή. Σε αντίθεση με την εφαρμογή VPN ο χρήστης, σε αυτό το βήμα, μπορεί να επιλέξει όποια εφαρμογή επιθυμεί για την σύνδεση με τον εξυπηρετητή. Στα πλαίσια της πτυχιακής εργασίας θα χρησιμοποιηθεί η εφαρμογή PuTTY για την εν λόγω σύνδεση.

Η εκτέλεση του προγράμματος θα εμφανίσει το παρακάτω παράθυρο.



Εικόνα 5: Σύνδεση μέσω προγράμματος Putty

Στο πεδίο Host Name (or IP address) συμπληρώνεται η διεύθυνση IP του εξυπηρετητή στον οποίο θα πραγματοποιηθεί η σύνδεση.



Εικόνα 6: Ταυτοποίηση χρήστη μέσω Putty

Αμέσως μετά την ολοκλήρωση της σύνδεσης, εμφανίζεται το prompt το οποίο αντιπροσωπεύει την γραμμή εντολών του εξυπηρετητή που έχουμε συνδεθεί, στο οποίο συμπληρώνουμε τα διαπιστευτήρια επιπέδου χρήστη του εξυπηρετητή.

Στη συνέχεια, για την εκτέλεση των υπόλοιπων εντολών που χρειάζονται, είναι απαραίτητη η απόκτηση των δικαιωμάτων διαχειριστή. Αυτό επιτυγχάνεται με την εκτέλεση της παρακάτω εντολής:

```
$ su -
```

και συμπληρώνοντας στη συνέχεια τα διαπιστευτήρια διαχειριστή.

Μετά από την ενέργεια αυτή, κρίνεται πρακτικό να δημιουργηθεί ένας νέος κατάλογος στη διαδρομή `/var/www/` στον οποίο θα αποθηκευτούν το σύνολο των αρχείων `.php` που θα υλοποιήσουν την λειτουργικότητα του συστήματος LMS. Στη γραμμή εντολών, εισάγουμε:

```
mkdir /var/www/teilms.
```

Όπου `teilms` είναι το όνομα καταλόγου το οποίο επιλέχθηκε. Μία επιπρόσθετη πληροφορία που αξίζει να σημειωθεί είναι ότι μέσω τη διαδρομής καταλόγου `/var/www/`, ορίζεται η ιεραρχία των καταλόγων στην οποία θα αποθηκεύονται όλα τα δεδομένα που αφορούν στην λειτουργία του συστήματος LMS. Οι καταλόγοι αυτοί είναι προσπελάσιμοι μέσω του μηχανισμού URL της address bar του web browser. Έτσι λοιπόν γίνεται κατανοητό ότι μετά την προσθήκη της παραπάνω εντολής και την αποθήκευση όλων των δεδομένων εκεί, η πρόσβαση πλέον στο σύστημα LMS θα πραγματοποιείται μέσω της διεύθυνσης: <http://192.168.16.58/teilms/>.

3.1.3 Εργαλεία λογισμικού

Το επόμενο βήμα στη διαδικασία είναι η εγκατάσταση των εργαλείων λογισμικού πιο συγκεκριμένα του MySQL, PHP, APACHE και του PhPMyAdmin. Πριν ξεκινήσει η εγκατάσταση, είναι απαραίτητος ο έλεγχος των repositories του εξυπηρετητή ώστε να ελεγχθεί αν το λειτουργικό σύστημα του εξυπηρετητή διαθέτει τις τελευταίες ενημερώσεις λογισμικού. Για να γίνει ο έλεγχος, εισάγουμε στη γραμμή εντολών, το εξής:

```
#apt-get update
```

Στη συνέχεια, θα αναφέρουμε τον τρόπο με τον οποίο γίνεται η εγκατάσταση των εφαρμογών που χρειαζόμαστε στο περιβάλλον Linux Debian με τη χρήση της γραμμής εντολών.

MySQL

Αρχικά πραγματοποιείται η εγκατάσταση του εξυπηρετητή διαχείρισης βάσης δεδομένων MySQL (database server) με την εντολή:

```
#apt-get install mysql-server
```

Μετά την εγκατάσταση, εμφανίζεται η οθόνη επιλογής όπου ορίζεται ο κωδικός πρόσβασης για τον χρήστη root τον οποίο δημιουργεί η MySQL αυτόματα ως διαχειριστή.

PHP

Παρομοίως για την εγκατάσταση της PHP (έκδοση 5), εισάγουμε τα παρακάτω:

```
#apt-get install php5
```

Να σημειωθεί ότι πέρα από την ίδια την PHP είναι απαραίτητη η εγκατάσταση επιπλέον τριών πακέτων λογισμικού με τα οποία η PHP μπορεί να επικοινωνήσει με την MySQL τον APACHE server και τον LDAP server αντίστοιχα (για τον ldap server θα δοθούν εξηγήσεις σε επόμενη ενότητα). Για την προσθήκη αυτών των πακέτων εισάγουμε τα εξής:

```
#apt-get install php5-mysql
```

```
#apt-get install libapache2-mod-php5
```

```
#apt-get install php5-ldap
```

APACHE

Μετά την την εγκατάσταση των προαναφερθέντων στοιχείων λογισμικού, σειρά έχει η εγκατάσταση του web server Apache ο οποίος θα είναι υπεύθυνος για την εξυπηρέτηση των αιτήσεων των χρηστών.

```
#apt-get install apache2 apache2-doc
```

PhPMyAdmin

Για την ευκολότερη διαχείριση των βάσεων δεδομένων, χρησιμοποιείται το εργαλείο PhPMyAdmin το οποίο, παρέχοντας ένα εύχρηστο περιβάλλον διασύνδεσης, περιορίζει την ανάγκη για απευθείας χρήση εντολών SQL. Η εγκατάστασή του πραγματοποιείται με την εισαγωγή των παρακάτω εντολών.

```
#apt-get install phpmysqladmin
```

```
#apt-get install php5-suhosin
```

3.2 Υλοποίηση του βασικού συστήματος

Η ενότητα αυτή αποτελεί μία πλήρη καταγραφή των βημάτων καθώς και την χρονολογική σειρά εκτέλεσης τους, με βάση τα οποία κατασκευάστηκε η πλατφόρμα LMS η οποία θα καλύψει τους στόχους και τις ανάγκες που ορίζονται στην εν λόγω συγκεκριμένη πτυχιακή εργασία.

3.2.1 Η βάση δεδομένων

Η δημιουργία της βάσης δεδομένων αποτελεί το πρώτο βήμα στην διαδικασία υλοποίησης του συστήματος καθώς, πριν πραγματοποιηθεί οποιαδήποτε διαδικασία προγραμματισμού, είναι απαραίτητο να δημιουργηθεί η βάση δεδομένων, ώστε το σύστημα να έχει την ικανότητα να διαχειρίζεται όλες τις απαραίτητες πληροφορίες οι οποίες σχετίζονται με τα συστατικά στοιχεία του όπως, παραδείγματος χάριν, στοιχεία χρηστών και βαθμολογίες μαθητών. Οι πληροφορίες αυτές χρειάζεται να βρίσκονται σε κατάσταση μόνιμης αποθήκευσης ώστε το σύστημα να έχει πρόσβαση σε αυτές οποιαδήποτε χρονική στιγμή χρειαστεί.

Παρακάτω θα θα πραγματοποιηθεί μια αναλυτική περιγραφή και αναφορά των πινάκων της βάσης δεδομένων, το σκοπό που υλοποιεί ο καθένας τους καθώς και τις συσχετίσεις μεταξύ των πινάκων.

3.2.1.1 Οι πίνακες

Σε αυτήν την ενότητα θα περιγραφούν οι πίνακες πάνω στους οποίους θα προγραμματιστεί το σύστημα, το ρόλο του κάθε πίνακα καθώς και το σύνολο των πληροφοριών που ο κάθε πίνακας συντηρεί.

Ο πίνακας lesson

Είναι ο πίνακας στον οποίο αποθηκεύονται όλες οι πληροφορίες οι οποίες σχετίζονται με το κάθε μάθημα. Αναλυτικά ο πίνακας περιέχει τις εξής πληροφορίες:

- Το αναγνωριστικό “*tei_id*” το οποίο αποτελεί τον κωδικό του μαθήματος όπως αυτός αναφέρεται στο πρόγραμμα σπουδών του τμήματος πληροφορικής του Α.Τ.Ε.Ι.Θ.
- Το πεδίο “*title*” το οποίο αποτελεί το όνομα του μαθήματος
- Ένα πεδίο “*description*” στο οποίο μπορεί κάποιος να αποδώσει μία σύντομη περιγραφή για το μάθημα σε μορφή κείμενου.
- Το πεδίο “*Semester*” το οποίο περιγράφει σε ποιο εξάμηνο του προγράμματος σπουδών (Α-Η) ανήκει το μάθημα.
- Ένα πεδίο “*id*” το οποίο χρησιμοποιείται για λόγους αρίθμησης.

Από τα παραπάνω ο συνδυασμός των πεδίων “*id*”, “*tei_id*” και “*title*” αποτελεί το κύριο κλειδί του πίνακα lesson.

Ο πίνακας sublesson

Στον πίνακα sublesson καταχωρούνται πληροφορίες που αφορούν τα τμήματα μαθητών που δημιουργούνται για ένα συγκεκριμένο μάθημα. Αναλυτικά ο πίνακας sublesson περιέχει τα εξής:

- Το πεδίο “*id*”, ένα αναγνωριστικό μορφής ακεραίου αριθμού, με το οποίο κάθε ξεχωριστό τμήμα ενός μαθήματος θα είναι γνωστό στο σύστημα
- Το πεδίο “*title*” το οποίο χρησιμοποιείται για να αποδώσουμε μία ονομασία (με τη μορφή κειμένου) στο κάθε ένα τμήμα ενός μαθήματος

- Το πεδίο “*start_date*” με το οποίο ορίζεται η ημερομηνία έναρξης (χρονιά, μήνας, μέρα, ώρα) ενός τμήματος
- Το πεδίο “*end_date*” με το οποίο ορίζεται η ημερομηνία λήξης (χρονιά, μήνας, μέρα, ώρα) ενός τμήματος

Από τα παραπάνω, τα πεδία “*id*” και “*title*” συναποτελούν το κύριο κλειδί για τον πίνακα *sublesson*.

Ο πίνακας *quiz*

Στον πίνακα αυτό θα διατηρούνται όλες οι πληροφορίες οι οποίες σχετίζονται με τα ερωτηματολόγια πολλαπλής επιλογής που δημιουργούνται για το κάθε μάθημα. Αναλυτικότερα, ο πίνακας *quiz* περιέχει τα παρακάτω:

- Το πεδίο “*id*”, ένα αναγνωριστικό μορφής ακεραίου αριθμού, με το οποίο κάθε ξεχωριστό ερωτηματολόγιο θα είναι γνωστό στο σύστημα
- Το πεδίο “*title*” το οποίο χρησιμοποιείται για να αποδώσουμε μία ονομασία (με τη μορφή κειμένου) στο ερωτηματολόγιο με την οποία θα μπορούν να το αναγνωρίζουν οι χρήστες του συστήματος. Αποφεύγεται η χρήση του πεδίου “*id*” για αυτό τον σκοπό καθώς είναι πιο εύκολο για ένα χρήστη να αναγνωρίσει παραδείγματος χάριν ότι το ερωτηματολόγιο “*Δίκτυα_2013_T5*” αποτελεί ένα ερωτηματολόγιο το οποίο δημιουργήθηκε για το τμήμα T5 του μαθήματος των Δικτύων σε σχέση με έναν ακέραιο αριθμό ο οποίος δε μπορεί να περιγράψει καμία από τις παραπάνω πληροφορίες.
- Το πεδίο “*date_starting*” στο οποίο περιγράφεται η ημερομηνία (χρονιά, μήνας, μέρα, ώρα) έναρξης της διάρκειας κατά την οποία ένα ερωτηματολόγιο θα είναι διαθέσιμο.
- Το πεδίο “*date_ending*” στο οποίο αναγράφεται η ημερομηνία (χρονιά, μήνας, μέρα, ώρα) λήξης της διάρκειας ενός ερωτηματολογίου.
- Το πεδίο “*length*” το οποίο περιγράφει τον διαθέσιμο χρόνο εξέτασης πάνω στο συγκεκριμένο ερωτηματολόγιο.

Από τα παραπάνω το πεδίο “*id*” αποτελεί το κύριο κλειδί για τον πίνακα *quiz*.

Ο πίνακας *question*

Ο πίνακας *question* περιέχει τις απαραίτητες πληροφορίες για την κάθε ερώτηση η οποία μπορεί να αποτελέσει μέρος ενός ερωτηματολογίου. Ο πίνακας αυτός περιέχει τα εξής στοιχεία:

- Το πεδίο “*id*”, ένα αναγνωριστικό μορφής ακεραίου αριθμού, με το οποίο κάθε ξεχωριστή ερώτηση θα είναι γνωστή στο σύστημα

- Το πεδίο “*description*” στο οποίο θα καταχωρείται το κείμενο (προσφώνηση) της ερώτησης
- Το πεδίο “*points*” το οποίο αποθηκεύει τη βαθμολογική αξία της συγκεκριμένης ερώτησης
- Το πεδίο “*image*”. Στο σημείο αυτό πρέπει να σημειωθεί ότι κάθε ερώτηση θα συνοδεύεται, προαιρετικά, με μία εικόνα η οποία θα προσδίδει ένα επίπεδο πληροφορίας στην ερώτηση που δεν μπορεί να καλυφθεί από το κείμενο της ερώτησης (π.χ. η τοπολογία ενός δικτύου υπολογιστών). Το πεδίο αυτό λοιπόν περιέχει σα πληροφορία το όνομα του αρχείου εικόνας το οποίο θα συνοδεύει την κάθε ερώτηση.

Ο πίνακας *answer*

Ο πίνακας *answer* περιέχει τα δεδομένα των απαντήσεων μιας συγκεκριμένης ερώτησης. Τα περιεχόμενα του είναι τα παρακάτω:

- Το πεδίο “*id*” ένα αναγνωριστικό μορφής ακεραίου αριθμού με το οποίο κάθε ξεχωριστή απάντηση θα είναι γνωστή στο σύστημα
- Το πεδίο “*description*” το οποίο περιέχει το κείμενο της συγκεκριμένης απάντησης
- Το πεδίο “*binary_truth*” το οποίο περιγράφει αν μία απάντηση είναι σωστή ή λάθος

Ο πίνακας *student*

Ο πίνακας αυτός περιέχει όλες τις απαραίτητες πληροφορίες για τους φοιτητές που αλληλεπιδρούν με το σύστημα LMS και τα μαθήματα που αυτό περιέχει. Τα στοιχεία που περιέχει ο πίνακας *student* είναι τα εξής:

- Ένα πεδίο “*id*” το οποίο χρησιμοποιείται για λόγους αρίθμησης.
- Το πεδίο “*username*” το οποίο αποτελεί το ακαδημαϊκό όνομα χρήστη του φοιτητή
- Το πεδίο “*firstname*” το οποίο περιέχει το όνομα του φοιτητή
- Το πεδίο “*lastname*” το οποίο περιέχει το επώνυμο του φοιτητή

Από τα παραπάνω πεδία, το “*username*” αποτελεί το κύριο κλειδί του πίνακα *student*

Ο πίνακας *admin*

Περιέχει τα στοιχεία του διαχειριστή του συστήματος LMS. Τα περιεχόμενά του είναι τα παρακάτω:

- Ένα πεδίο *“id”* το οποίο χρησιμοποιείται για λόγους αρίθμησης σε περίπτωση που έχουν οριστεί παραπάνω από ένας διαχειριστές για το σύστημα
- Το πεδίο *“username”* το οποίο περιέχει το όνομα χρήστη του διαχειριστή του συστήματος
- Το πεδίο *“password”* το οποίο περιέχει των κωδικό πρόσβασης του διαχειριστή
- Το πεδίο *“email”* το οποίο περιέχει την ηλεκτρονική διεύθυνση του διαχειριστή.

Από τα παραπάνω , κύριο κλειδί του πίνακα αποτελεί το πεδίο *“id”*.

3.2.1.2 Οι συσχετίσεις

Αφού πλέον έχουμε ορίσει τους πίνακες οι οποίοι περιγράφουν τις βασικές οντότητες του συστήματος, σειρά έχει η διαδικασία δημιουργίας συσχετίσεων μεταξύ των πινάκων. Οι συσχετίσεις αυτές θα περιγράψουν τον τρόπο με τον οποίο οι πίνακες αλληλεπιδρούν μεταξύ τους, προσδίδοντας έτσι επιπρόσθετη πληροφορία η οποία είναι απαραίτητη για τη σωστή λειτουργία του συστήματος.

Η συσχέτιση “Τμήμα - Μάθημα”

Μία από τις συσχετίσεις η οποία πρέπει να οριστεί είναι η σύνδεση μεταξύ ενός τμήματος με το μάθημα στο οποίο αυτό ανήκει. Για την συσχέτιση αυτή, αποδεχόμαστε την λογική ανάγκη ένα μάθημα να μπορεί να διαθέτει περισσότερα από ένα τμήματα καθώς, ένα τμήμα μπορεί να μην είναι ικανό να εξυπηρετήσει το πλήθος των φοιτητών που θέλει να παρακολουθήσει το συγκεκριμένο μάθημα. Για την επίλυση λοιπόν όλων των παραπάνω, στον πίνακα *sublesson* πραγματοποιείται η προσθήκη του πεδίου *“lesson_id”* το οποίο αποτελεί αναφορά προς το πεδίο *“id”* του πίνακα *lesson*.

Η συσχέτιση “Καθηγητής - Μάθημα”

Αφορά τη συσχέτιση η οποία προσδίδει την πληροφορία της διδασκαλίας ενός μαθήματος από ένα συγκεκριμένο καθηγητή. Η πληροφορία καθίσταται δυνατή με την προσθήκη του πεδίου *“teacher_username”* στον πίνακα *lesson*, το οποίο θα

περιέχει το ακαδημαϊκό όνομα χρήστη του καθηγητή στον οποίο έχει οριστεί η διδασκαλία του συγκεκριμένου μαθήματος.

Η συσχέτιση “Ερωτηματολόγιο - Τμήμα”

Στα πλαίσια υλοποίησης του συγκεκριμένου συστήματος έχει οριστεί ο περιορισμός κατά τον οποίο, ένα ερωτηματολόγιο δεν θα είναι γενικά διαθέσιμο μέσα στο σύστημα αλλά πρωτίστως θα πρέπει να έχει ένα οριστεί ένα συγκεκριμένο τμήμα το οποίο θα εξεταστεί στο ερωτηματολόγιο αυτό. Αυτό λοιπόν συνεπάγεται την ανάγκη συσχέτισης των οντοτήτων του τμήματος ενός μαθήματος και ενός ερωτηματολογίου το οποίο έχει οριστεί για το τμήμα αυτό. Η συσχέτιση αυτή επιτυγχάνεται με την προσθήκη του πεδίου “*sublesson_id*”, το οποίο αποτελεί αναφορά στο πεδίο “*id*” του πίνακα *sublesson*, στον πίνακα *quiz*.

Η συσχέτιση “Απάντηση – Ερώτηση”

Αποτελεί μία από τις πιο προφανείς συσχετίσεις οι οποίες πρέπει να οριστούν. Το γεγονός ότι οι πληροφορίες των ερωτήσεων και των απαντήσεων αποθηκεύονται σε ξεχωριστούς πίνακες και, όπως είναι λογικό μια απάντηση πρέπει πάντα να συνδέεται με μια ερώτηση, καθιστά απαραίτητη τη συσχέτιση μεταξύ αυτών των δύο οντοτήτων. Αυτό επιτυγχάνεται με τη προσθήκη του πεδίου “*question_id*” στον πίνακα *answer* το οποίο αποτελεί αναφορά στο πεδίο “*id*” του πίνακα *question* καθώς και τον ορισμό στον πίνακα *answer* των συνδυασμό των πεδίων “*id*” και “*question_id*” ως κύριο κλειδί. Ο λόγος για τον οποίο επιλέγουμε την προσθήκη του επιπλέον πεδίου συσχέτισης στον πίνακα *answers* είναι πρακτικός καθώς, μία απάντηση σχετίζεται με μία μόνο ερώτηση σε αντίθεση με τις ερωτήσεις οι οποίες πολύ πιθανόν να σχετίζονται με παραπάνω από μια απαντήσεις. Άρα λοιπόν η επιλογή του πίνακα *question* για την προσθήκη του αντίστοιχου πεδίου συσχέτισης θα δημιουργούσε πολύ περισσότερο πλήθος εγγράφων.

Η συσχέτιση “Ερώτηση – Μάθημα”

Ένας ακόμη περιορισμός του συγκεκριμένου συστήματος είναι ότι, κάθε ερώτηση η οποία δημιουργείται για να χρησιμοποιηθεί στα πλαίσια ενός ερωτηματολογίου, πρέπει πάντα να σχετίζεται με ένα συγκεκριμένο μάθημα. Η ανάγκη ύπαρξης αυτού του περιορισμού γίνεται εύκολα αντιληπτή αν αναλογιστεί κανείς, παραδείγματος χάριν, ότι είναι προφανώς αναμενόμενο στην διαδικασία σύνθεσης ενός ερωτηματολογίου για το μάθημα Προγραμματισμός C++ να μην εμφανίζονται ως διαθέσιμες για προσθήκη, ερωτήσεις οι οποίες σχετίζονται με το μάθημα Δίκτυα Η/Υ. Για την επίτευξη λοιπόν της παραπάνω συσχέτισης, πραγματοποιείται η προσθήκη του πεδίου “*lesson_id*” στον πίνακα *question*, το οποίο αποτελεί αναφορά στο πεδίο “*id*” του πίνακα *lesson*, καθώς και του ορισμού στον πίνακα *question* των πεδίων “*id*” και “*lesson_id*” ως κύριο κλειδί.

Η συσχέτιση “Φοιτητής – Τμήμα”

Μία ακόμη συσχέτιση απαραίτητη για το σύστημα, είναι αυτή που περιγράφει σε ποιο τμήμα ανήκει ένας φοιτητής στα πλαίσια ενός μαθήματος. Επειδή όμως, η συσχέτιση του τμήματος με το μάθημα στο οποίο ανήκει καλύφθηκε παραπάνω, το μόνο που απομένει είναι η συσχέτιση του φοιτητή με το τμήμα. Για επίτευξη αυτής της συσχέτισης, δημιουργούμε τον πίνακα `student_sublesson` ο οποίος:

- Περιέχει το πεδίο `“student_id”` το οποίο αποτελεί αναφορά στο πεδίο `“id”` του πίνακα `student`
- Περιέχει το πεδίο `“sublesson_id”` το οποίο αποτελεί αναφορά στο πεδίο `“id”` του πίνακα `sublesson`
- Όλα τα παραπάνω πεδία αποτελούν το κύριο κλειδί αυτού του πίνακα.

Ο λόγος για τον οποίο δημιουργήθηκε ένας ξεχωριστός πίνακας για να περιγράψει την εν λόγω συσχέτιση είναι καθαρά πρακτικός καθώς, η φύση της συγκεκριμένης συσχέτισης είναι τέτοια ώστε να δημιουργεί επαναλαμβανόμενη πληροφορία και για τις δύο οντότητες οι οποίες συμμετέχουν στην συγκεκριμένη συσχέτιση καθώς:

- Ένας φοιτητής μπορεί να ανήκει σε περισσότερα από ένα τμήματα
- Ένα τμήμα μπορεί να διαθέτει περισσότερους από έναν φοιτητές

Δεδομένων των παραπάνω καθώς και του γεγονότος ότι η δομή των πινάκων `student` και `sublesson` είναι τέτοια ώστε να έχουμε μία εγγραφή για κάθε ένα φοιτητή και τμήμα αντίστοιχα, κρίνεται σωστό η δημιουργία ενός ξεχωριστού πίνακα για την συγκεκριμένη συσχέτιση.

Η συσχέτιση “Φοιτητής – Ερωτηματολόγιο”

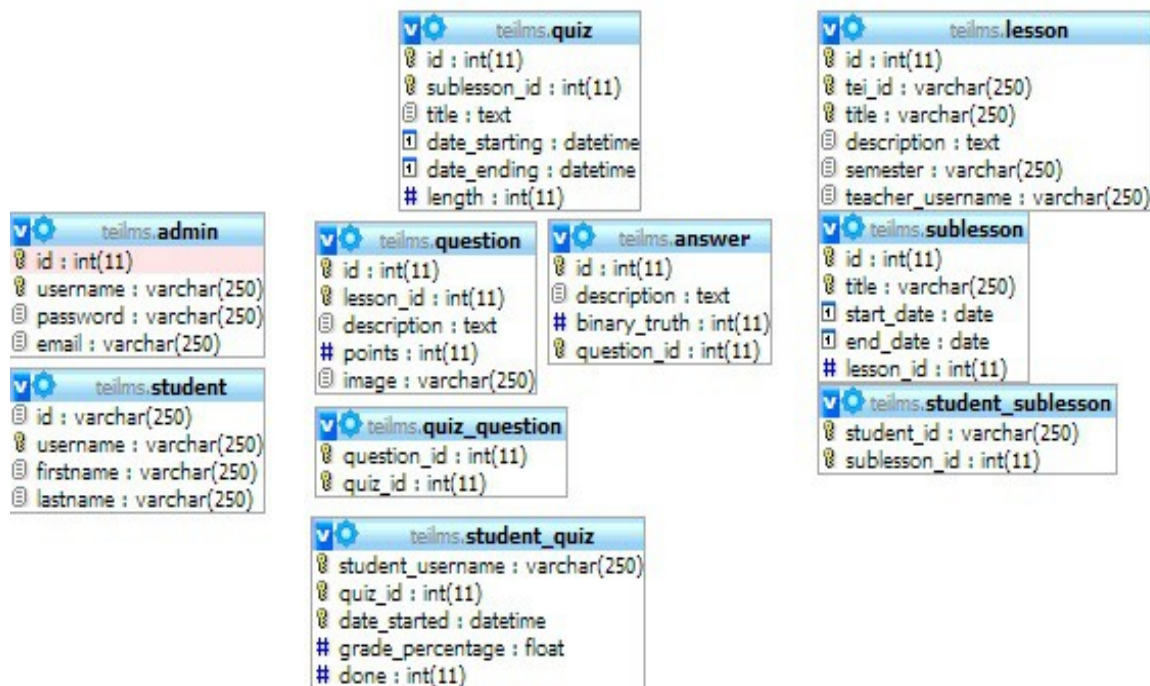
Η συσχέτιση αυτή περιγράφει την διαδικασία αξιολόγησης ενός φοιτητή σε ένα συγκεκριμένο ερωτηματολόγιο. Επειδή, για ακόμη μία φορά, συναντάμε την περίπτωση η συσχέτιση να δημιουργεί επαναλαμβανόμενη πληροφορία για τα στοιχεία των πινάκων τους οποίους συνδέει, θεωρείται προτιμότερη η δημιουργία ενός ξεχωριστού πίνακα για την περιγραφή της. Έτσι λοιπόν δημιουργείται ο πίνακας `student_quiz` ο οποίος περιέχει τα εξής:

- Το πεδίο `“student_username”` το οποίο αποτελεί αναφορά στο πεδίο `“username”` του πίνακα `student` και περιγράφει, με βάση το ακαδημαϊκό όνομα χρήστη, τον φοιτητή ο οποίος εξετάστηκε σε ένα συγκεκριμένο ερωτηματολόγιο

- Το πεδίο “*quiz_id*” το οποίο αποτελεί αναφορά στο πεδίο “*id*” του πίνακα quiz και περιγράφει σε ποιο ερωτηματολόγιο εξετάστηκε ο φοιτητής του οποίου τα στοιχεία παρέχονται απ' το παραπάνω πεδίο
- Το πεδίο “*date_started*” το οποίο περιγράφει την ημερομηνία (χρονιά, μήνας, μέρα, ώρα) κατά την οποία εξετάστηκε ο φοιτητής στο συγκεκριμένο ερωτηματολόγιο
- Το πεδίο “*grade_percentage*” στο οποίο καταχωρείται η βαθμολογία που σημείωσε ο φοιτητής στο συγκεκριμένο ερωτηματολόγιο
- Το πεδίο “*done*” το οποίο παρέχει την πληροφορία για το αν ο φοιτητής έχει ήδη εξεταστεί μια φορά στο συγκεκριμένο ερωτηματολόγιο

Από τα παραπάνω, ο συνδυασμός των πεδίων “*student_username*”, “*quiz_id*” και “*date_started*” αποτελεί τον ορισμό του κυρίου κλειδιού γι αυτόν τον πίνακα.

Για την διευκόλυνση του αναγνώστη, παρέχεται μία οπτική αναπαράσταση των πινάκων και των συσχετίσεων που δημιουργήθηκαν παραπάνω. Η αναπαράσταση αυτή περιγράφεται στην παρακάτω εικόνα:



Εικόνα 7: Πίνακες της Βάσης Δεδομένων

3.2.2 Η ανάπτυξη του συστήματος με τη γλώσσα PHP

3.2.2.1 Sessions

Σε αυτήν την ενότητα θα γίνει μια σύντομη αναφορά για την χρήση των sessions στο σύστημα LMS καθώς και τον σκοπό που αυτές εξυπηρετούν στην λειτουργικότητα του συστήματος.

Όπως αναφέρθηκε προηγουμένως στο κεφάλαιο 2, το πρωτόκολλο HTTP είναι ένα stateless πρωτόκολλο επικοινωνίας το οποίο δεν επιτρέπει να συγκρατούνται πληροφορίες σχετικά με τις αιτήσεις τις οποίες δέχεται ο web server από τους clients. Αυτό έχει ως αποτέλεσμα διαδοχικές αιτήσεις οι οποίες προέρχονται από τον ίδιο χρήστη, να μην μπορούν να συσχετιστούν μεταξύ τους κάτω απ την κοινή αυτή συνιστώσα καθώς το HTTP, είναι σχεδιασμένο να αναγνωρίζει μία αίτηση για κάθε μία αλληλεπίδραση ή αλλιώς συνεδρία (session) του κάθε διαφορετικού χρήστη με τον εξυπηρετητή.

Για την αντιμετώπιση του παραπάνω προβλήματος θα γίνει η χρήση sessions, μια τεχνική η οποία επιτρέπει να διατηρούνται πληροφορίες για τον κάθε χρήστη για το σύνολο της χρονικής διάρκειας όπου αυτός επικοινωνεί και ανταλλάσσει δεδομένα με τον εξυπηρετητή. Η php είναι μια γλώσσα προγραμματισμού η οποία υποστηρίζει την χρήση sessions οπότε και θα εκμεταλλευτούμε αυτήν την ιδιότητά της στην ανάπτυξη του συστήματος.

Ο τρόπος με τον οποίο διαχειρίζεται η php τα sessions είναι αρκετά απλή και περιγράφεται από τα παρακάτω βήματα:

- Για κάθε ένα χρήστη που πραγματοποιεί σύνδεση με τον web server, η php αρχικοποιεί μία διαδικασία συνεδρίας με τη χρήση `session_start()` κατά την οποία ο server δημιουργεί ένα μοναδικό αναγνωριστικό (`session_ID`), με το οποίο αναγνωρίζεται το σύνολο των δεδομένων τα οποία ανταλλάσσονται μεταξύ του χρήστη και του server. Η php αποθηκεύει τα δεδομένα αυτά σε μορφή μεταβλητών οι οποίες ονομάζονται session μεταβλητές.
- Στην περίπτωση όπου ένας χρήστης διαθέτει ήδη `session_ID`, πιθανόν δεδομένα τα οποία έχουν δημιουργηθεί σε προηγούμενη φάση και σχετίζονται με αυτόν, ανασύρονται και τοποθετούνται στο χώρο μεταβλητών του χρήστη
- Οι μεταβλητές που δημιουργούνται κατά την διάρκεια ενός session καθώς και τα δεδομένα που αυτές περιέχουν, είναι διαθέσιμες μέσω της superglobal μεταβλητής `$_SESSION`.
- Τέλος, κατά την αποσύνδεση του χρήστη από το σύστημα, η συνεδρία η οποία είχε δημιουργηθεί τερματίζεται με τη χρήση της `session_destroy()`.

3.2.2.2 Σύνδεση με την βάση

Πριν από το σχεδιασμό οποιασδήποτε λειτουργικότητας του συστήματος, προαπαιτούμενο βήμα αποτελεί η υλοποίηση του κώδικα με την οποία, θα πραγματοποιείται η σύνδεση με τη βάση δεδομένων για την πρόσβαση στους πίνακες καθώς και την διαχείριση των δεδομένων τους. Αυτό πραγματοποιείται με την χρήση της συνάρτησης *mysqli_connect()* η οποία δημιουργεί ένα ρεύμα σύνδεσης (connection) με τη βάση δεδομένων και λαμβάνει τις εξής παραμέτρους για την σύνδεση :

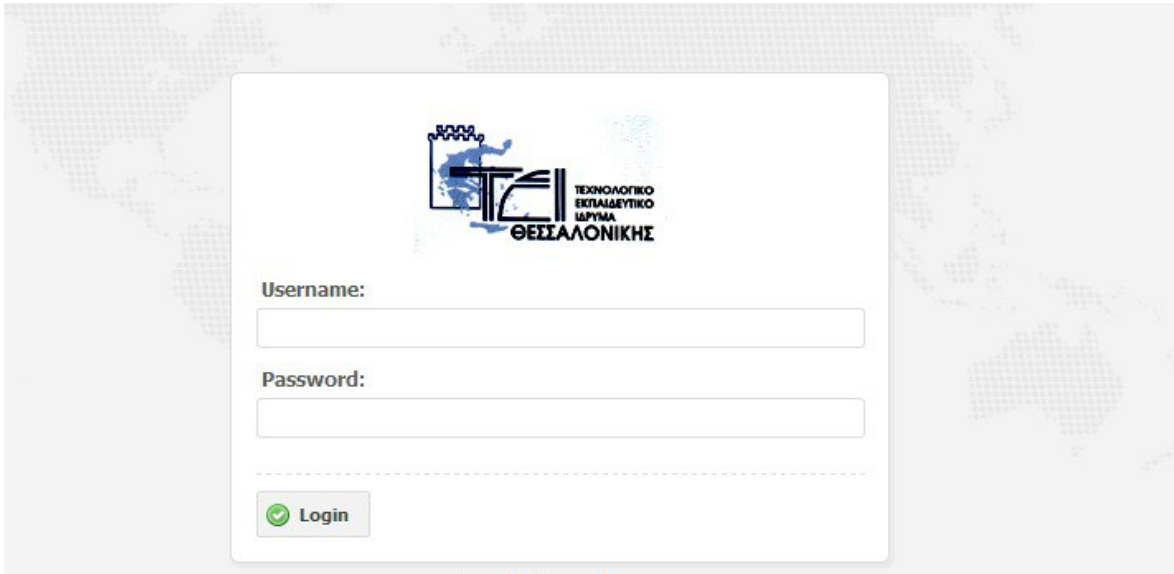
- Την διεύθυνση του database server (στη συγκεκριμένη περίπτωση τον MySQL server)
- Τα διαπιστευτήρια του διαχειριστή τον οποίο ορίσαμε για τον MySQL server
- Το όνομα της βάσης δεδομένων μέσα στην οποία δημιουργήσαμε τους πίνακες

Επειδή η περίπτωση να έχουμε ανοικτή την σύνδεση προς την βάση δεδομένων είναι μια ενέργεια η οποία καταναλώνει άσκοπα τους πόρους του συστήματος για το διάστημα που δεν χρησιμοποιείται η βάση, αντ' αυτού προτιμάται να δημιουργείται η σύνδεση με την βάση μόνο όταν αυτό κριθεί αναγκαίο και στη συνέχεια η σύνδεση να τερματίζεται. Για τον λόγο αυτό λοιπόν αλλά και για να αποφύγουμε τη επανάληψη του κώδικα για την σύνδεση με την βάση, η κλήση της συνάρτησης αποθηκεύεται στην μεταβλητή *\$mysqli_link* στο αρχείο *conenct.php* το οποίο και θα εισάγεται με την χρήση της εντολής *include* σε οποιαδήποτε άλλο αρχείο το οποίο χρειαστεί σύνδεση με την βάση.

3.2.2.3 Η σελίδα login

Είναι η πρώτη σελίδα την οποία θα μελετήσουμε καθώς αποτελεί το απαραίτητο πρώτο βήμα που πρέπει να ακολουθήσει ο κάθε χρήστης, ώστε να αναγνωριστεί από το σύστημα με βάση τα στοιχεία σύνδεσής του και στη συνέχεια να του επιτραπεί η χρήση του συστήματος.

Η σελίδα login είναι διαθέσιμη μέσω της διεύθυνσης URL <http://192.168.16.58/teilms/login.php> η οποία έχει την εξής εμφάνιση:



Εικόνα 8: Σελίδα Login

Στα κενά πεδία τα οποία φαίνονται, ο χρήστης συμπληρώνει τα στοιχεία του ακαδημαϊκού ονόματος χρήστη καθώς και του κωδικού πρόσβασης και στην συνέχεια επιλέγει το κουμπί Login ώστε να ξεκινήσει η διαδικασία ταυτοποίησης. Τα παραπάνω πεδία αποτελούν το σώμα της φόρμας μέσω της οποίας θα προωθηθούν τα δεδομένα προς επεξεργασία ώστε να ταυτοποιηθεί ο χρήστης. Τέλος το οπτικό περιεχόμενο της σελίδας δημιουργείται με την χρήση σημάνσεων (tags) HTML καθώς και την χρήση μορφοποιήσεων (css).

Στο λειτουργικό κομμάτι τώρα της σελίδας πραγματοποιείται η εισαγωγή (include) δύο επιπλέον αρχείων: το *connect.php* στο οποίο αναφερθήκαμε παραπάνω καθώς και το αρχείο *functions.php* που περιέχει μία συλλογή συναρτήσεων κάθε μία από τις οποίες εξυπηρετούν μία συγκεκριμένη λειτουργικότητα του συστήματος. Μερικές από αυτές τις συναρτήσεις αυτές θα αναλυθούν στην πορεία περιγραφής του συστήματος.

Στη συνέχεια με την εκτέλεση της εντολής *session_start()* αρχικοποιείται η διαδικασία δημιουργίας συνεδρίας για τον χρήστη ο οποίος προσπαθεί να συνδεθεί στο σύστημα. Στα πλαίσια της διαδικασίας αυτής πραγματοποιείται έλεγχος κατά τον οποίο εξετάζεται εάν ο χρήστης σχετίζεται ήδη με μια συνεδρία ή όχι. Εάν όντως σχετίζεται τότε, η προσωπική του συνεδρία ενεργοποιείται και συνεχίζει από το σημείο στο οποίο είχε σταματήσει ενώ διαφορετικά μία καινούρια συνεδρία δημιουργείται γι αυτόν.

Μετά την εκτέλεση των παραπάνω, επόμενο βήμα αποτελεί ο έλεγχος για τον εάν έχει πατηθεί το κουμπί *login* στην φόρμα της σελίδας, γεγονός το οποίο συνεπάγεται την έναρξη της διαδικασίας ταυτοποίησης του χρήστη.

Η διαδικασία αυτή χωρίζεται σε δύο υποπεριπτώσεις

1. Ο χρήστης που επιχειρεί να συνδεθεί είναι ο διαχειριστής του συστήματος.
2. Οποιαδήποτε άλλη περίπτωση

Στην πρώτη περίπτωση, ο έλεγχος πραγματοποιείται στη βάση δεδομένων η οποία περιέχει τα στοιχεία σύνδεσης του διαχειριστή αποθηκευμένα στον πίνακα `admin`. Κατά τον έλεγχο αυτό δημιουργείται ένα ερώτημα `sql` προς την βάση δεδομένων για το εάν υπάρχει εγγραφή στον πίνακα `admin` η οποία να συμφωνεί με τα στοιχεία εισόδου τα οποία εισήχθησαν στην φόρμα της σελίδας `login`. Εάν το αποτέλεσμα είναι θετικό τότε ενημερώνονται οι παρακάτω `session` μεταβλητές με τα εξής στοιχεία:

- Η μεταβλητή `logged_in` λαμβάνει την τιμή “1” το οποίο συνεπάγεται ότι η είσοδος του χρήστη ήταν επιτυχής
- Η μεταβλητή `profile`, μέσω της οποίας γίνεται γνωστό στο σύστημα σε ποια ομάδα χρηστών ανήκει, λαμβάνει την τιμή “admin”
- Η μεταβλητή `username` στην οποία αποθηκεύεται το όνομα χρήστη το οποίο εισήγαγε ο χρήστης στην φόρμα της σελίδας `login` λαμβάνει την τιμή “admin”.

Τέλος πραγματοποιείται ανακατεύθυνση του χρήστη προς την αρχική σελίδα του περιβάλλοντος του διαχειριστή.

Στην δεύτερη περίπτωση περιλαμβάνονται όλες οι υπόλοιπες ομάδες χρηστών δηλαδή, η προσπάθεια εισόδου στο σύστημα ως φοιτητής ή ως καθηγητής. Ο λόγος για τον οποίο οι δύο αυτές περιπτώσεις τοποθετούνται μαζί είναι γιατί, πολύ απλά, και για τις δύο περιπτώσεις χρησιμοποιείται ακριβώς η ίδια διαδικασία ταυτοποίησης η οποία περιγράφεται παρακάτω.

Για την διατήρηση του μεγάλου όγκου πληροφορίας των ατόμων που ανήκουν στο τμήμα Πληροφορικής, έχει επιλεγεί η χρήση ενός `LDAP server` ο οποίος εξυπηρετεί λειτουργίες βάσης δεδομένων και περιέχει τα στοιχεία των φοιτητών και των καθηγητών του τμήματος Πληροφορικής. Έτσι λοιπόν για την ταυτοποίηση φοιτητών ή καθηγητών αντί της βάσης δεδομένων του `LMS`, ο έλεγχος θα πραγματοποιείται στην βάση δεδομένων του `LDAP server`.

Πρώτο βήμα του ελέγχου αυτού αποτελεί η κλήση από το αρχείο `functions.php` της συνάρτησης `verify_user()` η οποία δέχεται σαν ορίσματα το `username` και το `password` τα οποία εισήχθησαν στην φόρμα της σελίδας `login` και στη συνέχεια εκτελεί αναζήτηση στη βάση δεδομένων του `LDAP server` για την εύρεση ενός χρήστη (φοιτητή ή καθηγητή) ο οποίος να ικανοποιεί τα κριτήρια αναζήτησης. Μετά το τέλος της αναζήτησης, η συνάρτηση επιστρέφει την μεταβλητή `$status` η οποία

περιέχει το αποτέλεσμα της αναζήτησης. Οι τιμές που λαμβάνει η μεταβλητή είναι οι εξής:

- **0** εάν το ένα από τα δύο ή και τα δύο στοιχεία του χρήστη δε βρέθηκαν
- **stud** εάν η ταυτοποίηση ήταν επιτυχής και ο χρήστης είναι φοιτητής
- **staff** εάν η ταυτοποίηση ήταν επιτυχής και ο χρήστης είναι καθηγητής
- **2** εάν απέτυχε η αυθεντικοποίηση των στοιχείων του διαχειριστή LDAP
- **3** εάν γενικά απέτυχε η σύνδεση με τον LDAP server.

Ο κώδικας της συνάρτησης είναι διαθέσιμος στο παράρτημα K-1.6

Στη συνέχεια, με βάση το αποτέλεσμα το οποίο επιστρέφει η συνάρτηση, πραγματοποιείται ένα δεύτερο βήμα ελέγχου ροής κατά το οποίο εάν η ταυτοποίηση ήταν επιτυχής (δηλαδή `$status="stud"` ή `$status="staff"`) τότε:

- Η μεταβλητή `logged_in` λαμβάνει την τιμή "1"
- Η μεταβλητή `username` λαμβάνει ως τιμή ο,τι συμπλήρωσε ο χρήστης στο πεδίο username της φόρμας της σελίδας login.

Τέλος πραγματοποιείται ένας τελευταίος έλεγχος ροής κατά τον οποίο:

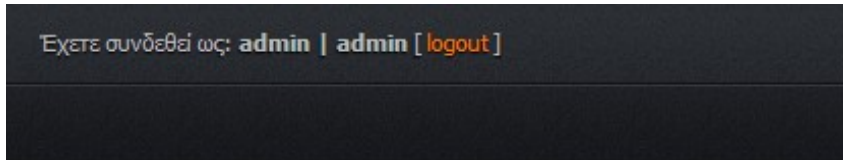
- Εάν ο χρήστης είναι φοιτητής, η μεταβλητή `profile` λαμβάνει την τιμή "student" και ο χρήστης μεταφέρεται στην αρχική σελίδα του φοιτητή.
- Εάν ο χρήστης είναι καθηγητής, η μεταβλητή `profile` λαμβάνει την τιμή "staff" και ο χρήστης μεταφέρεται στην αρχική σελίδα του καθηγητή.

Περισσότερες πληροφορίες γι' αυτές τις σελίδες θα δοθούν σε επόμενες ενότητες αυτού του κεφαλαίου.

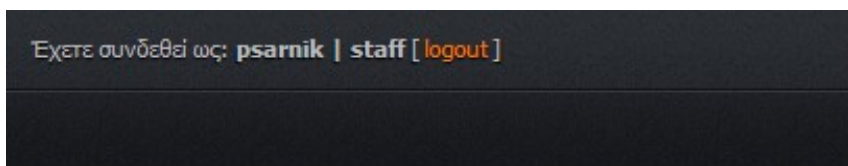
Βέβαια πρέπει να σημειωθεί ότι πέρα όλων των παραπάνω περιπτώσεων, υπάρχει πάντα το ενδεχόμενο κάποιος να προσπαθήσει να προσπελάσει την σελίδα login ενώ έχει ήδη ταυτοποιηθεί και συνδεθεί στο σύστημα. Για την προηγούμενη περίπτωση λοιπόν η σελίδα login διαθέτει ακόμα έναν έλεγχο κατά τον οποίο εξετάζεται εάν η τιμή της session μεταβλητής `logged_in` είναι ίση με "1", γεγονός το οποίο ισχύει όταν ο χρήστης είναι πλέον συνδεδεμένος στο σύστημα. Στην περίπτωση που αυτό ισχύει, τότε ο χρήστης μεταφέρεται στην κατάλληλη σελίδα, ανάλογα με την τιμή της μεταβλητής `profile`.

Έτσι λοιπόν, ανάλογα με τις τιμές τις οποίες συλλέγει η μεταβλητή `profile` πραγματοποιείται η επιλογή του κατάλληλου περιεχομένου το οποίο θα εμφανιστεί στον κάθε χρήστη. Η εικόνα που φαίνεται παρακάτω, δίνεται ως ένα μικρό παράδειγμα για την συμπεριφορά του συστήματος στις περιπτώσεις όπου ένας

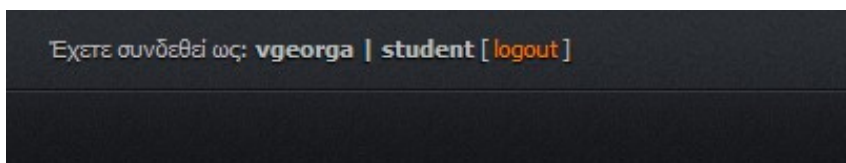
χρήστης επιχειρεί είσοδο στο σύστημα ως διαχειριστής, καθηγητής και φοιτητής αντίστοιχα.



Εικόνα 9: Σύνδεση με δικαιώματα διαχειριστή



Εικόνα 10: Σύνδεση με δικαιώματα καθηγητή



Εικόνα 11: Σύνδεση με δικαιώματα φοιτητή

Στις επόμενες ενότητες, θα περιγράψουμε το περιβάλλον του συστήματος που είναι διαθέσιμο για την κάθε μία ομάδα χρηστών, καθώς και το σύνολο των εργαλείων που αυτό προσφέρει.

3.2.2.3 Header.php

Προτού γίνει οποιαδήποτε αναφορά στην λειτουργία του υπόλοιπου συστήματος, θεωρείται σημαντικό να γίνει μία μικρή αναφορά στο αρχείο *header.php* καθώς και τον λόγο χρήσης του.

Το αρχείο *header.php* χρησιμοποιείται ως ένα στοιχείο “δεξαμενή”, στο οποίο ενσωματώνονται μέρη κώδικα που αποτελούν κοινό τόπο χρήσης από τα υπόλοιπα αρχεία του συστήματος. Χαρακτηριστικά θα αναφερθεί ότι εμπεριέχει,

εκτός πολλών άλλων, με χρήση της *include* τα αρχεία *connect.php* και *functions.php* ώστε τα υπόλοιπα αρχεία να έχουν την ικανότητα σύνδεσης με τη βάση καθώς και τη χρήση συναρτήσεων αντίστοιχα χωρίς τη χρήση επιπρόσθετου κώδικα. Έτσι λοιπόν σε όσες αναφορές πρόσβασης στην βάση δεδομένων ή χρήσης συναρτήσεων πραγματοποιηθούν στις επόμενες ενότητες από τα διάφορα αρχεία κώδικα, θα είναι λόγω της ενσωμάτωσης (*include*) του αρχείου *header.php* σε αυτά

Παράλληλα το αρχείο αυτό παρέχει την συνεχή πληροφορία των στοιχείων του χρήστη ο οποίος είναι συνδεδεμένος στο σύστημα με τη χρήση των μεταβλητών *profile* και *username* τις οποίες αναφέραμε στην προηγούμενη ενότητα. Τα στοιχεία αυτά εμφανίζονται στην πάνω αριστερά περιοχή του παραθύρου.

Τέλος, το αρχείο *header.php* είναι υπεύθυνο για την διαδικασία αποσύνδεσης του χρήστη από το σύστημα καθώς και του τερματισμού της συνεδρίας (*session*) η οποία ήταν ενεργή όσο ήταν ακόμη συνδεδεμένος. Η επιλογή αυτή βρίσκεται δίπλα από τα στοιχεία σύνδεσης του χρήστη.

3.2.2.4 Η σελίδα του Διαχειριστή (Administrator)

Μετά από την επιτυχή του σύνδεση με το σύστημα, ο διαχειριστής μεταφέρεται στην προσωπική του σελίδα της οποίας το interface φαίνεται στην παρακάτω εικόνα.



The screenshot shows the Administrator interface. On the left is a sidebar with navigation options: > Διαχείριση Μαθημάτων, Διαθέσιμα Μαθήματα, Προσθήκη Μαθήματος, Επεξεργασία Μαθημάτων, > Ρυθμίσεις, Αλλαγή κωδικού administrator, Συγχρονισμός με τη βάση LDAP, and Διαγραφή Ανενεργών Φοιτητών. The main content area is titled 'Διαθέσιμα Μαθήματα' and 'Μαθήματα'. It contains a table with the following data:

Κωδικός	Τίτλος	Διδάσκων
3401	Δίκτυα Η/Υ	peris,psarnik,adamidis
4202	Εισαγωγή στα Λειτουργικά Συστήματα	asidirop

Εικόνα 12: Interface σελίδας διαχειριστή

Το περιβάλλον αποτελείται από τρία panel, ένα κεντρικό στη μέση της σελίδας και δύο ακόμα στο αριστερό μέρος της.

Η δημιουργία του κεντρικού panel αποτελεί το αποτέλεσμα της εκτέλεσης του αρχείου *list_lessons.php*, του οποίου το περιεχόμενο συμπεριλαμβάνεται σαν αυτόματη ενέργεια κατά την φόρτωση της αρχικής σελίδας του διαχειριστή. Κατά την εκτέλεση του, το παραπάνω script δημιουργεί ένα ερώτημα προς τη βάση δεδομένων και τα ανασύρει όλες τις εγγραφές του πίνακα lesson, κάθε μία από τις οποίες εμφανίζει με τα επιμέρους πεδία της στον γραφικό πίνακα ο οποίος δημιουργείται με τη χρήση σημάτων HTML και στοιχείων μορφοποίησης CSS. Σε περίπτωση που δεν έχουν δημιουργηθεί μαθήματα ακόμα, το κεντρικό panel παρουσιάζεται κενό. Όσο για τα δύο panel τα οποία βρίσκονται αριστερά, περιέχουν σε μορφή λίστας τις επιλογές των εργαλείων διαχείρισης του χρήστη. Η περιγραφή του εκάστοτε εργαλείου καθώς και της λειτουργικότητάς του θα πραγματοποιηθεί στη συνέχεια της ενότητας.

Το πρώτο panel με τίτλο “Διαχείριση Μαθημάτων” περιέχει τις εξής επιλογές:

- Διαθέσιμα Μαθήματα
- Προσθήκη Μαθήματος
- Επεξεργασία Μαθημάτων

Η πρώτη επιλογή έχει ακριβώς την ίδια λειτουργικότητα με αυτή που περιγράφηκε παραπάνω για το κεντρικό panel οπότε και δε θα αναλυθεί περαιτέρω.

Η λειτουργία “Προσθήκη Μαθήματος” προσφέρει στον διαχειριστή την δυνατότητα να δημιουργήσει ένα νέο μάθημα. Κατά την επιλογή της λειτουργίας αυτής εμφανίζεται στο κεντρικό panel της σελίδας μία φόρμα αλληλεπίδρασης με τον χρήστη στην οποία συμπληρώνει τα στοιχεία του νέου μαθήματος το οποίο θέλει να προσθέσει. Τα στοιχεία αυτά είναι τα εξής:

- Ο κωδικός του μαθήματος όπως αυτός αναφέρεται στον πρόγραμμα σπουδών του τμήματος Πληροφορικής του Α.Τ.Ε.Ι.Θ
- Ο τίτλος του μαθήματος
- Το username του καθηγητή ο οποίος θα είναι υπεύθυνος για το εν λόγω μάθημα
- Το εξάμηνο του προγράμματος σπουδών στο οποίο ανήκει το μάθημα
- Ένα σύντομο κείμενο για την περιγραφή του μαθήματος.

Η φόρμα αυτή φαίνεται στην παρακάτω εικόνα:

Προσθήκη Μαθήματος

Κωδικός Μαθήματος *

4804

Τίτλος *

Ειδικά Θέματα Δικτύων 1

Καθηγητής (username) *

psarnik

Εξάμηνο*

θ

Περιγραφή

CCNA 1, CCNA 2

+ Προσθήκη X Εκκαθάριση

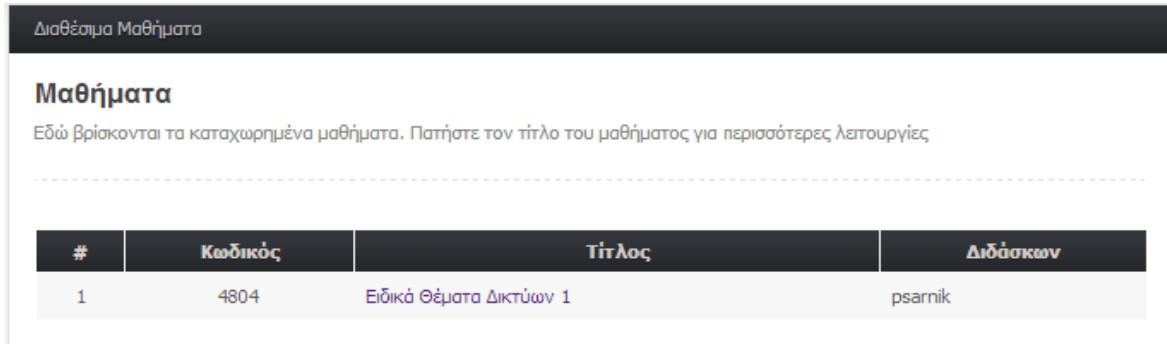
Εικόνα 13: Προσθήκη μαθήματος

Μετά την συμπλήρωση των πεδίων, ο χρήστης μπορεί να επιλέξει το κουμπί “Προσθήκη” μέσω της οποίας, εκτελείται ένα ερώτημα προς την βάση και δημιουργείται μία νέα εγγραφή στον πίνακα lesson που περιέχει τα στοιχεία της φόρμας ή εναλλακτικά μπορεί να επιλέξει “Εκκαθάριση” ώστε να καθαρίσει η φόρμα από τα στοιχεία που συμπλήρωσε ο χρήστης χωρίς τη δημιουργία εγγραφής στη βάση δεδομένων. Εάν η διαδικασία της προσθήκης του μαθήματος είναι επιτυχής στο επάνω μέρος της φόρμας θα εμφανιστεί το εξής μήνυμα:

✔ Το μάθημα **Ειδικά Θέματα Δικτύων 1** προστέθηκε επιτυχώς. Επιστροφή στα **Μαθήματα**

Εικόνα 14: Ανάδραση προσθήκης μαθήματος

Η επιβεβαίωση της προσθήκης του νέου μαθήματος μπορεί να γίνει επίσης με την επιλογή του συνδέσμου “Μαθήματα” του παραπάνω μηνύματος ή με την επιλογή του συνδέσμου “Διαθέσιμα Μαθήματα” από το panel “Διαχείριση Μαθημάτων”.



#	Κωδικός	Τίτλος	Διδάσκων
1	4804	Ειδικά Θέματα Δικτύων 1	psarnik



Εικόνα 15: Διαθέσιμα μαθήματα

Ένα ακόμη σημείο το οποίο αξίζει να σημειωθεί είναι ότι κατά την επιλογή προσθήκης των στοιχείων του νέου μαθήματος, πριν την καταχώρησή τους στην βάση δεδομένων, πραγματοποιείται έλεγχος στην βάση δεδομένων του LDAP server για την εγκυρότητα του username του καθηγητή το οποίο συμπλήρωσε ο χρήστης στην φόρμα. Στην περίπτωση που το username βρεθεί να μην είναι έγκυρο, η εγγραφή των στοιχείων στον πίνακα lesson ακυρώνεται και εμφανίζεται το παρακάτω μήνυμα.







Εικόνα 16: Ανάδραση ελέγχου username καθηγητή

Η λειτουργικότητα για την προσθήκη ενός νέου μαθήματος υλοποιείται από τον κώδικα που περιέχεται στο αρχείο `add_lesson.php`. Επιπρόσθετα γίνεται χρήση της συνάρτησης `add_lesson()` που είναι υπεύθυνη για την δημιουργία της νέας εγγραφής στον πίνακα lesson και της `check_user()` η οποία αναλαμβάνει τον έλεγχο εγκυρότητας του καθηγητή. Το πλήρες περιεχόμενο των `add_lesson.php` και `add_lesson()` βρίσκεται στο παράρτημα κώδικα K-1.4 και K-1.5 αντίστοιχα.

Διαθέσιμα Μαθήματα				
Μαθήματα				
Εδώ βρίσκονται τα καταχωρημένα μαθήματα. Πατήστε επεξεργασία ή διαγραφή.				
#	Κωδικός	Τίτλος	Διδάσκων	Επεξεργασία
1	4804	Ειδικά Θέματα Δικτύων 1	psarnik	 

Εικόνα 17: Εργαλείο επεξεργασίας Διαθέσιμων Μαθημάτων

Η λειτουργία “Εξεργασία Μαθημάτων” δίνει την δυνατότητα στον διαχειριστή του συστήματος να αλλάξει τα στοιχεία ενός καταχωρημένου μαθήματος ή να το διαγράψει εντελώς. Με την επιλογή της λειτουργίας, ενεργοποιείται το script `edit_lessons.php` του οποίου το περιεχόμενο εμφανίζεται στο χώρο του κεντρικού panel όπως φαίνεται στην παρακάτω εικόνα.

Κωδικός	Τίτλος	Διδάσκων	Επεξεργασία
3401	Δίκτυα Η/Υ	peris	 
4202	Εισαγωγή στα Λειτουργικά Συστήματα	asidirop	 

Εικόνα 18: Τα διαθέσιμα μαθήματα μαζί με τα εργαλεία επεξεργασίας

Για κάθε μάθημα, δίπλα από το πεδίο του διδάσκοντα καθηγητή παρατίθεται ένα ζεύγος εργαλείων τα οποία εξυπηρετούν τις λειτουργίες επεξεργασίας και διαγραφής του μαθήματος αντίστοιχα.

Στην επιλογή του εργαλείου επεξεργασίας, εμφανίζεται στο κεντρικό panel μία φόρμα αλληλεπίδρασης, όμοια με αυτή της προσθήκης μαθήματος, στην οποία περιέχονται τα στοιχεία του μαθήματος όπως αυτά καταχωρήθηκαν στην διαδικασία της προσθήκης του μαθήματος, εάν το μάθημα αυτό επεξεργάζεται για πρώτη φορά. Σε διαφορετική περίπτωση περιέχει τα δεδομένα από την τελευταία ενημέρωση. Ο χρήστης έχει την δυνατότητα να αλλάξει οποιαδήποτε από τα πεδία με μία νέα τιμή και στη συνέχεια να επιλέξει το κουμπί “Ενημέρωση”, εάν επιθυμεί οι αλλαγές αυτές να περαστούν στη βάση δεδομένων ώστε να έχουν μόνιμη ισχύ, ή εναλλακτικά να επαναφέρει την φόρμα στις τιμές τις οποίες είχε κατά την εμφάνισή της με την επιλογή του κουμπιού “Επαναφορά”.

Αντίστοιχα στην επιλογή του εργαλείου διαγραφής πραγματοποιείται η απαραίτητη επικοινωνία με την βάση δεδομένων και πιο συγκεκριμένα τον πίνακα lesson, από τον οποίο και διαγράφεται η εγγραφή των στοιχείων του μαθήματος που

επιλέχθηκε. Αξίζει να σημειωθεί ότι η διαδικασία διαγραφής δεν πραγματοποιείται σε περίπτωση που έχουν προστεθεί τμήματα στο μάθημα αυτό.

Η λειτουργικότητα για την ενημέρωση του μαθήματος υλοποιείται από τον κώδικα που περιέχεται στο αρχείο *edit_lesson.php* ενώ επιπρόσθετα, γίνεται χρήση της συνάρτησης *update_lesson()* που είναι υπεύθυνη για την λειτουργία της ενημέρωσης των δεδομένων στη βάση και της συνάρτησης *delete_lesson()* για την διαγραφή του μαθήματος. Αξίζει να σημειωθεί ότι πραγματοποιείται για άλλη μια φορά η χρήση της συνάρτησης *check_user()*, καθώς αναπόφευκτα δημιουργείται η ανάγκη ελέγχου εγκυρότητας του *username* του καθηγητή στην περίπτωση που αυτό τροποποιηθεί κατά την διαδικασία επεξεργασίας του μαθήματος.

Το δεύτερο panel με τίτλο “Ρυθμίσεις” περιέχει τις εξής λειτουργίες:

- Αλλαγή κωδικού administrator
- Συγχρονισμός με τη βάση LDAP
- Διαγραφή Ανενεργών Φοιτητών

Όπως γίνεται αντιληπτό και από τον τίτλο του, η πρώτη επιλογή προσφέρει την δυνατότητα στον διαχειριστή να αλλάξει τον κωδικό πρόσβασής του. Κατά την ενεργοποίηση της λειτουργίας, στο κεντρικό panel εμφανίζεται η παρακάτω φόρμα αλληλεπίδρασης:

The screenshot shows a web interface for changing a password. At the top, there is a dark header with the text "Αλλαγή κωδικού". Below the header, the main content area has a title "Κωδικός" and a subtitle "Συμπληρώστε τον παλιό κωδικό, δυο φορές τον καινούριο και πατήστε επιβεβαίωση." There are three input fields: "Παλιός Κωδικός", "Καινούριος κωδικός", and "Επαλήθευση καινούριου κωδικού". At the bottom of the form is a button with a green plus sign and the text "Αλλαγή".

Εικόνα 19: Αλλαγή κωδικού πρόσβασης

Μετά τη συμπλήρωση των πεδίων της φόρμας, ο χρήστης επιβεβαιώνει την αλλαγή κωδικού πρόσβασης επιλέγοντας το κουμπί “Αλλαγή” με το οποίο και εκτελείται η διαδικασία αλλαγής του πεδίου password στον πίνακα admin. Η παραπάνω διαδικασία εκτελείται από τον κώδικα του αρχείου *change_pass.php*.

Η δεύτερη λειτουργία προσφέρει τη δυνατότητα στο σύστημα LMS να ελέγχει τη λίστα των φοιτητών που είναι εγγεγραμμένοι στην βάση δεδομένων σε σχέση τις καταχωρίσεις των φοιτητών στη βάση του LDAP server. Η λειτουργία αυτή είναι χρήσιμη στις περιπτώσεις όπου ένας φοιτητής του τμήματος αφαιρείται από την λίστα των φοιτητών του τμήματος στον LDAP server (π.χ. είναι πτυχιούχος) όμως, παρόλα αυτά, είναι ακόμα καταχωρημένος στην τοπική βάση του LMS.

Κατά την επιλογή της λειτουργίας, στο κεντρικό panel της σελίδας, εμφανίζεται το παρακάτω interface:

Συγχρονισμός με τη βάση LDAP

LDAP

Πατήστε το κουμπί για να σβηστούν οι ανενεργοί λογαριασμοί φοιτητών. (Όσες δηλαδή δεν υπάρχουν στον LDAP server)

[Sync](#)

Εικόνα 20: Συγχρονισμός με LDAP server

Εάν ο χρήστης επιλέξει το κουμπί “Sync” τότε, πραγματοποιείται αναζήτηση σε όλες τις εγγραφές του πίνακα student κάθε μία από τις οποίες ελέγχεται για την εγκυρότητά της ως προς τα στοιχεία του LDAP server με τη χρήση της συνάρτησης `check_user()`. Στην περίπτωση που κάποιος φοιτητής δεν περάσει τον έλεγχο εγκυρότητας, τότε τα στοιχεία του εμφανίζονται κάτω από το αρχικό interface όπως φαίνεται στην παρακάτω εικόνα.

Οι παρακάτω φοιτητές δεν είναι πλέον καταχωρημένοι στον LDAP server. Είστε σίγουροι οπ θέλετε να διαγράψετε τους παρακάτω φοιτητές; **Οι φοιτητές θα διαγραφούν ολοκληρωτικά απο το σύστημα!**

#	Username	Όνομα	Επώνυμο
1	vaspap	Βασίλης	Παπαθανασίου

Εικόνα 21: Ανάδραση ελέγχου φοιτητών εκτός του LDAP

Εάν ο χρήστης δώσει θετική απάντηση, τότε με τη χρήση της συνάρτησης `delete_user()` ο εν λόγω φοιτητής διαγράφεται ολοκληρωτικά από τον πίνακα student καθώς και από οποιαδήποτε άλλη εγγραφή άλλου πίνακα στην οποία μπορεί να εμφανίζονται τα στοιχεία του. Η παραπάνω διαδικασία εκτελείται από τον κώδικα που περιέχεται στο αρχείο `sync_db.php` (παράρτημα κώδικα K-1.7).

Τέλος με την επιλογή “Διαγραφή Ανενεργών Φοιτητών” και στην συνέχεια την επιλογή του συνδέσμου “Διαγραφή” στην σελίδα που εμφανίζεται, πραγματοποιείται ένας έλεγχος στην βάση δεδομένων κατά τον οποίο, εξετάζονται περιπτώσεις φοιτητών οι οποίοι δεν είναι εγγεγραμμένοι σε κανένα τμήμα κάποιου μαθήματος. Στις περιπτώσεις αυτές, οι φοιτητές αναγνωρίζονται ως ανενεργοί από το σύστημα οπότε και διαγράφονται δια παντός από τον πίνακα των φοιτητών (student) εάν και εφόσον ο διαχειριστής το επιλέξει. Τα παραπάνω υλοποιούνται με τη χρήση του `delete_studs.php` σε συνδυασμό πάλι με την χρήση της `delete_user()` η οποία αναφέρθηκε παραπάνω. Η διαδικασία αποτυπώνεται στην παρακάτω εικόνα.

Διαγραφή Ανενεργών Φοιτητών

Φοιτητές

Πατήστε τον παρακάτω σύνδεσμο για να σβηστούν οι ανενεργοί λογαριασμοί φοιτητών.
(Δηλαδή όσοι δεν είναι καταχωρημένοι σε κάποιο τμήμα)

[Διαγραφή](#)

Οι παρακάτω φοιτητές δεν είναι πλέον καταχωρημένοι σε κάποιο τμήμα. Είστε σίγουροι οτι θέλετε να διαγράψετε τους παρακάτω φοιτητές; **Οι φοιτητές θα διαγραφούν ολοκληρωτικά απο το σύστημα!**

#	Username	Όνομα	Επώνυμο
1	pantov	Παντελεζήμων	Ντόβας

Εικόνα 22: Διαγραφή φοιτητών

Στο σημείο αυτό, προτού αναλυθεί η σελίδα του καθηγητή, ας αναφερθεί ότι ο admin με δικαιώματα superuser έχει τη δυνατότητα να κάνει αλλαγές στο σύστημα, που αποτελούν υπερσύνολο αυτών που δικαιούται να κάνει ένας καθηγητής. Ενδεικτικά αναφέρεται ότι ο καθηγητής όπως θα αναλυθεί σε επόμενο στάδιο έχει την ικανότητα παραμετροποίησης των μαθημάτων που διδάσκει σε αντίθεση με τον administrator που διαχειρίζεται το σύνολο των μαθημάτων του συστήματος.





3.2.2.5 Η σελίδα του Καθηγητή

Παρόμοια με την σελίδα του διαχειριστή, στην επιτυχημένη είσοδο ενός καθηγητή, γίνεται η χρήση του script `list_lessons.php` ώστε αυτή τη φορά να εμφανιστεί μόνο το σύνολο των μαθημάτων στα οποία ο καθηγητής είναι διδάσκων. Για τον επιτυχή διαχωρισμό των δύο περιπτώσεων στο εν λόγω αρχείο, παρέχεται ένας έλεγχος ροής κατά τον οποίο στην περίπτωση όπου πραγματοποιείται είσοδος ενός καθηγητή, γεγονός το οποίο αναγνωρίζεται από την τιμή της session μεταβλητής `profile`, δημιουργείται ένα διαφορετικό ερώτημα στη βάση δεδομένων από αυτό της περίπτωσης του διαχειριστή στο οποίο, ανασύρονται μόνο οι εγγραφές του πίνακα `lesson` για τις οποίες περιέχεται το username του καθηγητή ως τιμή του πεδίου `teacher_username`. Οι πληροφορίες των μαθημάτων εμφανίζονται με τον γνωστό τρόπο που συναντήσαμε και στην φόρτωση της σελίδας διαχειριστή. Επίσης, αξίζει να σημειωθεί ότι οι επιλογές προσθήκης και διαχείρισης μαθήματος δεν υπάρχουν καθώς ο καθηγητής σαν χρήστης δεν είναι εξουσιοδοτημένος γι' αυτές τις ενέργειες.

Διαθέσιμα Μαθήματα			
Μαθήματα			
Εδώ βρίσκονται τα καταχωρημένα μαθήματα. Πατήστε τον τίτλο του μαθήματος για περισσότερες λειτουργίες			
#	Κωδικός	Τίτλος	Διδάσκων
1	4804	Ειδικά Θέματα Δικτύων 1	psarnik

Εικόνα 23: Φόρμα Διαθέσιμων Μαθημάτων

Κάθε ένας από τους τίτλους των μαθημάτων αποτελεί ένα σύνδεσμο (link) με τον οποίο ο χρήστης μπορεί να έχει περισσότερες επιλογές διαχείρισης για το συγκεκριμένο μάθημα. Κατά την επιλογή του συνδέσμου του μαθήματος πραγματοποιείται η κλήση του αρχείου `edit_sublessons.php` ενώ ταυτόχρονα το id που αντιστοιχεί στο μάθημα αυτό (η τιμή του φαίνεται στη στήλη #) περνάει ως παράμετρος στο URL της κλήσης με τη χρήση της μεθόδου GET. Το αποτέλεσμα της εκτέλεσης φαίνεται στην παρακάτω εικόνα.

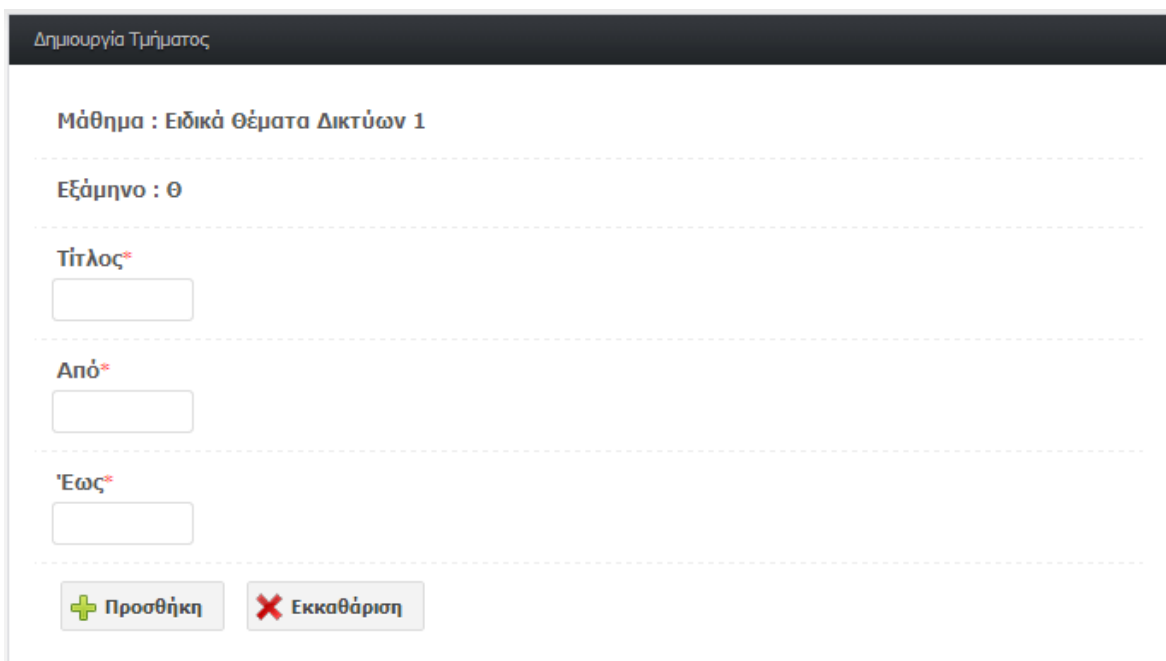
Ειδικά Θέματα Δικτύων 1		Διαθέσιμα Τμήματα			
Τμήματα		Ειδικά Θέματα Δικτύων 1			
Εδώ βρίσκονται τα καταχωρημένα τμήματα του μαθήματος Ειδικά Θέματα Δικτύων 1.					
Τίτλος	Από / Έως	Εγγεγραμμένοι φοιτητές	Ερωτηματολόγια	Επεξεργασία	
T3	2013-06-13 / 2013-06-20	3	1	 	
T1	2013-06-02 / 2013-06-02	0	0	 	

Εικόνα 24: Το περιβάλλον διαχείρισης ενός τμήματος

Στο κεντρικό panel εμφανίζονται, σαν αυτόματη ενέργεια, τα καταχωρημένα τμήματα του μαθήματος που έχει επιλεγεί. Αν το επιλεγμένο μάθημα δεν διαθέτει ακόμα τμήματα τότε ο πίνακας στον οποίο εμφανίζονται τα στοιχεία του τμήματος, όπως αυτά εμφανίζονται στον πίνακα `sublesson`, είναι κενός. Επιπρόσθετα, μαζί με κάθε τμήμα που εμφανίζεται, συμπεριλαμβάνεται ένα ζευγάρι εργαλείων για την επεξεργασία ή την διαγραφή του τμήματος αντίστοιχα. Να σημειωθεί ότι στην

περίπτωση διαγραφής ενός τμήματος, διαγράφονται και οι εγγραφές στον πίνακα `student_sublession` των φοιτητών που σχετίζονται με αυτό το μάθημα. Η λειτουργικότητα των εργαλείων υλοποιείται με τη χρήση του αρχείου `edit_sublession.php` για την λειτουργία της επεξεργασίας καθώς και της συνάρτησης `delete_sublession()` για την λειτουργία της διαγραφής. Επίσης να σημειωθεί ότι ακριβώς την ίδια λειτουργικότητα έχει η επιλογή “Καταχωρημένα Τμήματα”, η οποία βρίσκεται δεύτερη στο πρώτο αριστερό panel που τιτλοφορείται με το όνομα του μαθήματος.

Πρώτη επιλογή του πρώτου panel αποτελεί η “Δημιουργία Τμήματος” η οποία προσφέρει τη δυνατότητα να προστεθεί ένα νέο τμήμα στο επιλεγμένο μάθημα. Κατά την επιλογή αυτή εκτελείται το αρχείο `add_sublession.php` το οποίο εμφανίζει στο κεντρικό panel της σελίδας την παρακάτω φόρμα.



Δημιουργία Τμήματος

Μάθημα : Ειδικά Θέματα Δικτύων 1

Εξάμηνο : Θ

Τίτλος*

Από*

Έως*

+ Προσθήκη X Εκκαθάριση

Εικόνα 25: Η φόρμα δημιουργίας τμήματος

Στην φόρμα αυτή ο χρήστης συμπληρώνει τον τίτλο του τμήματος καθώς και την χρονική διάρκεια λειτουργίας που αυτό θα έχει. Για την επιλογή των ημερομηνιών λειτουργίας του τμήματος γίνεται χρήση της JavaScript συνάρτησης `datepicker`. Τέλος με την ολοκλήρωση συμπλήρωσης των στοιχείων, ο χρήστης μπορεί να επιλέξει “Προσθήκη” ώστε να δημιουργηθεί η εγγραφή του νέου μαθήματος στον πίνακα `sublession` ή εναλλακτικά να καθαρίσει την φόρμα από τα δεδομένα που εισήχθησαν επιλέγοντας “Εκκαθάριση”. Για την δημιουργία της νέας εγγραφής του τμήματος στη βάση δεδομένων χρησιμοποιείται η συνάρτηση `add_sublession()`.

Με την επιλογή “Εγγραφή Φοιτητών” παρέχεται η δυνατότητα της προσθήκης φοιτητών σε ένα από τα τμήματα τα οποία έχουν δημιουργηθεί για το επιλεγμένο

μάθημα. Μετά από την επιλογή του χρήστη, στο κεντρικό panel της σελίδας εμφανίζεται η παρακάτω φόρμα αλληλεπίδρασης.

Εγγραφή φοιτητών

Εγγραφή σε τμήμα

Εισάγετε τις πληροφορίες για το φοιτητή/καθηγητή που θέλετε να προσθέσετε. Εάν εγγραφεί καθηγητής, θα έχει τη δυνατότητα να επισκεφτεί τη σελίδα του φοιτητή.

Μάθημα : Ειδικά Θέματα Δικτύων 1

Τμήμα
T3

Username (Φοιτητή/Καθηγητή)*

Επώνυμο

Όνομα

Εικόνα 26: Προσθήκη Φοιτητή σε τμήμα

Σε αυτήν τη φόρμα ο χρήστης συμπληρώνει τα εξής στοιχεία:

- Σε ποιο από τα διαθέσιμα τμήματα θα προστεθεί ο φοιτητής
- Το username του φοιτητή ή το username ενός καθηγητή ο οποίος επιθυμεί να συμμετάσχει στο τμήμα ως παρατηρητής.
- Το ονοματεπώνυμο του φοιτητή

Μετά την συμπλήρωση της φόρμας ο χρήστης μπορεί να επιβεβαιώσει την προσθήκη του φοιτητή επιλέγοντας “Προσθήκη” ή να την ακυρώσει επιλέγοντας “Εκκαθάριση”. Αξίζει να σημειωθεί ότι κατά την παραπάνω διαδικασία πραγματοποιείται έλεγχος ώστε να αποτρέπεται η προσθήκη ενός φοιτητή σ’ ένα τμήμα στο οποίο έχει ήδη εγγραφεί καθώς και η προσθήκη ενός φοιτητή σε ένα τμήμα ενώ προηγουμένως έχει εγγραφεί σε ένα άλλο τμήμα του ίδιου μαθήματος. Οποιαδήποτε από τις δύο προηγούμενες ενέργειες θα ακυρώσει την διαδικασία εγγραφής, εμφανίζοντας τα εξής μηνύματα που φαίνονται παρακάτω.

⚠ Ο χρήστης aikprot είναι ήδη εγγεγραμμένος στο τμήμα T3.

Εικόνα 27: Ανάδραση ελέγχου εγγραφής φοιτητή

❗ Ο χρήστης aikprot είναι ήδη εγγεγραμμένος σε τμήμα του μαθήματος Ειδικά Θέματα Δικτύων 1.

Εικόνα 28: Ανάδραση ελέγχου εγγραφής φοιτητή σε Μάθημα

Η λειτουργικότητα όλων των παραπάνω επιτυγχάνεται με τη χρήση του αρχείου `add_student.php` σε συνδυασμό με την κλήση της συνάρτησης `enroll_student()` (παράρτημα κώδικα K-1.8 και K-1.9 αντίστοιχα).

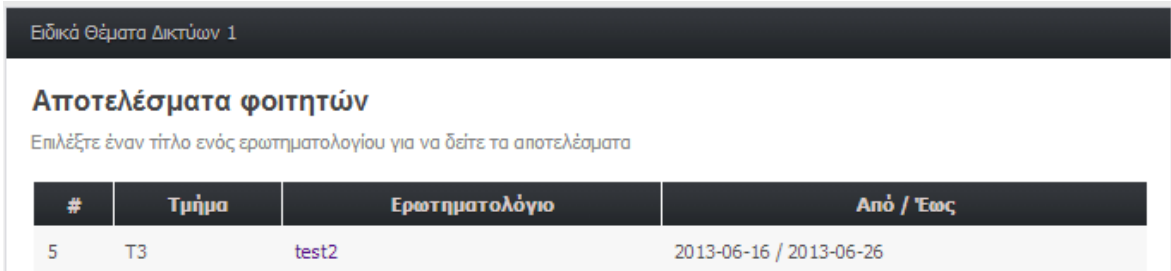
Μετά την προσθήκη ενός ή περισσότερων φοιτητών σε ένα τμήμα, ο χρήστης μπορεί να δει το αποτέλεσμα των ενεργειών αυτών χρησιμοποιώντας την λειτουργία “Εγγεγραμμένοι Φοιτητές”. Επιπρόσθετα η λειτουργία αυτή, για κάθε φοιτητή ο οποίος είναι εγγεγραμμένος στο τμήμα, προσφέρει τη δυνατότητα διαγραφής του από αυτό. Το αποτέλεσμα της λειτουργίας αυτής εμφανίζεται στο κεντρικό panel της σελίδας του καθηγητή και έχει την εξής μορφή:

Εγγεγραμμένοι φοιτητές στο μάθημα Ειδικά Θέματα Δικτύων 1			
Τμήμα	Username	Όνοματεπώνυμο	Διαγραφή
T3	aachpari		✗
T3	adamidis	Μη διαθέσιμο	✗
T3	aikprot		✗

Εικόνα 29: Εγγεγραμμένοι Φοιτητές σε μάθημα

Η παραπάνω λειτουργία υλοποιείται από τον κώδικα που περιέχεται στο αρχείο `add_students.php` σε συνδυασμό με την χρήση της συνάρτησης `delete_student_enrollment()` η οποία είναι υπεύθυνη για την οριστική αφαίρεση της συσχέτισης του φοιτητή από το συγκεκριμένο τμήμα.

Ακόμα μία λειτουργία που υπάρχει στη σελίδα του καθηγητή έχει τίτλο “Αποτελέσματα Φοιτητών” και του επιτρέπει να παρακολουθήσει τα αποτελέσματα εξέτασης των φοιτητών σε ερωτηματολόγια τα οποία εξετάστηκαν . Το αποτέλεσμα εκτέλεσης της συγκεκριμένης λειτουργίας φαίνεται στην παρακάτω εικόνα.



Ειδικά Θέματα Δικτύων 1

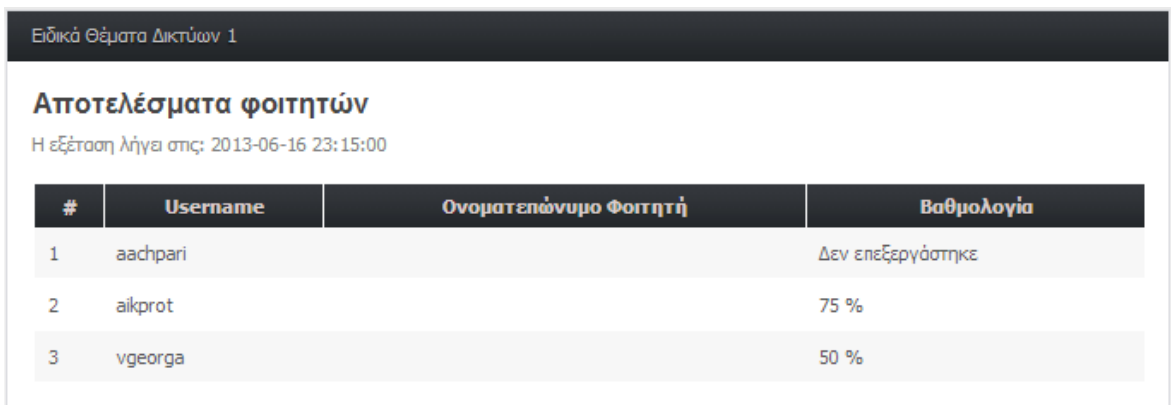
Αποτελέσματα φοιτητών

Επιλέξτε έναν τίτλο ενός ερωτηματολογίου για να δείτε τα αποτελέσματα

#	Τμήμα	Ερωτηματολόγιο	Από / Έως
5	T3	test2	2013-06-16 / 2013-06-26

Εικόνα 30: Το interface των Αποτελεσμάτων Φοιτητών

Ο τίτλος του κάθε ερωτηματολογίου είναι σε μορφή link η οποία, αν ενεργοποιηθεί, εμφανίζει πληροφορίες για τα άτομα που εξετάστηκαν στο ερωτηματολόγιο καθώς και τη βαθμολογία την οποία σημείωσαν σε αυτό.



Ειδικά Θέματα Δικτύων 1

Αποτελέσματα φοιτητών

Η εξέταση λήγα στις: 2013-06-16 23:15:00

#	Username	Ονοματεπώνυμο Φοιτητή	Βαθμολογία
1	aachpari		Δεν επεξεργάστηκε
2	aikprot		75 %
3	vgeorga		50 %

Εικόνα 31: Αποτελέσματα φοιτητών μετά από εξέταση

Η παραπάνω διαδικασία πραγματοποιείται με τη χρήση του κώδικα ο οποίος περιέχεται στο αρχείο `view_results.php`.

Στο δεύτερο panel, συνοψίζονται οι λειτουργίες που αφορούν στη διαχείριση των ερωτηματολογίων. Πρώτο βήμα στη διαδικασία αυτή αποτελεί η λειτουργία “Προσθήκη Ερώτησης” η εκτέλεση της οποίας εμφανίζει το παρακάτω αποτέλεσμα στο κεντρικό panel της σελίδας.

Πτυχιακή εργασία του φοιτητή Γεωργακόπουλου Βασίλη

Προσθήκη Ερώτησης

Μάθημα : Γραφικά Υπολογιστών

Ερώτηση (*)

Βαρύτητα (Μονάδες) (*)

Φωτογραφία (jpg/png/gif μέχρι 2 mb)

Δεν έχει επιλεγεί κανένα αρχείο

Απάντηση (*)

Απάντηση (*)

Εικόνα 32: Η φόρμα για την προσθήκη νέας ερώτησης

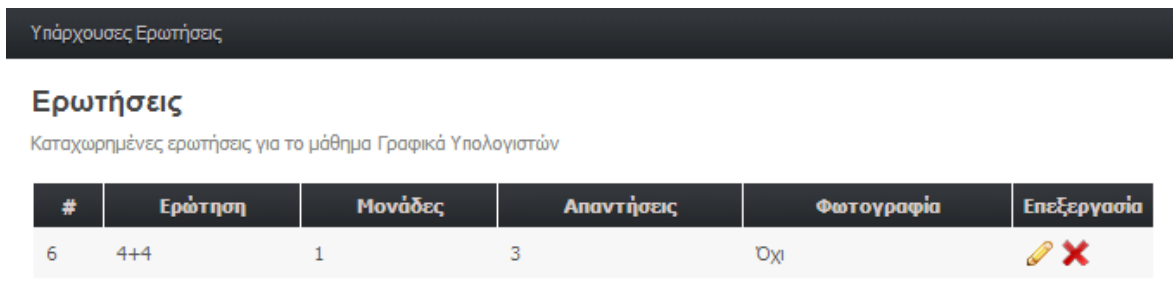
Στη φόρμα που εμφανίζεται ο χρήστης, συμπληρώνει τα εξής στοιχεία



- Το κείμενο της ερώτησης
- Τη βαρύτητα της ερώτησης (προεπιλογή 1)
- Προαιρετικά, μια εικόνα
- Το κείμενο των απαντήσεων καθώς και το χαρακτηρισμό τους με σωστή ή λάθος απάντηση

- Σε περίπτωση που η ερώτηση έχει παραπάνω από δύο απαντήσεις ο χρήστης μπορεί να προσθέσει επιπλέον components απαντήσεων επιλέγοντας “Προσθήκη Απάντησης”,
- Τέλος, ο χρήστης μπορεί να επιβεβαιώσει τη δημιουργία της νέας ερώτησης επιλέγοντας “Προσθήκη” ή διαφορετικά να την ακυρώσει επιλέγοντας “Εκκαθάριση”

Η λειτουργία των παραπάνω, υλοποιείται από τον κώδικα που περιέχεται στο αρχείο *add_question.php* σε συνδυασμό με τη συνάρτηση *add_question()*, η οποία είναι υπεύθυνη για την εγγραφή των δεδομένων της ερώτησης και των απαντήσεων στους πίνακες *question* και *answer* αντίστοιχα. Τα παραπάνω βρίσκονται στο παράρτημα κώδικα K-1.10 και K-1.11 αντίστοιχα.

Μετά την προσθήκη ερωτήσεων, ο χρήστης μπορεί να δει το αποτέλεσμα των ενεργειών του επιλέγοντας τη λειτουργία “Καταχωρημένες Ερωτήσεις”, η οποία εμφανίζει το παρακάτω περιεχόμενο στο κεντρικό panel της σελίδας.



#	Ερώτηση	Μονάδες	Απαντήσεις	Φωτογραφία	Επεξεργασία
6	4+4	1	3	Όχι	 

Εικόνα 33: Ερωτήσεις ερωτηματολογίου και επεξεργασία

Για κάθε μία ερώτηση που εμφανίζεται στη λίστα, παρατίθεται ένα ζευγάρι εργαλείων επεξεργασίας και διαγραφής της συγκεκριμένης ερώτησης, αντίστοιχα.

Οι παραπάνω λειτουργίες αντιστοιχίζονται στα αρχεία *list_questions.php* και *edit_question.php* για την επεξεργασία και την ενημέρωση της ερώτησης καθώς και τη χρήση της συνάρτησης *delete_question()* για την διαγραφή της ερώτησης, μαζί με τις απαντήσεις που σχετίζονται με αυτή, από τη βάση δεδομένων.

Μετά την ολοκλήρωση της προσθήκης και παραμετροποίησης των ερωτήσεων, ο χρήστης έχει τη δυνατότητα να δημιουργήσει το δικό του ερωτηματολόγιο. Αυτό επιτυγχάνεται με το πάτημα του κουμπιού “Προσθήκη Ερωτηματολογίου” που εμφανίζει την εξής φόρμα.

Προσθήκη Ερωτηματολογίου

Ερωτηματολόγιο

Προσθήκη ερωτήσεων απο τον παρακάτω πίνακα.

Μάθημα : Γραφικά Υπολογιστών

Τμήμα (*) T4

Τίτλος (Περιγραφή) (*)

Ημερομηνία έναρξης (*)

Ημερομηνία λήξης (*)

Διάρκεια (λεπτά) (*)

+ Προσθήκη

- Εκκαθάριση

Επιλογή Ερωτήσεων

Υπάρχουσες Ερωτήσεις

Επιλογή τυχαίων απαντήσεων (αριθμός) (*) +

#	Ερώτηση	Μονάδες	Απαντήσεις	Φωτογραφία	Προσθήκη
6	4+4	1	3	Όχι	<input type="checkbox"/>

Εικόνα 34: Η φόρμα δημιουργίας ερωτηματολογίου

Στη φωτογραφία αποτυπώνονται οι εξής λειτουργίες που πλαισιώνουν την προσθήκη ενός ερωτηματολογίου.

- **Τμήμα:** Αναφέρεται στο τμήμα για το οποίο το ερωτηματολόγιο θα είναι ενεργό
- **Τίτλος:** Πρόκειται για το όνομα του ερωτηματολογίου
- **Ημ. Έναρξης:** Εμφανίζει σε ημερολογιακή μορφή το μενού από το οποίο ο χρήστης θα προσδιορίσει την ημερομηνία έναρξης.
- **Ημ. Λήξης:** Όπως παραπάνω, ο χρήστης στο σημείο αυτό προσδιορίζει την ημερομηνία (ώρα) λήξης του ερωτηματολογίου.

- **Διάρκεια:** Πρόκειται για το χρόνο περάτωσης του ερωτηματολογίου, μετρημένο σε λεπτά.





Μετά την βασική αυτή παραμετροποίηση του ερωτηματολογίου, κρίνεται απαραίτητο ο χρήστης να επιλέξει το σύνολο των ερωτήσεων που θα το απαρτίζουν. Οι ερωτήσεις εμφανίζονται στο κάτω μέρος της φόρμας, δίνοντας στον χειριστή την ικανότητα να προσθέσει είτε χειροκίνητα είτε τυχαία είτε με συνδυασμό των δύο τρόπων τις ερωτήσεις.

Σημειώνεται η ύπαρξη δικλίδας ασφάλειας του συστήματος που απαγορεύει την δημιουργία ερωτηματολογίου που παρουσιάζει έλλειψη ερωτήσεων ή χρόνου διάρκειας.

Προκειμένου να αντιληφθεί κανείς τον τρόπο υλοποίησης της παραπάνω σύνθετης διαδικασίας, παρατίθενται τα εξής:

- Η συνάρτηση `add_quiz()` είναι υπεύθυνη για την καταχώρηση των δεδομένων του ερωτηματολογίου στη βάση δεδομένων (παράρτημα κώδικα K-1.12)
- Το αρχείο κώδικα `add_quiz.php` είναι υπεύθυνο για την εμφάνιση της αλληλεπιδραστικής φόρμας και της λειτουργικότητας της (παράρτημα K-1.13)

Συνεχίζοντας την περιγραφή της λειτουργικότητας της σελίδας του καθηγητή, μένει να περιγραφεί η επιλογή “Καταχωρημένα Ερωτηματολόγια” που εμφανίζει το παρακάτω αποτέλεσμα μετά από την ανάδραση του χρήστη. Όπως και στην περίπτωση των κατοχυρωμένων ερωτήσεων, στη φόρμα εμφανίζεται η λίστα των δημιουργημένων ερωτηματολογίων, με ταυτόχρονη δυνατότητα επεξεργασίας αλλά και διαγραφής τους.

#	Ερωτηματολόγιο	Τμήμα	Από / Έως	Αριθμός Ερωτήσεων	Επεξεργασία
4	εξέταση1	T4	2013-07-03 / 2013-07-11	1	 
1	test11	T4	2013-06-17 / 2013-07-01	1	 

Εικόνα 35: Καταχωρημένα ερωτηματολόγια

Ο κώδικας που υλοποιεί την παραπάνω διαδικασία συνοψίζεται στα:

- `list_quiz.php`: που εμφανίζει τη φόρμα με τα καταχωρημένα ερωτηματολόγια
- `edit_quiz.php`: που εμφανίζει τη φόρμα επεξεργασίας ενός ερωτηματολογίου

- `update_quiz()`: που ενημερώνει τη βάση δεδομένων με τυχόν αλλαγές που έχουν προκύψει από την παραπάνω διαδικασία
- `delete_quiz()`: που διαγράφει ένα ερωτηματολόγιο από τη βάση δεδομένων.

3.2.2.6 Η σελίδα του Φοιτητή

Αφού πραγματοποιηθεί είσοδος του χρήστη με δικαιώματα φοιτητή, οι φόρμες στις οποίες έχει πρόσβαση είναι εκείνες των ερωτηματολογίων που του έχουν ανατεθεί (“Διαθέσιμα Ερωτηματολόγια”) και των βαθμολογιών (“Βαθμολογίες”).

Παρακάτω αναλύονται οι εν λόγω φόρμες.

Μάθημα	Τμήμα	Ερωτηματολόγιο	Από / Έως
Ψηφιακά Συστήματα	RWERWE	εξέταση2	2013-06-17 19:04 / 2013-07-01 00:00

Εικόνα 36: Ερωτηματολόγιο προς εξέταση

Στην παραπάνω εικόνα, απεικονίζεται το ερωτηματολόγιο που είναι διαθέσιμο στον φοιτητή. Με την επιλογή του ερωτηματολογίου, στην συγκεκριμένη περίπτωση του ερωτηματολογίου “εξέταση2”, η επόμενη φόρμα που εμφανίζεται στο σύστημα είναι:

Θέλετε να ξεκινήσετε την εξέταση; Θα έχετε διαθέσιμα 15 λεπτά!

Εικόνα 37: Διεξαγωγή εξέτασης

Διεξαγωγή Εξέτασης

εξέταση2

Επιλέξτε τις σωστές απαντήσεις και πατήστε Υποβολή.

1. Γιατι η γη γυρίζει

λόγω του μαγνητικού πεδίου

γιατί έτσι

+ Υποβολή

Απομένουν:
14 λεπτά 57 δευτερόλεπτα

Εικόνα 38: Φόρμα Εξέτασης

Η λειτουργικότητα της φόρμας συνοψίζεται στην ενημέρωση του φοιτητή για την εκκίνηση της εξέτασης που θα διαρκέσει ακριβώς 15 λεπτά και θα υποβληθεί αυτόματα με το πέρας του χρόνου αυτού. Σε θετική απόκριση του φοιτητή, το σύστημα προβάλλει το ερωτηματολόγιο μαζί με το χρονόμετρο που του υπενθυμίζει τον εναπομείναντα χρόνο (Εικόνα 38). Σημειώνεται ότι ο χρήστης δεν έχει το δικαίωμα για τον οποιονδήποτε λόγο να ξεπεράσει ή να σταματήσει το χρονόμετρο ακόμα και αν εξέλθει από τη σελίδα εμφάνισης του ερωτηματολογίου ή γενικότερα σε περίπτωση αποσύνδεσης από το σύστημα.

Αφού λοιπόν ο φοιτητής συμμετάσχει επιτυχώς ή όχι στην εξέταση και αφού αυτή ολοκληρωθεί, του δίνεται η δυνατότητα επισκόπησης των αποτελεσμάτων του με την εμφάνιση -ύστερα από επιλογή- της λειτουργίας “Βαθμολογίες” όπως φαίνεται στην επόμενη εικόνα. Εκεί ενημερώνεται για το αποτέλεσμα της εξέτασης σε κάθε ένα από τα ερωτηματολόγια που του ανατέθηκαν.

Βαθμολογίες

Αποτελέσματα ολοκληρωμένων ερωτηματολογίων

Μάθημα	Ημερομηνία	Ερωτηματολόγιο	Αποτέλεσμα
Γραφικά Υπολογιστών	2013-06-17	εξέταση1	0 %
Γραφικά Υπολογιστών	2013-06-17	test11	100 %

Εικόνα 39: Φόρμα Βαθμολογιών

Οι παραπάνω λειτουργίες, με σειρά εμφάνισής τους στις εικόνες της ενότητας, υλοποιούνται ως εξής:

- Το αρχείο κώδικα student.php είναι υπεύθυνο για την εμφάνιση των στοιχείων της αρχικής σελίδας του φοιτητή

- Με την επιλογή εξέτασης σε ένα ερωτηματολόγιο ενεργοποιείται το αρχείο *do_quiz.php* (παράρτημα K-1.14) το οποίο λαμβάνει παραμετρικά το αναγνωριστικό *id* του ερωτηματολογίου κατά την επιλογή του ερωτηματολογίου με τη χρήση της μεθόδου GET
- Μετά το πέρας της εξέτασης, καλείται το αρχείο *quiz_results.php* το οποίο εμφανίζει τη βαθμολογία που σημείωσε ο φοιτητής.
- Τέλος το αρχείο *result_history.php* είναι υπεύθυνο για την εμφάνιση του ιστορικού των βαθμολογιών του φοιτητή στα ερωτηματολόγια τα οποία του έχουν ανατεθεί

3.2.2.7 Βελτιώσεις του συστήματος

Σε αυτήν την ενότητα θα γίνει μία αναφορά στα προβλήματα τα οποία προέκυψαν κατά την ανάπτυξη του συστήματος, ενώ παράλληλα θα περιγράφει και ο τρόπος με τον οποίο αυτά επιλύθηκαν.

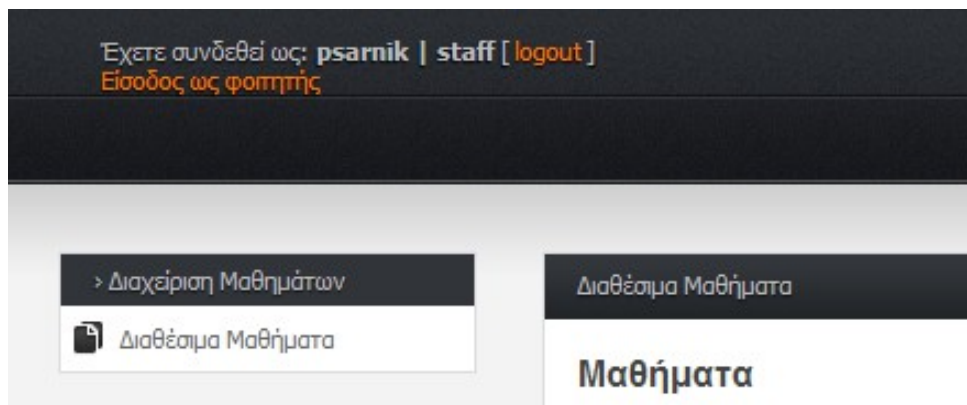
Όπως αναφέρθηκε σε προηγούμενη ενότητα, το σύστημα παρέχει την δυνατότητα σε χρήστες με δικαιώματα διαχειριστή καθώς και καθηγητή, να μπορούν να εγγραφούν ένα καθηγητή στο τμήμα ενός μαθήματος ώστε ο τελευταίος να έχει το δικαίωμα συμμετοχής στα δρώμενα του τμήματος αυτού (λειτουργία “Εγγραφή Φοιτητών”). Όμως η λειτουργία αυτή, πέρα από την λειτουργικότητα την οποία προσέδωσε, παράλληλα δημιούργησε το εξής πρόβλημα: Το σύστημα αδυνατούσε να διαχειριστεί την κατάσταση στην οποία ένας καθηγητής συμμετείχε ως παρατηρητής σε ένα τμήμα ενός μαθήματος ενώ παράλληλα αποτελούσε διδάσκοντα καθηγητή ενός μαθήματος.

Για την επίλυση λοιπόν του παραπάνω προβλήματος, προστέθηκε ένας ακόμη έλεγχος ροής στην περίπτωση εισόδου ενός καθηγητή στο σύστημα, κατά τον οποίο εξετάζεται κατά πόσο υπάρχει το *username* του συγκεκριμένου καθηγητή καταχωρημένο σε εγγραφές τμήματος ενός μαθήματος στη βάση δεδομένων. Στην περίπτωση όπου ο καθηγητής βρεθεί να ανήκει σε κάποια εγγραφή τότε εκτελείται το νεοσυσταθέν αρχείο *stud_or_staff.php* το οποίο εμφανίζει στον χρήστη το εξής Interface:

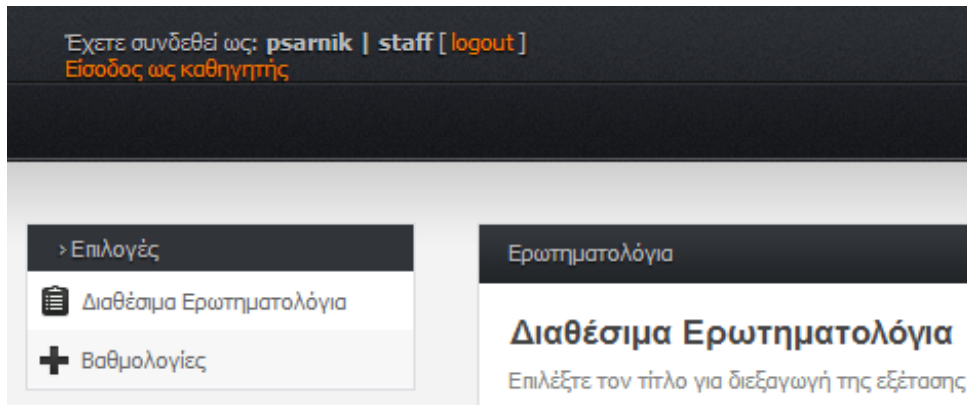


Εικόνα 40: Βελτιστοποίηση σελίδας Login

Εάν ο χρήστης επιλέξει να εισέλθει ως καθηγητής, τότε αυτόματα μεταφέρεται στην γνωστή αρχική σελίδα καθηγητή ενώ στην αντίθετη περίπτωση, εμφανίζεται σε αυτόν το περιβάλλον της αρχικής σελίδας φοιτητή, για το μάθημα στο οποίο έχει εγγραφεί ως παρατηρητής. Αυτό που αξίζει να σημειωθεί και το οποίο φαίνεται και στις παρακάτω εικόνες είναι ότι ο καθηγητής, ανεξάρτητα από την αρχική του επιλογή στην προηγούμενη σελίδα, μπορεί πάντοτε να αλλάξει περιβάλλον εργασίας μέσω των κατάλληλων συνδέσμων μετάβασης η οποίοι εμφανίζονται στην πάνω αριστερά γωνία του παραθύρου χρήστη, κάτω από τα στοιχεία σύνδεσης. Ο κώδικας βρίσκεται στο παράρτημα Κ-1.15.



Εικόνα 41: Είσοδος ως καθηγητής με επιλογή μετάβασης στο περιβάλλον εργασίας του φοιτητή



Εικόνα 42: Είσοδος ως φοιτητής με επιλογή μετάβασης στο περιβάλλον εργασίας του καθηγητή



Ένα ακόμη πρόβλημα το οποίο προέκυψε ήταν ότι ο αρχικός σχεδιασμός του συστήματος δεν παρείχε κάλυψη στην περίπτωση όπου, τα τμήματα ενός μαθήματος θα έπρεπε να διαχειριστούν από παραπάνω από έναν καθηγητές. Για την αντιμετώπιση του συγκεκριμένου προβλήματος πραγματοποιήθηκαν οι εξής ενέργειες:

Το πρώτο βήμα ήταν η απαραίτητη αλλαγή στην βάση δεδομένων ώστε να είναι σε θέση να συντηρήσει αυτήν την πληροφορία. Για τον σκοπό αυτό λοιπόν προστέθηκε το πεδίο *teacher_username* στον πίνακα *sublesson*, με το οποίο πλέον ορίζεται η συσχέτιση του υπεύθυνου καθηγητή για το συγκεκριμένο τμήμα. Επιπλέον, ο κώδικας του αρχείου *view_sublessons.php* εμπλουτίστηκε με έναν επιπλέον έλεγχο ροής ώστε να εμφανίζει, ανάλογα με το ποιος καθηγητής εισέρχεται στο σύστημα, μόνο τα δεδομένα που αφορούν στο τμήμα το οποίο ο ίδιος διαχειρίζεται. Τα αποτελέσματα αυτής της αλλαγής μπορούν να γίνουν αντιληπτά από την παρατήρηση των παρακάτω εικόνων. Στην πρώτη εικόνα φαίνεται το σύνολο των καταχωρημένων τμημάτων του μαθήματος Δίκτυα Η/Υ καθώς και τον διδάσκοντα καθηγητή για το κάθε ένα. Στις επόμενες δύο εικόνες φαίνονται τα καταχωρημένα τμήματα του μαθήματος στην περίπτωση εισόδου του κάθε ενός καθηγητή αντίστοιχα.



Πτυχιακή εργασία του φοιτητή Γεωργακόπουλου Βασίλη

Διαθέσιμα Τμήματα					
Δίκτυα Η/Υ					
Τμήματα					
Εδώ βρίσκονται τα καταχωρημένα τμήματα του μαθήματος Δίκτυα Η/Υ.					
Τίτλος	Διδάσκων	Από / Έως	Εγγεγραμμένοι φοιτητές	Ερωτηματολόγια	Επεξεργασία
T2	peris	2013-06-20 / 2013-06-27	0	0	 
T1	psarnik	2013-06-20 / 2013-06-27	2	0	 

Εικόνα 43: Το σύνολο των καταχωρημένων τμημάτων

Διαθέσιμα Τμήματα					
Δίκτυα Η/Υ					
Τμήματα					
Εδώ βρίσκονται τα καταχωρημένα τμήματα του μαθήματος Δίκτυα Η/Υ.					
Τίτλος	Διδάσκων	Από / Έως	Εγγεγραμμένοι φοιτητές	Ερωτηματολόγια	Επεξεργασία
T1	psarnik	2013-06-20 / 2013-06-27	2	0	 

Εικόνα 44: Καταχωρημένα τμήματα του καθηγητή με username psarnik

Διαθέσιμα Τμήματα					
Δίκτυα Η/Υ					
Τμήματα					
Εδώ βρίσκονται τα καταχωρημένα τμήματα του μαθήματος Δίκτυα Η/Υ.					
Τίτλος	Διδάσκων	Από / Έως	Εγγεγραμμένοι φοιτητές	Ερωτηματολόγια	Επεξεργασία
T2	peris	2013-06-20 / 2013-06-27	0	0	 

Εικόνα 45: Καταχωρημένα τμήματα καθηγητή με username peris

ΣΥΜΠΕΡΑΣΜΑΤΑ

Ολοκληρώνοντας τη συγγραφή του εγγράφου της πτυχιακής εργασίας, θα παρατεθούν ορισμένες τελικές σκέψεις και εντυπώσεις οι οποίες προέκυψαν στα πλαίσια της ανάπτυξης του συστήματος διαχείρισης μάθησης.

Αναφορικά με τα βήματα τα οποία ακολουθήθηκαν καθώς και των εργαλείων τα οποία χρησιμοποιήθηκαν, μπορούν να εξαχθούν τα παρακάτω συμπεράσματα:

Η δημιουργία της βάσης δεδομένων αποδείχτηκε μία αρκετά απαιτητική διαδικασία καθώς, παρά την προσπάθεια αποτύπωσης όλων των οντοτήτων που αλληλεπιδρούν με το σύστημα καθώς και των μεταξύ τους σχέσεων, το ενδεχόμενο να παραλειφθεί μια πληροφορία που αφορά τα παραπάνω ή να εμφανιστεί μία νέα, αποτελεί μία υπαρκτή πιθανότητα σε μια διαδικασία όπως αυτή της σχεδίασης ενός ολόκληρου συστήματος. Ενδεικτικά θα αναφερθεί η περίπτωση της βελτίωσης της βάσης δεδομένων ώστε να καλυφθεί η περίπτωση της διαχείρισης ενός μαθήματος από παραπάνω από έναν καθηγητές, ενδεχόμενο το οποίο δεν είχε ληφθεί υπ όψιν στον αρχικό σχεδιασμό. Αντιθέτως, στην διαδικασία προγραμματισμού του συστήματος, η γλώσσα php αποδείχτηκε ένα εύχρηστο και εύρωστο εργαλείο, αφού παρείχε ένα απλό και κατανοητό συντακτικό καθώς και ένα ευέλικτο σύστημα διαχείρισης μεταβλητών, στοιχεία που διευκόλυναν σε ένα πολύ μεγάλο βαθμό τις διαδικασίες εκμάθησης και χρήσης της γλώσσας. Επιπρόσθετα, το γεγονός ότι αποτελεί μία εξαιρετικά δημοφιλή γλώσσα, συνετέλεσε στο να βρεθούν εύκολα απαντήσεις σε προβλήματα σχεδιασμού και κατανόησης από πολυάριθμα επίσημα και ανεπίσημα forum συζητήσεων τα οποία αποτελούν ισχυρές πηγές γνώσης και τεκμηρίωσης της php.

Σε γενικές γραμμές και όντας πλέον σε αυτό το σημείο, είναι εύκολα κατανοητό να αναγνωρισθεί ότι το σύνολο των εργαλείων που χρησιμοποιήθηκαν στην διαδικασία ανάπτυξης της εφαρμογής αποδείχθηκαν αντάξια του ρόλου αυτού, γεγονός το οποίο προκύπτει από το ότι με τη χρήση τους ήταν εφικτή η παράδοση ενός πλήρους και λειτουργικού συστήματος το οποίο κατάφερε να καλύψει όλους τους δοθέντες στόχους.

Σχετικά τώρα με το σύστημα διαχείρισης μάθησης στην ολότητα του μπορούν να αναφερθούν τα παρακάτω:

Το σύστημα το οποίο αναπτύχθηκε στα πλαίσια της εν λόγω πτυχιακής εργασίας, αποτελεί μία μινιμαλιστική προσέγγιση στην υλοποίηση ενός συστήματος διαχείρισης μάθησης καθώς, αντίστοιχα συστήματα τα οποία έχουν αναπτυχθεί προσφέρουν ένα πολύ μεγαλύτερο εύρος λειτουργιών, τις οποίες αυτή τη στιγμή το ανεπτυχθέν σύστημα δεν μπορεί να καλύψει. Παρόλα αυτά όμως, αξίζει να σημειωθεί ότι ενώ τα υπόλοιπα συστήματα LMS είναι προκατασκευασμένα και

γενικού σκοπού, το ανεπτυχθέν μας σύστημα διαθέτει το πλεονέκτημα ότι σχεδιάστηκε αποκλειστικά πάνω στις ανάγκες λειτουργικότητας του τμήματος Πληροφορικής. Το γεγονός αυτό, προσδίδει ένα επίπεδο εξειδίκευσης στην υλοποίηση του συστήματος το οποίο τα υπόλοιπα συστήματα δε μπορούν να προσφέρουν τόσο εύκολα. Ενδεικτικά θα αναφερθεί ότι το σύστημα ενσωματώνει τη χρήση του ακαδημαϊκού συστήματος LDAP, στο οποίο κρατούνται αποθηκευμένοι οι φοιτητές και οι καθηγητές του τμήματος Πληροφορικής, ώστε να πραγματοποιήσει τον έλεγχο εισόδου χρηστών. Η λειτουργία αυτή δεν υπάρχει εξορισμού σε άλλα συστήματα διαχείρισης μάθησης τα οποία συνήθως χρησιμοποιούν μία ανεξάρτητη βάση δεδομένων για την καταχώρηση των χρηστών. Παράλληλα, το γεγονός ότι το σύστημα αναπτύχθηκε με χρήση γλώσσας προγραμματισμού η οποία επιτρέπει την διαρθρωτή ανάπτυξή του, το σύστημα μπορεί να εμπλουτιστεί με μελλοντικές επεκτάσεις όπως είναι, παραδείγματος χάριν, η προσθήκη μαθησιακού υλικού και λειτουργίες επικοινωνίας (chat, e-mail) δίνοντας έτσι την δυνατότητα να επιτευχθεί ένα επίπεδο λειτουργικότητας, αν μη τι άλλο, αντάξιο με αυτό των ανταγωνιστικών συστημάτων.

ΑΝΑΦΟΡΕΣ

1. Scholer M. (1983), La technologie de l'éducation. Concept, bases et application, Montréal, Presses de l'Université du Québec.
2. Ruth C. Clark, Richard E. Mayer, e-Learning and the Science of Instruction: Proven Guidelines for Consumers and Designers of Multimedia Learning(2011), pp 8.
3. Szabo, M., & Flesher, K. (2002). CMI theory and practice: Historical roots of learning management systems. Paper presented at the E-Learn 2002 World Conference on E-Learning in Corporate, Government, Healthcare, & Higher Education, Montreal, Canada.
4. Hamish Coates, Richard James & Gabrielle Baldwin (2005): A critical examination of the effects of learning management systems on university teaching and learning, Tertiary Education and Management, 11:1, pp 19.
5. P. Baumgartner, H.Hafele, and K.H.Maier, E-Learning Praxishandbuch -Auswahl Von Lernplattformen. Studienverlag, Innsbruck, 2002
6. McCormack, C., & Jones, J. D. (1997). Building a Web-based Education System, New York: Wiley Computer Publishing.
7. Learning Circuits. (2005). A field guide to learning management systems. Retrieved November 12, 2006, from http://www.learningcircuits.org/NR/rdonlyres/BFEC9F41-66C2-42EF-BE9DE4FA0D3CE1CE/7304/LMS_fieldguide1.pdf
8. Wikipedia, Aviation Industry Computer-Based Training Committee, http://en.wikipedia.org/wiki/Aviation_Industry_Computer-Based_Training_Committee, visited on 14/3/2013
9. Stuart Campell, AICC and the LMS, <http://www.syberworks.com/articles/aicc-and-the-lms-article.htm>, visited on 15/3/2013
10. Wikipedia, IMS Learning Design, http://en.wikipedia.org/wiki/IMS_Learning_Design , visited on 16/3/2013
- A1. Jeffery, A., & Currier, S. (6-9-2005). What Is... IMS Learning Design, CETIS
- A2. Ramakrishnan, R., & Gehrke, J. (2002). Συστήματα Διαχείρισης Βάσεων Δεδομένων, Εκδόσεις Τζιόλα, Τόμος Α, pp. 30-37

ΠΑΡΑΡΤΗΜΑ ΚΩΔΙΚΑ

Κ-1.1 Κώδικας login.php

```
<?
include('connect.php');
include('functions.php');
session_start();
if($_SESSION['logged_in']==1)
    header("Location: index.php");
//verify user login
if(isset($_POST['login_submit']))
{
    //admin logs in with local verification
    if($_POST['username']=='admin')
    {
        $sql_v = "SELECT * FROM admin WHERE username = 'admin' AND password =
        '".md5($_POST['pass'])."'";
        $res_sql = mysqli_query($mysql_link,$sql_v);
        if(mysqli_num_rows($res_sql)>0)
        {
            $_SESSION['logged_in'] = 1;
            $_SESSION['username'] = 'admin';
            $_SESSION['profile'] = 'admin';
            header("Location: index.php");
        }
    }
    else
        $login_problem = 1;
```

```
}  
else  
{  
    //check user through LDAP returns 0 for error and stud/staff on success  
    $check = verify_user($_POST['username'],$_POST['pass']);  
    if(strcmp($check, "staff")==0 || strcmp($check, "stud")==0)  
    {  
        $_SESSION['logged_in'] = 1;  
        $_SESSION['username'] = $_POST['username'];  
        //set user profile.student gets redirected to his page  
        if(strcmp($check, "stud")==0)  
        {  
            $_SESSION['profile'] = "student";  
            header("Location: student.php");  
        }  
        elseif(strcmp($check, "staff")==0)  
        {  
            $_SESSION['profile'] = "staff";  
            //check if staff is enrolled in any sublessons  
            $sql_check_staff = "SELECT * FROM student_sublesson WHERE student_id = '".  
$_SESSION['username']."'";  
            $res_sql_check_staff = mysqli_query($mysql_link,$sql_check_staff);  
            if(mysqli_num_rows($res_sql_check_staff)>0)  
            {  
                $_SESSION['stud_or_staff'] = 1;  
                header("Location: stud_or_staff.php");  
            }  
        }  
    }  
}
```



```
        else
            header("Location: lesson.php");
        }
    }
    else
        $login_problem = 1;
    }
}
?>

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

<html xmlns="http://www.w3.org/1999/xhtml" lang="pl" xml:lang="pl">
<head>
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<title>SimpleAdmin</title>
<link rel="stylesheet" type="text/css" href="css/login.css" media="screen" />
</head>
<body>
<div class="wrap">
    <div id="content">
        <div id="main">
            <div class="full_w">
                
                <?
                if($login_problem == 1)
                    echo "<span style='color:red;padding-left: 120px;='>Wrong
username/password</span>";
```

```
?>

    <form action="#" method="post">
        <label for="login">Username:</label>
        <input id="login" name="username" class="text" />
        <label for="pass">Password:</label>
        <input id="pass" name="pass" type="password"
class="text" />
        <div class="sep"></div>
        <button type="submit" class="ok"
name="login_submit">Login</button>
    </form>
</div>
<div class="footer">&raquo; <a href="">http://teithess.gr</a> | Tei
LMS</div>
</div>
</div>
</div>
</body>
</html>
```

K-1.2 Κώδικας Connect.php

```
<?php
//ini_set("display_errors",1);
//mysql connection

$link = mysqli_connect('localhost','root','**password**','teilms');

mysqli_query($link,'SET CHARACTER SET UTF8');

mysqli_query($link,'SET NAMES utf8');

?>
```

K-1.3 Κώδικας list_lessons.php

```
<div id="main">
    <div class="full_w">
        <div class="h_title">Διαθέσιμα Μαθήματα</div>
    <h2>Μαθήματα</h2>
    <p>Εδώ βρίσκονται τα καταχωρημένα μαθήματα. Πατήστε τον
    τίτλο του μαθήματος για περισσότερες λειτουργίες </p>
    <div class="entry">
        <div class="sep"></div>
    </div>
    <table>
        <thead>
            <tr>
                <th scope="col">Κωδικός</th>
                <th scope="col">Τίτλος</th>
                <!--<th scope="col">Εξάμηνο</th-->
                <th scope="col">Διδάσκων</th>
            </tr>
        </thead>
        <tbody>
            <?
            if($_SESSION['username']=='admin')
                $sql = "SELECT * FROM lesson";
            else
                $sql = "SELECT * FROM lesson WHERE teacher_username LIKE '%" .
$_SESSION['username']."%'";
```

Πτυχιακή εργασία του φοιτητή Γεωργακόπουλου Βασίλη

```
$res_sql = mysqli_query($mysql_link,$sql);
while($lesson = mysqli_fetch_array($res_sql))
{
    ?>
    <tr>
        <td class="align-center"><?=$lesson['tei_id']?></td>
        <td><a href="lesson.php?
action=view_sublessons&id=<?=$lesson['id']?>"><?=$lesson['title']?></a></td>
        <!--<td><?=$lesson['semester']?></td-->
        <td><?=$str_replace(";",",",
$lesson['teacher_username'])?></td>
    </tr>
    <?
}
?>
</tbody>
</table>
</div>
</div>
<div class="clear"></div>
</div>
```

K-1.4 Κώδικας add_lesson.php

```
<div id="main">
    <div class="full_w">
        <?
        if($_GET['action']=='add_lesson')
```

Πτυχιακή εργασία του φοιτητή Γεωργακόπουλου Βασίλη

```
{
?>
<div class="h_title">Προσθήκη Μαθήματος</div>
<form action="#" method="post">
  <?
  if(isset($_POST['add_lesson']))
  {
    if($_POST['name']!="")
    {
      $flag = add_lesson($_POST);
      if($flag==1)
      {
        ?>
        <div class="η_ok">
          <p>Το μάθημα <b><?=$_POST['name']?></b> προστέθηκε επιτυχώς.
          Επιστροφή στα <a href="lesson.php">Μαθήματα</a>
          </p>
        </div>
        <?
      }
    }
    elseif($flag==0)
    {
      ?>
      <div class="η_error"><p>Υπήρξε κάποιο σφάλμα κατα την ενημέρωση της
      βάσης δεδομένων!</p></div>
      <?
    }
  }
}
```

```
elseif($flag[0]==2)
{
  ?>
  <div class="n_error"><p>Μη έγκυρο username καθηγητή <b><?
=substr($flag,1)?></b>!</p></div>
  <?
}
elseif($flag==3)
{
  ?>
  <div class="n_error"><p>Υπάρχει ήδη καταχωρημένο μάθημα με τον ίδιο
τίτλο/κωδικό μαθήματος!Παρακαλώ εισάγετε διαφορετικά στοιχεία.</p></div>
  <?
}
elseif($flag==4)
{
  ?>
  <div class="n_error"><p>Δεν υπάρχει δυνατότητα καταχώρησης μαθήματος
χωρίς εξάμηνο.Παρακαλώ επιλέξτε ένα εξάμηνο.</p></div>
  <?
}
}
else
{
  ?>
  <div class="n_error"><p>Παρακαλώ εισάγετε τίτλο μαθήματος.</p></div>
  <?
```

```
}  
}  
?>  
  
<div class="element">  
    <label for="name">Κωδικός Μαθήματος <span  
class="red">*</span></label>  
    <input id="name" name="tei_id" class="text"/>  
</div>  
  
<div class="element">  
    <label for="name">Τίτλος <span class="red">*</span></label>  
    <input id="name" name="name" class="text" />  
</div>  
  
<div class="element">  
    <label for="name">Καθηγητές (username) <span class="red">*</span>  
    <br /><span>Διαχωρίστε τα username των καθηγητών με ";" (πχ  
user1;user2;user3)</span></label>  
    <input id="name" name="teacher_username" class="text" />  
</div>  
  
<div class="element">  
    <label for="content">Εξάμηνο<span class="red">*</span></label>  
    <select name="semester">  
    <option value="-1" selected>Επιλογή</option>  
  
    <option value="A">A</option>  
    <option value="B">B</option>  
    <option value="Γ">Γ</option>  
    <option value="Δ">Δ</option>
```

Πτυχιακή εργασία του φοιτητή Γεωργακόπουλου Βασίλη

```
<option value="E">E</option>
<option value="Z">Z</option>
<option value="H">H</option>
<option value="Θ">Θ</option>
</select>

<div style="display: none;">
<label for="content">Έτος<span class="red">*</span></label>
<input name="year" class="text" style="width:75px"/>
</div>
</div>

<div class="element">
<label for="content">Περιγραφή</label>
<textarea name="description" class="textarea" rows="10"></textarea>
</div>

<div class="entry">
<button type="submit" name="add_lesson"
class="add">Προσθήκη</button>
<button type="reset" class="cancel">Εκκαθάριση</button>
</div>

</form>

<?
}
elseif($_GET['action']=='add_sublesson')
{
?>

<div class="h_title">Δημιουργία Τμήματος</div>
<form onsubmit="return validateDates();" action="#" method="post">
```



```
<?
if(isset($_POST['add_sublesson']))
{
if(strtotime($_POST['date_end'])-strtotime($_POST['date_start'])>0)
{
if($_SESSION['username']=='admin')
    $flag=add_sublesson($_POST,$_POST['teacher']);
else
    $flag=add_sublesson($_POST,$_SESSION['username']);
if($flag==1)
{
?>
<div class="n_ok">
    <p>Το Τμήμα <b><?=get_col_by_id("lesson","title",(int)$_GET['id'])?> -
    <?=$_POST['title'].($_POST['date_start']."/".$_POST['date_end'].")"?
></b> προστέθηκε επιτυχώς.
    Επιστροφή στο μάθημα <a href="lesson.php?action=view&id=<?
=$_GET['id'])?>"><?=get_col_by_id("lesson","title",(int)$_GET['id'])?></a>
    </p>
</div>
?>
}
elseif($flag==2)
{
?>
    <div class="n_error"><p>Υπάρχει ήδη καταχωρημένο τμήμα <?
=$_POST['title'])?> στο μάθημα <?=get_col_by_id("lesson","title",(int)$_GET['id'])?>!
<br>Παρακαλώ εισάγετε ένα διαφορετικό τίτλο τμήματος.</p></div>
```

Πτυχιακή εργασία του φοιτητή Γεωργακόπουλου Βασίλη

```
<?
}
elseif($flag==3)
{
?>
<div class="n_error"><p>Μη έγκυρο username καθηγητή! Παρακαλώ
επιλέξτε καθηγητή.</p></div>
<?
}
else
{
?>
<div class="n_error"><p>Υπήρξε κάποιο σφάλμα κατά την ενημέρωση της
βάσης δεδομένων!</p></div>
<?
}
}
else
{
?>
<div class="n_error"><p>Η ημερομηνία λήξης είναι πριν την ημερομηνία
έναρξης. Παρακαλώ εισάγετε τις σωστές ημερομηνίες.</p></div>
<?
}
}
?>
<div class="element">
```

Πτυχιακή εργασία του φοιτητή Γεωργακόπουλου Βασίλη

```
<label for="name">Μάθημα : <?=get_col_by_id("lesson","title",(int)
$_GET['id'])?></label>

</div>

<div class="element">

    <label for="content">Εξάμηνο : <?=get_col_by_id("lesson","semester",
(int)$_GET['id'])?></label>

</div>

<?
if($_SESSION['username']=='admin')
{
    ?>

    <div class="element">

        <label for="content">Username Καθηγητή<span
class="red">*</span></label>

        <select name="teacher">

            <option value="-1">Επιλογή</option>

            <?

            $teachers = get_col_by_id("lesson","teacher_username",(int)$_GET['id']);
            $teacher_arr = explode(";", $teachers);
            for($i=0;$i<count($teacher_arr);$i++)
            {
                ?>

                <option value="<?=$teacher_arr[$i]?>"><?=$teacher_arr[$i]?></option>

                <?

            }

            ?>

        </select>

    </div>

}

?>
```

Πτυχιακή εργασία του φοιτητή Γεωργακόπουλου Βασίλη

```
</div>
<?
}
?>
<div class="element">
  <label for="content">Τίτλος<span class="red">*</span></label>
  <input name="title" class="text" style="width:75px" />
</div>
<div class="element">
  <label for="content">Από<span class="red">*</span></label>
  <input id="date1" name="date_start" class="text dateonly"
style="width:75px" />
</div>
<div class="element">
  <label for="content">Έως<span class="red">*</span></label>
  <input id="date2" name="date_end" class="text dateonly" style="width:75px" />
</div>
<input type="hidden" name="lesson_id" value="<?=$_GET['id']?>" />
  <div class="entry">
    <button type="submit" name="add_sublesson"
class="add">Προσθήκη</button>
    <button type="reset" class="cancel">Εκκαθάριση</button>
  </div>
</form>
<?
}
?>
```

```
</div>
```

```
</div>
```

```
<div class="clear"></div>
```

```
</div>
```

K-1.5 Κώδικας add_lesson()

```
function add_lesson($post)
```

```
{
```

```
    global $mysql_link;
```

```
    if($post['semester']==-1)
```

```
        return 4;
```

```
    //check if lesson with same id/title exists
```

```
    if(check_ex_lesson(safeaddslashes($post['tei_id']),safeaddslashes($post['name']),0)==1)
```

```
        return 3;
```

```
    //check if valid usernames
```

```
    $teachers = explode(";", $post['teacher_username']);
```

```
    for($i=0;$i<count($teachers);$i++)
```

```
    {
```

```
        if(strcmp(check_user($teachers[$i]),"staff")!=0)
```

```
            return "2".$teachers[$i];
```

```
    }
```

```
    $sql = "INSERT INTO lesson(id,title,description,semester,tei_id,teacher_username)
```

```
        VALUES(".get_new_id("lesson").",
```

```
        "'.safeaddslashes($post['name'])."',
```

```
        "'.safeaddslashes($post['description'])."',
```

```
        "'.safeaddslashes($post['semester'])."',
```

```
        "'.safeaddslashes($post['tei_id'])."',
```

```
"".safeaddslashes($post["teacher_username"]."");  
if(mysqli_query($mysql_link,$sql)==1)  
    return 1;  
else  
    return 0; }
```

K-1.6 Κώδικας verify_user()

```
function verify_user($username,$pass)  
{  
    /**  
     * returns 0 : failed authentication  
     * returns stud/staff successful authentication  
     * returns 2 : cant bind to ldap server  
     * returns 3 : cant connect to ldap server  
     **/  
    $ldap = get_ldap_data('/var/ldap/ldap.txt');  
    $ldapconn = ldap_connect($ldap['host'],$ldap['port']);  
    //set protocol version  
    ldap_set_option($ldapconn, LDAP_OPT_PROTOCOL_VERSION, 3);  
    if($ldapconn)  
    {  
        $ldapbind=ldap_bind($ldapconn, $ldap['bind_rdn'], $ldap['bind_pass']);  
        // verify binding  
        if($ldapbind)  
        {
```

```
$result = ldap_search($ldapconn,"dc=teithe,dc=gr", "(uid=$username)") or die ("Error  
in search query");
```

```
$info = ldap_get_entries($ldapconn, $result); // iterate over array and print data for  
each entry
```

```
if($info['count']>0)
```

```
{
```

```
if(crypt($pass,'$6$TjegyG.E$')== $info[0]["userpassword"][0])
```

```
{
```

```
    $status = get_profile_from_dn($info[0]["dn"]);
```

```
}
```

```
else
```

```
    $status = 0;
```

```
}
```

```
else
```

```
    $status = 0;
```

```
}
```

```
else
```

```
    $status = 2;
```

```
}
```

```
else
```

```
    $status = 3;
```

```
ldap_close($ldapconn);
```

```
return $status;
```

```
}
```

K-1.7 Κώδικας sync_db.php

```
<div id="main">
```

```
    <div class="full_w">
```

```
<div class="h_title">Συγχρονισμός με τη βάση LDAP</div>
```

```
<h2>LDAP</h2>
```

```
<p>Πατήστε τον παρακάτω σύνδεσμο για να σβηστούν οι ανενεργοί λογαριασμοί φοιτητών. (Δηλαδή όσοι δεν υπάρχουν στον LDAP server)</p>
```

```
<p><a href="settings.php?action=sync_db&action2=do">Sync</a></p>
```

```
<div class="entry">
```

```
<div class="sep"></div>
```

```
</div>
```

```
<?
```

```
if(isset($_GET['action2']) && $_GET['action2']=="do")
{
    $sql = "SELECT * FROM student";
    $res_sql = mysqli_query($mysql_link,$sql);
    if(mysqli_num_rows($res_sql)>0)
    {
        $deleted = 0;
        while($student = mysqli_fetch_array($res_sql))
        {
            if(strcmp(check_user($student['username']),"stud")!=0)
            {
                $stud_del[$deleted]['username']=$student['username'];
                $stud_del[$deleted]['firstname']=$student['firstname'];
                $stud_del[$deleted]['lastname']=$student['lastname'];
                if(isset($_GET['action3']) && $_GET['action3']=="yes")
                    delete_user($stud_del[$deleted]['username']);
                $deleted++;
            }
        }
    }
}
```



```
    }
  }
}
}

if(isset($deleted) && $deleted>0)
{
  if(isset($_GET['action3']) && $_GET['action3']=="yes")
  {
    ?>
    <div class="η_warning">
      <p>Διαγράφηκαν επιτυχώς οι <b><?=$deleted?></b> παρακάτω ανενεργοί
      λογαριασμοί φοιτητών από το σύστημα.</p>
    </div>
    <?
  }
  else
  {
    ?>
    <p>
      <span>Οι παρακάτω φοιτητές δεν είναι πλέον καταχωρημένοι στον LDAP
      server.
      Ειστε σίγουροι ότι θέλετε να διαγράψετε τους παρακάτω φοιτητές;</span>
      <span style="color: red;">Οι φοιτητές θα διαγραφούν ολοκληρωτικά από το
      σύστημα!</span>
      <div style="padding-left: 15px;">
        <a href="settings.php?action=sync_db&action2=do&action3=yes"><button>
        Ναι</button></a>

```

Πτυχιακή εργασία του φοιτητή Γεωργακόπουλου Βασίλη

```
<a href="settings.php"><button> Οχι</button></a>
</div>
</p>
<?
}
?>
<table>
    <thead>
        <tr>
            <th scope="col">#</th>
            <th scope="col">Username</th>
            <th scope="col">Όνομα</th>
            <th scope="col">Επώνυμο</th>
        </tr>
    </thead>
    <tbody>
<?
for($i=0;$i<count($stud_del);$i++)
{
    ?>
    <tr>
        <td class="align-center"><?=$i+1?></td>
        <td class="align-center"><?=$stud_del[$i]
['username']?></td>
        <td class="align-center"><?=$stud_del[$i]['firstname']?></td>
        <td class="align-center"><?=$stud_del[$i]['lastname']?></td>
    </tr>
}
```

```
<?
}
}
?>

</tbody>

</table>

</div>

</div>

<div class="clear"></div>

</div>
```

K-1.8 Κώδικας add_students.php

```
<div id="main">

    <div class="full_w">

        <div class="h_title">Εγγραφή φοιτητών</div>

        <h2>Εγγραφή σε τμήμα</h2>

        <p>Εισάγετε τις πληροφορίες για το φοιτητή/καθηγητή που θέλετε να προσθέσετε. Εάν εγγραφεί καθηγητής, θα έχει τη δυνατότητα να επισκεφτεί τη σελίδα του φοιτητή.</p>

        <?

        $lesson_title = get_col_by_id("lesson","title",(int)$_GET['id']);

        if(isset($_GET['action2']))

        {

            if($_GET['action2']=='delete_student')

            {

                delete_student_enrollment($_GET['student_id'],(int)$_GET['sublesson_id']);

            }

        }

        <div class="n_warning"><p>Ο χρήστης διαγράφηκε επιτυχώς απο το τμήμα.</p></div>
```

Πτυχιακή εργασία του φοιτητή Γεωργακόπουλου Βασίλη

```
<?
}
elseif($_GET['action2']=='update')
{
    $sql_update = "UPDATE student SET firstname = '".safeaddslashes($_GET['f'])."',
                lastname = '".safeaddslashes($_GET['l'])."' WHERE username =
                '".safeaddslashes($_GET['studid'])."'";
    $res_sql_update = mysqli_query($mysql_link,$sql_update);
}
}
if(isset($_POST['enroll_student']))
{
    $flag = enroll_student($_POST);
    if($flag==1)
    {
        ?>
        <div class="n_ok">
            <p>Ο χρήστης <b><?=$_POST['username']?></b> προστέθηκε επιτυχώς στο
            τμήμα <?=get_col_by_id("sublesson","title",$_POST['sublesson'])?>.
        </p>
    </div>
    <?
}
elseif($flag==2)
{
    ?>
```

```
<div class="n_warning"><p>Ο χρήστης <b><?=$_POST['username']?></b> είναι  
ήδη εγγεγραμμένος στο τμήμα <?=get_col_by_id("sublesson","title",$_POST['sublesson'])?>.</p></div>
```

```
<?>
```

```
}
```

```
elseif($flag==3)
```

```
{
```

```
?>
```

```
<div class="n_error"><p>Μη έγκυρο username χρήστη: <b><?=$_POST['username']?></b>.</p></div>
```

```
<?>
```

```
}
```

```
elseif($flag==4)
```

```
{
```

```
?>
```

```
<div class="n_error"><p>Ο χρήστης <?=$_POST['username']?> είναι ήδη  
εγγεγραμμένος σε τμήμα του μαθήματος <?=$lesson_title?>.</p></div>
```

```
<?>
```

```
}
```

```
elseif($flag==5)
```

```
{
```

```
?>
```

```
<div class="n_ok">
```

```
<p>Ο χρήστης <b><?=$_POST['username']?></b> προστέθηκε επιτυχώς στο  
τμήμα <?=get_col_by_id("sublesson","title",$_POST['sublesson'])?>.</p>
```

```
</p>
```

```
</div>
```

```
<div class="n_warning"><p>Ο χρήστης <b><?=$_POST['username']?></b> είναι  
ήδη καταχωρημένος ως "<?=get_student_name($_POST['username'])?>".<br />
```

Πτυχιακή εργασία του φοιτητή Γεωργακόπουλου Βασίλη

Θέλετε να γίνει ενημέρωση της βάσης με τα καινούρια στοιχεία "<?=\$_POST['firstname']. " ".\$_POST['lastname']?>";

<a href="lesson.php?action=add_students&id=<?=\$_GET['id']?>&action2=update&studid=<?=\$_POST['username']?>&f=<?=\$_POST['firstname']?>&l=<?=\$_POST['lastname']?>">ναι <a href="lesson.php?action=add_students&id=<?=\$_GET['id']?>">οχι</p>

</div>

<?>

}

elseif(\$flag==6)

{

?>

<div class="η_error"><p>Δεν υπάρχει δυνατότητα καταχώρησης φοιτητή χωρίς τμήμα. Παρακαλώ επιλέξτε ένα τμήμα.</p></div>

<?>

}

else

{

?>

<div class="η_error"><p>Υπήρξε κάποιο σφάλμα κατά την ενημέρωση της βάσης δεδομένων!</p></div>

<?>

}

}

?>

<form action="lesson.php?action=add_students&id=<?=\$_GET['id']?>" method="post">

<div class="element">

<label for="name">Μάθημα : <?=\$lesson_title?></label>

```
</div>
<div class="element">
    <label for="content">Τμήμα</label>
    <?
    if($_SESSION['username']!='admin')
        $sql = "SELECT id,title FROM sublesson WHERE lesson_id = ".(int)
$_GET['id']." AND teacher_username = '".$_SESSION['username']."'";
    else
        $sql = "SELECT id,title FROM sublesson WHERE lesson_id = ".(int)
$_GET['id'];
    $res_sql = mysqli_query($mysql_link,$sql);
    if(mysqli_num_rows($res_sql)>0)
    {
        ?>
        <select name="sublesson">
            <option value="-1" selected>Επιλογή</option>
            <?
            while($sublesson = mysqli_fetch_array($res_sql))
            {
                ?>
                <option value="<?=$sublesson['id']?>"><?
=$sublesson['title']?></option>
            }
        ?>
    </select>
    <?

```

```
?>
</div>
<div class="element">
    <label for="name">Username <span
class="grey">(Φοιτητή/Καθηγητή)</span><span class="red">*</span></label>
    <input style="width:100px" id="username" name="username" class="text"
onfocus="document.getElementById('firstname').value =
";document.getElementById('lastname').value = ";" />
</div>
<div class="element">
    <label for="name">Όνομα</label>
    <input style="width:100px" id="firstname" name="firstname" class="text"
onfocus="autoCompleteName(document.getElementById('username').value)" />
</div>
<div class="element">
    <label for="name">Επώνυμο</label>
    <input style="width:100px" id="lastname" name="lastname" class="text"
onfocus="autoCompleteName(document.getElementById('username').value)" />
</div>
<input type="hidden" name="lesson_id" value="<?=$_GET['id']?>" />
<div class="entry">
    <button type="submit" name="enroll_student"
class="add">Προσθήκη</button>
    <button type="reset" class="cancel">Εκκαθάριση</button>
</div>
</form>
<div class="h_title">Εγγεγραμμένοι φοιτητές</div>
<table>
    <thead>
```



```
<tr>
  <th scope="col">Τμήμα</th>
  <th scope="col">Username</th>
  <th scope="col">Όνοματεπώνυμο</th>
  <th scope="col" style="width: 65px;">Διαγραφή</th>
</tr>
</thead>
<tbody>
  <?
if($_SESSION['username']!= 'admin')
{
  $sql = "SELECT student_sublesson.student_id,sublesson.id AS sublesson_id
FROM student_sublesson
      INNER JOIN sublesson ON sublesson.id = student_sublesson.sublesson_id
      WHERE sublesson.lesson_id = ".(int)$_GET['id']." AND
sublesson.teacher_username = '".$_SESSION['username']."'";
}
else
{
  $sql = "SELECT student_sublesson.student_id,sublesson.id AS sublesson_id
FROM student_sublesson
      INNER JOIN sublesson ON sublesson.id = student_sublesson.sublesson_id
      WHERE sublesson.lesson_id = ".(int)$_GET['id'];
}
$res_sql = mysqli_query($mysql_link,$sql) or die(mysqli_error($mysql_link));
while($student = mysqli_fetch_array($res_sql))
{
```

Πτυχιακή εργασία του φοιτητή Γεωργακόπουλου Βασίλη

```
?>
<tr>
<td><?=get_col_by_id("sublesson","title",$student['sublesson_id'])?></td>
<td><?=$student['student_id']?></td>
<td><?=get_student_name($student['student_id'])?></td>
<td>
<a href="lesson.php?
action=add_students&id=<?=$_GET['id']?>&action2=delete_student&student_id=<?
=$student['student_id']?>&sublesson_id=<?=$student['sublesson_id']?>" class="table-icon
delete" title="Delete" onclick="return confirm('Θέλετε να διαγράψετε το συγκεκριμένο φοιτητή
απο το τμήμα;');"></a>
</td>
</tr>
<?
}
?>
</tbody>
</table>
<?
}
else
echo "Δεν υπάρχουν διαθέσιμα τμήματα στο συγκεκριμένο μάθημα.Παρακαλώ
δημιουργείστε πρώτα ενα τμήμα <a href=\"lesson.php?id=
$_GET['id'].\"&action=add_sublesson\">εδώ</a>.";
?>
</div>
</div>
<div class="clear"></div>
```

</div>

K-1.9 Κώδικας enroll_student()

```
function enroll_student($post)
{
    global $mysql_link;

    $user = check_user($post['username']);

    if($post['sublesson']==-1)
        return 6;

    //check if username is a valid student/prof
    if(strcmp($user,"stud")!=0 && strcmp($user,"staff")!=0)
        return 3;

    //check if user is enrolled already in that sublesson
    $sql_check_ex_enroll = "SELECT * FROM student_sublesson WHERE student_id = '".
    $post['username']."' AND sublesson_id = ".$post['sublesson'];

    $res_sql_check_ex_enroll = mysqli_query($mysql_link,$sql_check_ex_enroll);

    if(mysqli_num_rows($res_sql_check_ex_enroll)>0)
        return 2;

    //check if user is enrolled in another sublesson of this lesson
    $sql_enrolled = "SELECT * FROM student_sublesson INNER JOIN sublesson ON
    student_sublesson.sublesson_id = sublesson.id WHERE student_id = '".
    $post['username']."' AND sublesson.lesson_id = ".(int)$post['lesson_id'];

    $res_sql_enrolled = mysqli_query($mysql_link,$sql_enrolled);

    if(mysqli_num_rows($res_sql_enrolled)>0)
        return 4;

    //if user is a student check if he exists in the database or add him
    if(strcmp($user,"stud")==0)
    {
```

Πτυχιακή εργασία του φοιτητή Γεωργακόπουλου Βασίλη

```
$sql_check_ex = "SELECT * FROM student WHERE username = ".$post['username']."";
$res_sql = mysqli_query($mysql_link,$sql_check_ex);
if(mysqli_num_rows($res_sql)==0)
{
    $sql_add = "INSERT INTO student(username,firstname,lastname) VALUES(" .
$post['username'].",". $post['firstname'].",". $post['lastname'].")";
    $res_sql_add = mysqli_query($mysql_link,$sql_add);
}
else
{
    //check if user is stored with different name
    if(strcmp(get_student_name($post['username']),$post['firstname']." ".$post['lastname'])!
=0)
        $change = 1;
}

}

//student is enrolled in selected sublesson
$sql_enroll = "INSERT INTO student_sublesson(sublesson_id,student_id) VALUES(" .
$post['sublesson'].",". $post['username'].")";
$res_sql_enroll = mysqli_query($mysql_link,$sql_enroll);
if($res_sql_enroll==1)
{
    if(!isset($change))
        return 1;
    elseif($change==1)
```

```
        return 5;
    }
    else
        return 0;
}
```

K-1.10 Κώδικας add_question.php

```
<?
$lesson_id = (int)$_GET['lesson_id'];
?>
<div id="main">
    <div class="full_w">
        <div class="h_title">Προσθήκη Ερώτησης</div>
        <?
        if(isset($_GET['action2']) && $_GET['action2']=='delete_question')
        {
            if(delete_question((int)$_GET['question_id'])==1)
            {
                ?>
                <div class="h_warning"><p>Η ερώτηση διαγράφηκε επιτυχώς απο το
μάθημα.</p></div>
                <meta http-equiv="refresh" content="1; url=quiz.php?
action=add_question&lesson_id=<?=$lesson_id?>">
                <?
            }
        }
        if(isset($_POST['add_question']))
        {
```

```
$flag = add_question($_POST);

if($flag==1)
{
    ?>

    <div class="n_ok">

        <p>Η ερώτηση προστέθηκε επιτυχώς στο μάθημα <?
=get_col_by_id("lesson","title",(int)$_GET['lesson_id'])?>.

        </p>
    </div>

    <?
}

elseif($flag==2)
{
    ?>

    <div class="n_warning"><p>Η ερώτηση υπάρχει ήδη καταχωρημένη στο μάθημα <?
=get_col_by_id("lesson","title",(int)$_GET['lesson_id'])?>.</p></div>

    <?
}

elseif($flag==3)
{
    ?>

    <div class="n_ok">

        <p>Η ερώτηση προστέθηκε επιτυχώς στο μάθημα <?
=get_col_by_id("lesson","title",(int)$_GET['lesson_id'])?>.

        </p>
    </div>

    <div class="n_warning"><p>Υπήρξε πρόβλημα κατά την καταχώρηση του αρχείου
<?=$_FILES['file']['name'])?>.
```

Πτυχιακή εργασία του φοιτητή Γεωργακόπουλου Βασίλη

Παρακαλώ ελέξτε το μέγεθος (<2mb) και τον τύπο (jpg/png/gif).

```
</p></div>
<?
}
elseif($flag==4)
{
?>
<div class="η_error">
<p>Η ερώτηση πρέπει να έχει τουλάχιστο δυο απαντήσεις.
</p>
</div>
<?
}
else
{
?>
<div class="η_error"><p>Υπήρξε κάποιο σφάλμα κατα την ενημέρωση της βάσης
δεδομένων!</p></div>
<?
}
}
?>
<form action="#" method="post" enctype="multipart/form-data">
<div class="element">
<label for="name">Μάθημα : <?=get_col_by_id("lesson","title",
$lesson_id)?></label>
</div>
```

Πτυχιακή εργασία του φοιτητή Γεωργακόπουλου Βασίλη

```
<div class="element">
    <label for="content">Ερώτηση
    <span>(*)</span></label>
    <textarea name="question" class="textarea"
    rows="4"></textarea>
</div>
<div class="element">
    <label for="content">Βαρύτητα <span>(Μονάδες) (*)</span></label>
    <input name="points" value="1" type="text" />
</div>
<div class="element">
    <label for="attach">Φωτογραφία <span>(jpg/png/gif μέχρι 2 mb)</span></label>
    <input type="file" name="file" />
</div>
<div class="element answer" id="an1">
    <label for="content">Απάντηση <span>(*)</span>
    <select name="truth[]">
        <option value="0">Λάθος</option>
        <option value="1">Σωστή</option>
    </select>
</label>
    <textarea name="answer[]" class="textarea"
    rows="3"></textarea>
</div>
<div class="element answer" id="an2">
    <label for="content">Απάντηση <span>(*)</span>
    <select name="truth[]">
```


Πτυχιακή εργασία του φοιτητή Γεωργακόπουλου Βασίλη

```
<option value="0">Λάθος</option>
<option value="1">Σωστή</option>
</select>
</label>
<textarea name="answer[]" class="textarea"
rows="3"></textarea>
</div>
<div class="newAnswers"></div>
<input type="hidden" name="lesson_id" value="<?=$_GET['lesson_id']?>" />
<div class="element">
<button class="add" type="button" id="addAnswer">Προσθήκη
Απάντησης</button>
</div>
<div class="entry">
<button type="submit" name="add_question"
class="add">Προσθήκη</button>
<button type="reset" class="cancel">Εκκαθάριση</button>
</div>
</form>
<div class="h_title">Υπάρχουσες Ερωτήσεις</div>
<table>
<thead>
<tr>
<th scope="col">#</th>
<th scope="col">Ερώτηση</th>
<th scope="col">Μονάδες</th>
<th scope="col">Απαντήσεις</th>

```

```
<th scope="col">Φωτογραφία</th>
<th scope="col" style="width: 65px;">Διαγραφή</th>
</tr>
</thead>
<tbody>
<?
DESC";
$sql = "SELECT * FROM question WHERE lesson_id = ".$lesson_id." ORDER BY id
$res_sql = mysqli_query($mysql_link,$sql);
while($question = mysqli_fetch_array($res_sql))
{
    get_no_of_records_by_id("answer","question_id",$question['id']);
    if($question['image']!="")
        $pic = "Ναι";
    else
        $pic = "Όχι";
    ?>
<tr>
<td><?=$question['id']?></td>
<td><?=utf8_substr(strip_tags($question['description']),0,150)?></td>
<td><?=$question['points']?></td>
<td><?=get_no_of_records_by_id("answer","question_id",$question['id']);?
></td>
<td><?=$pic?></td>
<td>
<a href="quiz.php?
action=add_question&lesson_id=<?=$lesson_id?>&action2=delete_question&question_id=<?>
```

```
= $question['id']?>" class="table-icon delete" title="Delete" onclick="return confirm('Θέλετε να  
διαγράψετε τη συγκεκριμένη ερώτηση;');"></a>
```

```
</td>
```

```
</tr>
```

```
<?>
```

```
}
```

```
?>
```

```
</tbody>
```

```
</table>
```

```
</div>
```

```
</div>
```

```
<div class="clear"></div>
```

```
</div>
```

K-1.11 Κώδικας add_question()

```
function add_question($post)
```

```
{
```

```
    global $mysql_link;
```

```
    if($post['answer'][0]=='' || $post['answer'][1]==')
```

```
        return 4;
```

```
    $question_id = get_new_id("question");
```

```
    $lesson_id = (int)$post['lesson_id'];
```

```
    //check if question exists
```

```
    $sql_check_ex = "SELECT * FROM question WHERE description =  
    '".safeaddslashes($post['question'])."' AND lesson_id = ".$lesson_id;
```

```
    $res_sql_check_ex = mysqli_query($mysql_link,$sql_check_ex);
```

```
    if(mysqli_num_rows($res_sql_check_ex)>0)
```

```
        return 2;
```

Πτυχιακή εργασία του φοιτητή Γεωργακόπουλου Βασίλη

```
//add question

$sql = "INSERT INTO question(id,lesson_id,description,points) VALUES(".$question_id.", ".
$lesson_id.", ".safeaddslashes($post['question'])."',.(int)$post['points'].")";

$res_sql = mysqli_query($mysql_link,$sql);

//adding answers
for($i=0;$i<count($post['answer']);$i++)
{
    $sql_ans = "INSERT INTO answer(id,description,binary_truth,question_id)
        VALUES(".$get_new_id("answer").", ". $post['answer'][$i]."',.(int)$post['truth'][$i]."',.
        $question_id.)";

    $res_sql_ans = mysqli_query($mysql_link,$sql_ans);
}

//image upload
if(isset($_FILES['file']) && $_FILES['file']['error']==0)
{
    $name = $_FILES['file']['name'];
    switch($_FILES['file']['type'])
    {
        case 'image/jpeg': $ext = 'jpg'; break;
        case 'image/gif': $ext = 'gif'; break;
        case 'image/png': $ext = 'png'; break;
        case 'image/tiff': $ext = 'tif'; break;
        default: $ext = ""; break;
    }
    if ($ext)
    {
```

```
$new_filename = rand()." ".str_replace(" ", "_", filename($_FILES["file"]["name"])).".".
$ext;
move_uploaded_file($_FILES["file"]["tmp_name"], "photos/".$new_filename);
$sql_file = "UPDATE question SET image = ".$new_filename." WHERE id = ".
$question_id;
$res_sql_file = mysqli_query($mysql_link, $sql_file);
if($res_sql_file==0)
    $file_error = 1;
}
else
    $file_error = 1;
}
if($res_sql==1 && !isset($file_error))
    return 1;
if($res_sql==1 && isset($file_error))
    return 3;
else
    return 0; }
```

K-1.12 Κώδικας add_quiz()

```
function add_quiz($post,$questions)
{
    global $mysql_link;
    $quiz_id = get_new_id("quiz");
    //check title
    if($post['description']==")
        return 4;
    if($post['sublesson']==-1)
```

```
return 5;

//check if questions are selected
if(count($questions)==0)

return 2;

//check for quiz time
if(!isset($post['length']) || $post['length']=='0' || (int)$post['length']==0)

return 3;

//add quiz

$sql = "INSERT INTO quiz(id,sublesson_id,title,date_starting,date_ending,length)

VALUES(".$quiz_id.",".(int)
$post['sublesson'].",".safeaddslashes($post['description']).",".safeaddslashes($post['date_star
ting']).",".safeaddslashes($post['date_ending']).",".(int)$post['length'].")";

$res_sql = mysqli_query($mysql_link,$sql);

//if quiz added ,add questions to quiz
if($res_sql==1)
{
for($i=0;$i<count($questions);$i++)
{
$sql_q = "INSERT INTO quiz_question(question_id,quiz_id)

VALUES(".$questions[$i].",".$quiz_id.)";

$res_q = mysqli_query($mysql_link,$sql_q);
}
return 1;
}
else
return 0; }
```

K-1.13 Κώδικας add_quiz.php

```
<?
$lesson_id = (int)$_GET['lesson_id'];
?>
<div id="main">
    <div class="full_w">
        <div class="h_title">Προσθήκη Ερωτηματολογίου</div>
        <h2>Ερωτηματολόγιο</h2>
        <p>Προσθήκη ερωτήσεων απο τον παρακάτω πίνακα.</p>
        <?
        if(isset($_GET['action2']) && $_GET['action2']!='delete_question')
        {
            if(delete_question((int)$_GET['question_id'])==1)
            {
                ?>
                <div class="n_warning"><p>Η ερώτηση διαγράφηκε επιτυχώς απο το
μάθημα.</p></div>
                <meta http-equiv="refresh" content="1; url=quiz.php?
action=add_question&lesson_id=<?=$lesson_id?>">
                <?
            }
        }
        if(isset($_POST['add_quiz']))
        {
            if(strtotime($_POST['date_ending'])-strtotime($_POST['date_starting'])>0)
            {
                if(isset($_POST['question_id']))
```

```
$question_array = $_POST['question_id'];  
else  
    $question_array = array();  
$flag = add_quiz($_POST,$question_array);  
if($flag==1)  
{  
    $_SESSION['quiz'] = array();  
    ?>  
    <div class="η_ok">  
        <p>Το ερωτηματολόγιο προστέθηκε επιτυχώς στο τμήμα <?  
=get_col_by_id("sublesson","title",(int)$_POST['sublesson'])?>.</p>  
        </p>  
    </div>  
    <?  
}  
elseif($flag==2)  
{  
    ?>  
    <div class="η_error"><p>Δεν υπάρχει δυνατότητα καταχώρησης  
ερωτηματολογίου χωρίς ερωτήσεις! Παρακαλώ προσθέστε ερωτήσεις.</p></div>  
    <?  
}  
elseif($flag==3)  
{  
    ?>  
    <div class="η_error"><p>Δεν υπάρχει δυνατότητα καταχώρησης  
ερωτηματολογίου χωρίς χρόνο διάρκειας.</p></div>
```


Πτυχιακή εργασία του φοιτητή Γεωργακόπουλου Βασίλη

```
<?
}
elseif($flag==4)
{
?>
<div class="n_error"><p>Δεν υπάρχει δυνατότητα καταχώρησης
ερωτηματολογίου χωρίς τίτλο.</p></div>
<?
}
elseif($flag==5)
{
?>
<div class="n_error"><p>Δεν υπάρχει δυνατότητα καταχώρησης
ερωτηματολογίου χωρίς τμήμα.Παρακαλώ επιλέξτε ένα τμήμα.</p></div>
<?
}
else
{
?>
<div class="n_error"><p>Υπήρξε κάποιο σφάλμα κατα την ενημέρωση της βάσης
δεδομένων!</p></div>
<?
}
}
else
{
?>
```

Πτυχιακή εργασία του φοιτητή Γεωργακόπουλου Βασίλη

```
<div class="η_error"><p>Η ημερομηνία λήξης είναι πριν την ημερομηνία  
έναρξης. Παρακαλώ εισάγετε τις σωστές ημερομηνίες.</p></div>
```

```
<?
}
}
$lesson_title = get_col_by_id("lesson","title",$lesson_id);
?>

<form onsubmit="return validateDates();" id="quizform" action="#" method="post">
  <div class="element">
    <label for="name">Μάθημα : <?=$lesson_title?></label>
  </div>
  <div class="element">
    <?
    if($_SESSION['username']!= 'admin')
      $sql = "SELECT id,title FROM sublesson WHERE lesson_id = ".$lesson_id."
AND teacher_username = '".$_SESSION['username']."'";
    else
      $sql = "SELECT id,title FROM sublesson WHERE lesson_id = ".$lesson_id;
    $res_sql = mysqli_query($mysql_link,$sql);
    if(mysqli_num_rows($res_sql)>0)
    {
      ?>
      <label for="content">Τμήμα <span>(*)</span>
      <select name="sublesson">
        <option value="-1" selected>Επιλογή</option>
      <?

```

```
while($sublesson = mysqli_fetch_array($res_sql))
{
    ?>
    <option value="<?=$sublesson['id']?>"><?=$sublesson['title']?></option>
    <?
}
?>
</select>
</label>
<?
}
    else die("Δεν υπάρχουν διαθέσιμα τμήματα. Παρακαλώ δημιουργήστε πρώτα
ενα τμήμα <a href=\"lesson.php?id=
$_GET['lesson_id'].\"&action=add_sublesson\">εδώ</a>.")
    ?>
        </div>
<div class="element">
    <label for="content">Τίτλος <span>(Περιγραφή) (*)</span></label>
    <input name="description" type="text" />
</div>
<div class="element">
    <label for="content" >Ημερομηνία έναρξης <span> (*)</span><br><span>Για την
τωρινή ημερομηνία/ώρα επιλέξτε now</span></label>
    <input id="date1" name="date_starting" type="text" class="datepicker" />
</div>
<div class="element">
    <label for="content" >Ημερομηνία λήξης <span> (*)</span></label>
```

```
<input id="date2" name="date_ending" type="text" class="datepicker"/>
</div>
<div class="element">
  <label for="content">Διάρκεια <span> (λεπτά) (*)</span></label>
  <input name="length" type="text" />
</div>
<input type="hidden" name="lesson_id" value="<?=$_GET['lesson_id']?>" />
  <div class="entry">
    <button type="submit" name="add_quiz"
class="add">Προσθήκη</button>
    <button type="reset" class="cancel">Εκκαθάριση</button>
  </div>
</form>
<h2>Επιλογή Ερωτήσεων</h2>
<p><ol id="added">
</ol></p>
<div class="element" style="padding-left: 15px;">
  <div style="color: #575757;display: block;font-size: 12px;font-weight: bold;margin: 0
0 5px 0;padding: 0 0 0 3px;" >Επιλογή τυχαίων ερωτήσεων</div><br />
  <span style="color: #b8b8b8;font-size: 11px;font-weight: normal;"> (εισάγετε αριθμό
και πατήστε το κουμπί) (*)</span>
  <input id="randNum" type="text" style="width:22px;" title="εισάγετε αριθμό και
πατήστε το κουμπί" />
  <button style="padding: 12px;" class="add"
onclick="random(document.getElementById('randNum').value);" title="Εισάγετε
αριθμό"></button>
</div>
<br />
```

```
<div class="h_title">Υπάρχουσες Ερωτήσεις</div>
```

```
<table>
```

```
<thead>
```

```
<tr>
```

```
<th scope="col">#</th>
```

```
<th scope="col">Ερώτηση</th>
```

```
<th scope="col">Μονάδες</th>
```

```
<th scope="col">Απαντήσεις</th>
```

```
<th scope="col">Φωτογραφία</th>
```

```
<th scope="col">Μεγένθυση</th>
```

```
<th scope="col" style="width: 65px;">Προσθήκη</th>
```

```
</tr>
```

```
</thead>
```

```
<tbody>
```

```
<?>
```

```
$sql = "SELECT * FROM question WHERE lesson_id = ".$lesson_id." ORDER BY id";
```

```
$res_sql = mysqli_query($mysql_link,$sql);
```

```
while($question = mysqli_fetch_array($res_sql))
```

```
{
```

```
get_no_of_records_by_id("answer","question_id",$question['id']);
```

```
if($question['image']!="")
```

```
    $pic = "Ναι";
```

```
else
```

```
    $pic = "Όχι";
```

```
?>
```

```
<tr>
```

```
<td><?=$question['id']?></td>

<td id="tdQuest<?=$question['id']?>"><?
=utf8_substr(strip_tags($question['description']),0,150)?></td>

<td><?=$question['points']?></td>

<td><?=get_no_of_records_by_id("answer","question_id",$question['id']);?></td>

<td><?=$pic?></td>

<td><a data-fancybox-type="iframe" class="fancyquest"
href="fancy_question.php?id=<?=$question['id']?>"></a></td>

<td>

<input class="random" id="<?=$question['id']?>" onclick="add_question(<?
=$question['id']?>);" type="checkbox" name="add_question[<?=$question['id']?>]" />

</td>

</tr>

<?
}
?>

</tbody>

</table>

</div>
```

```
</div>
```

```
<div class="clear"></div>
```

```
</div>
```

K-1.14 Κώδικας do_quiz.php

```
<?
```

```
if(isset($_GET['quiz_id']))
```

```
{
```

```
    $quiz_id = (int)$_GET['quiz_id'];
```

Πτυχιακή εργασία του φοιτητή Γεωργακόπουλου Βασίλη

```
$quiz_title = get_col_by_id("quiz", "title", $quiz_id);
$quiz_time = get_col_by_id("quiz", "length", $quiz_id);
}
?>
<div id="main">
    <div class="full_w">
        <div class="h_title">Διεξαγωγή Εξέτασης</div>
        <h2><?=$quiz_title?></h2>
        <?
        //check if user has already done quiz
        $sql_check = "SELECT done FROM student_quiz WHERE student_username = '".
$_SESSION['username']."' AND quiz_id = ".$quiz_id;
        $res_sql_check = mysqli_query($mysql_link, $sql_check);
        while($check = mysqli_fetch_array($res_sql_check))
            $done = $check['done'];
        if(isset($done) && $done==1)
        {
            echo "<p>Η εξέταση έχει ολοκληρωθεί.</p>";
        }
        else
        {
            if(!isset($_GET['start']))
            {
                ?>
                <p>
                    <span>Θέλετε να ξεκινήσετε την εξέταση;</span>
                    <span style="color: red;">Θα έχετε διαθέσιμα <?=$quiz_time?> λεπτά!</span>
            }
        }
    }
?>
```

Πτυχιακή εργασία του φοιτητή Γεωργακόπουλου Βασίλη

```
<div style="padding-left: 15px;">
  <a href="student.php?action=do_quiz&quiz_id=<?=$quiz_id?>&start"><button>
Ναι</button></a>

  <a href="student.php"><button> Οχι</button></a>
</div>
</p>
<?
}
else
{
  echo "<p>Επιλέξτε τις σωστές απαντήσεις και πατήστε Υποβολή.</p>";
  //check if user has already started this quiz
  $sql_check = "SELECT * FROM student_quiz WHERE student_username = '".
$_SESSION['username']."' AND quiz_id = ".$quiz_id;
  $res_sql_check = mysqli_query($mysql_link,$sql_check);
  if(mysqli_num_rows($res_sql_check)==0)
  {
    //log time student started the quiz
    $sql_log = "INSERT INTO student_quiz(student_username,quiz_id,date_started)
VALUES ('".$_SESSION['username']."','".$quiz_id."','".$date("Y-m-d H:i:s")."')";
    mysqli_query($mysql_link,$sql_log);
  }
  $sql_time = "SELECT date_started,DATE_ADD(date_started,INTERVAL ".
$quiz_time." MINUTE) AS date_ending FROM student_quiz WHERE student_username = '".
$_SESSION['username']."' AND quiz_id = ".$quiz_id;
  $res_sql_time = mysqli_query($mysql_link,$sql_time);
  while($quiz = mysqli_fetch_array($res_sql_time))
  {
```


Πτυχιακή εργασία του φοιτητή Γεωργακόπουλου Βασίλη

```
$time_started = $quiz['date_started'];
$time_ending = $quiz['date_ending'];
}
//calculating the difference
$difference = strtotime($time_ending) - time();
//calculating hours, minutes and seconds (as floating point values)

if($difference>0)
{
    $hours = $difference / 3600; //one hour has 3600 seconds
    $minutes1 = $hours * 60;
    $seconds = floor(($minutes1 - floor($minutes1)) * 60);
    $minutes = floor($minutes1);
}
if($difference>0 && $minutes>0)
{
    ?>
    <script>
    function submitForm()
    {
        document.getElementById("quizForm").submit();
    }
    var interval;
    var minutes = <?=$minutes?>;
    var seconds = <?=$seconds?>;
    window.onload = function() {
```

```
countdown('countdown');
}

function countdown(element) {
    interval = setInterval(function() {
        var el = document.getElementById(element);
        if(seconds == 0) {
            if(minutes == 0) {
                el.innerHTML = "Ο χρόνος τελείωσε!";
                submitForm();
                clearInterval(interval);
                return;
            } else {
                minutes--;
                seconds = 60;
            }
        }
        if(minutes > 0) {
            var minute_text = minutes + (minutes > 1 ? ' λεπτά' : ' λεπτό');
        } else {
            var minute_text = "";
        }
        var second_text = seconds > 1 ? 'δευτερόλεπτα' : 'δευτερόλεπτο';
        el.innerHTML = minute_text + ' ' + seconds + ' ' + second_text;
        seconds--;
    }, 1000);
}
```

```
}
```

```
</script>
```

```
<div style="font-size: 15px;
```

```
border: 1px solid #ddddd;
```

```
background-color: #f7f7f7;
```

```
min-width: 150px;
```

```
float: right;
```

```
padding: 5px 5px 5px 5px;
```

```
margin-right: 10px;">
```

Απομένουν:

```
<div style="color: red;" id="countdown"></div>
```

```
</div>
```

```
<form id="quizForm" method="POST" action="student.php?
action=quiz_results&quiz_id=<?=$quiz_id?>">
```

```
<p><ol>
```

```
<?
```

```
//get questions from current quiz
```

```
$sql = "SELECT * FROM quiz_question WHERE quiz_id = ".$quiz_id;
```

```
$res_sql = mysqli_query($mysql_link,$sql);
```

```
while($question = mysqli_fetch_array($res_sql))
```

```
{
```

```
//get question details
```

```
$sql_quest = "SELECT * FROM question WHERE id = ".
$question['question_id'];
```

```
$res_sql_quest = mysqli_query($mysql_link,$sql_quest);
```

```
while($quest = mysqli_fetch_array($res_sql_quest))
```

```
{
```

```
?>
</li>
<h3><?=$quest['description']?>
<?
//get question image if available
if($quest['image']!=" && file_exists("photos/".$quest['image']))
{
    ?>
    <a data-fancybox-type="iframe" class="fancyquest"
href="fancy_question.php?id=<?=$quest['id']?>"></a></h3>
    
    <?
}
else
    echo "</h3>";
?>
</li>
<ul style="list-style: none;">
<?
//get answers for current question
$sql_ans = "SELECT * FROM answer WHERE question_id = ".$quest['id'];
$res_sql_ans = mysqli_query($mysql_link,$sql_ans);
while($answer = mysqli_fetch_array($res_sql_ans))
{
    ?>
```

Πτυχιακή εργασία του φοιτητή Γεωργακόπουλου Βασίλη

```
<li style="padding: 3px;"><input type="checkbox" name="<?
=$quest['id']?>[]" value="<?=$answer['id']?>" />&nbsp;<?=$answer['description']?></li>
    <?
  }
?>
</ul>
<?
}
}
?>
</ol></p>
<button type="submit" name="quiz_done" class="add" onclick="return confirm('Αν
υποβληθεί το ερωτηματολόγιο δεν υπάρχει δυνατότητα τροποποίησης');">Υποβολή</button>
</form>
<?
}
else
{
?>
<p style="color: red;">Ο χρόνος για το συγκεκριμένο ερωτηματολόγιο έχει
τελειώσει.</p>
<?
}
}
}
?>
</div>
```

```
</div>
```

```
<div class="clear"></div>
```

```
</div>
```

K-1.15 Κώδικας stud_or_staff.php

```
<?
```

```
session_start();
```

```
if(!isset($_SESSION['stud_or_staff']))
```

```
    header("Location: index.php");
```

```
?>
```

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

```
<html xmlns="http://www.w3.org/1999/xhtml" lang="pl" xml:lang="pl">
```

```
<head>
```

```
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
```

```
<title>SimpleAdmin</title>
```

```
<link rel="stylesheet" type="text/css" href="css/login.css" media="screen" />
```

```
</head>
```

```
<body>
```

```
<div class="wrap">
```

```
    <div id="content">
```

```
        <div id="main">
```

```
            <div class="full_w">
```

```
                <form>
```

```
                    <label><a href="student.php">Είσοδος ως
φοιτητής</a></label>
```

```
                    <label><a href="index.php">Είσοδος ως καθηγητής</a></label>
```

```
                </form>
```

Πτυχιακή εργασία του φοιτητή Γεωργακόπουλου Βασίλη

</div>

<div class="footer">» http://teithess.gr | Tei

LMS</div>

</div>

</div>

</div>

</body>

</html>