

ΠΕΡΙΕΧΟΜΕΝΑ

Εισαγωγή.....σελ.	
1.Συστήματα Εκμάθησης στο διαδίκτυο και το SQLzoo.σελ.	
1.1 E-learning πρακτικές που χρησιμοποιούνται.....σελ.	
1.2 Πλεονεκτήματα-Μειονεκτήματα.....σελ.	
1.3 Δημοφιλείς Πλατφόρμες.....σελ.	
1.4 SQLZoo το έναυσμα για αυτήν την εφαρμογή.....σελ.	
2.Προγράμματα-τεχνολογίες υλοποίησης της εφαρμογής.....σελ.	
2.1 Η πλευρά server.....σελ.	
2.1.1 Η γλώσσα προγραμματισμού.....σελ.	
2.1.1.1 Πληροφορίες.....σελ.	
2.1.1.2 Οδηγίες Εγκατάστασης.....σελ.	
2.1.2 Ο διακομιστής.....σελ.	
2.1.2.1 Πληροφορίες.....σελ.	
2.1.2.2 Οδηγίες Εγκατάστασης.....σελ.	
.2.2 Αλληλεπίδραση με την βάση δεδομένων.....σελ.	
2.2.1 Το σύστημα της Βάσης Δεδομένων.....σελ.	
2.2.2 ODBC σύγκριση με άλλες τεχνικές.....σελ.	
2.3 Η πλευρά client.....σελ	

2.4 Εργαλεία.....	σελ.
3.Απαιτήσεις Χρηστών.....	σελ.
3.1 Παρουσίαση εμπλεκόμενων προσώπων.....	σελ.
3.2 Παρουσίαση απαιτήσεων των ομάδων χρηστών.....	σελ.
4.Σενάρια Χρήσης.....	σελ.
4.1 Περίπτωση Χρήσης 1: Είσοδος φοιτητή.....	σελ.
4.2 Περίπτωση Χρήσης 2: Είσοδος καθηγητή.....	σελ.
5.Προδιαγραφές.....	σελ.
5.1 Το Σχεσιακό Σχήμα της Βάσης Δεδομένων.....	σελ.
5.2 Αρχιτεκτονική-Γενική Μορφή Συστήματος.....	σελ.
6.Λειτουργική Παρουσίαση Συστήματος	σελ.
7.Τεχνική Παρουσίαση Συστήματος	σελ.
7.1 Επισκόπηση.....	σελ.
7.2 Δημιουργία του μηχανισμού Ajax.....	σελ.
7.3 Επέκταση του μηχανισμού Ajax.....	σελ.
7.5 Ο server-side κώδικας.....	σελ.
7.5.1 Οι κλάσεις των ομάδων χρηστών.....	σελ.
7.5.3 Σύνδεση με την Βάση Δεδομένων της Εφαρμογής	σελ.
7.5.4 Απαγόρευση προσπέλασης με χρήση	

μεταβλητών Session.....	σελ.
7.5.5 Διαδικασία Εγγραφής.....	σελ.
7.5.5.1 Ολοκλήρωση Εγγραφής.....	σελ.
7.6 Η μεταφόρτωση ενός αρχείου.....	σελ
8.Βιβλιογραφία.....	σελ.

Εισαγωγή

Το θέμα αυτής της εργασίας είναι η ανάπτυξη μίας ιστοσελίδας και έχει τίτλο : “Διαδικτυακή Διαθεσιμότητα Περιβάλλοντος Εξάσκησης SQL”.

Η εφαρμογή αυτή απευθύνεται σε προπτυχιακούς φοιτητές τμημάτων πληροφορικής και σκοπός της είναι η εκμάθηση της γλώσσας αιτημάτων διαφόρων συστημάτων βάσεων δεδομένων, απαλλάσσοντας τον φοιτητή από τις δυσκολίες της απόκτησης και εγκατάστασης διαφόρων πακέτων λογισμικού.

Η φύση των παραδειγμάτων και ασκήσεων που θα είναι διαθέσιμες μέσω αυτού του εκπαιδευτικού περιβάλλοντος επιλέγεται από τον εκάστοτε υπεύθυνο καθηγητή στον οποίο παρέχεται μία πληθώρα δυνατοτήτων όπως να δημιουργεί συλλογές ασκήσεων οι οποίες ομαδοποιούνται σε ενότητες-εργαστήρια,ευρεση ήδη υπάρχοντων ασκήσεων από μία παλαιότερη συλλογή,παραπομές σε εξωτερικές πηγές , έγγραφα σε ηλεκτρονική μορφή με επεξηγητικό περιεχόμενο για τις ασκήσεις κ.α.

Στις ενότητες που ακολουθούν γίνεται μία προσπάθεια για την λεπτομερή επεξήγηση όλων των πτυχών της συγκεκριμένης εφαρμογής , αφενός για θέματα τεχνολογίας λογισμικού (περιπτώσεις χρήσης,απαιτήσεις χρηστών κλπ) , τεχνικά θέματα (κομμάτια κώδικα που αποτελούν σημεία κλειδιά για την εφαρμογή) αλλά και πληροφορίες για τις απαιτήσεις σχετικά με την εγκατάσταση και τα πακέτα λογισμικού που είναι αναγκαία για την σωστή λειτουργία της εφαρμογής.

SQL-Zoo-Πρακτικές Που χρησιμοποιούνται

1.1 Συστήματα εκμάθησης στο διαδίκτυο

Η τάση που κυριαχεί στο διαδίκτυο αυτήν την στιγμή όσον αφορά τα συστήματα e-learning είναι τα Συστήματα Διαχείρισης Μάθησης (Learning Management Systems). Με αυτό τον όρο αναφερόμαστε σε εφαρμογές με τις οποίες μπορεί κανείς να διαχειριστεί συγκεντρωμένες πληροφορίες ηλεκτρονικής μάθησης στο διαδίκτυο, και ο χρήστης έχει την δυνατότητα διεχείρισης αυτού του περιεχομένου με την βοήθεια νέος πλήθους εργαλείων. Έτσι μπορούμε να ανανεώνουμε μία ιστοσελίδα να περνάμε καινούριο περιεχόμενο, να αλλάζουμε κείμενα φωτογραφίες αρχεία και γενικά οτιδήποτε μπορεί να περιλαμβάνει το επιθυμητό εκπαιδευτικό υλικό. Φυσικά με την εξέλιξη των ταχυτήτων των συνδέσεων στον παγκόσμιο ιστό αλλά και με την αύξηση της υπολογιστικής δυνατότητας των μηχανημάτων που μπορεί να αποκτήσει ο μέσος χρήστης η παραπάνω διαδικασία μπορεί να γίνει εύκολα και γρήγορα. Από τα μεγαλύτερα υπέρ βέβαια των LMS είναι το γεγονός ότι οι δεξιότητες που απαιτούνται από τον χρήστη είναι πολύ περιορισμένες.

Συγκεκριμένα μερικές από τις δυνατότητες περιλαμβάνουν τα εξής:

- Online ανάρτηση υλικού μαθημάτων.
- Αξιολόγηση σπουδαστών-μαθητών με online ερωτήσεις, τεστ κλπ
- Φόρουμ συζητήσεων. Οι συζητήσεις αποτελούν έναν σημαντικό παράγοντα στην εκπαιδευτική διαδικασία, μπορούν να διεξάγονται με την επίβλεψη ενός καθηγητή να ανταλλάσσονται απόψεις-σημειώσεις, και να γίνονται συζητήσεις σε συγκεκριμένα θέματα κατά την δικαρκεια των μαθημάτων.

Τα Συστήματα Διαχείρισης Μάθησης (ΣΔΜ) στηρίζονται σε μια ποικιλία από πλατφόρμες ανάπτυξης, από αρχιτεκτονικές βα-σισμένες σε Java EE μέχρι Microsoft .NET, και συνήθως χρησιμοποιούν στο βάθος μια στιβαρή βάση δεδομένων. Πολύ γρήγορα φάνηκε η ανάγκη ύπαρξης προτύπων για την περιγραφή του μαθησιακού υλικού, ώστε τα συστήματα αυτά να προσφέρουν μεταφερισιμότητα (portability) των μαθησιακών πόρων, διαλειτουργισιμότητα (interoperability) μεταξύ τους και εύκολη αναζήτηση.

Τα δημοφιλέστερα ΣΔΜ που κυκλοφορούν βασίζονται στην αρχιτεκτονική client-server δηλαδή υπάρχει ένας κεντρικός υπολογιστής (server), ο οποίος διαχειρίζεται όλη τη διαδικασία υλοποίησης του μαθήματος. Κάποιες πλατφόρμες υποστηρίζουν τρεις τύπους μαθημάτων, ανάλογα με τη βούληση του διδάσκοντα: τα ανοικτά, τα ανοικτά σε εγγραφή και τα κλειστά.

- Τα ανοικτά μαθήματα είναι τα μαθήματα ελεύθερης πρόσβασης για οποιονδήποτε χρήστη, ακόμα κι αν δεν έχει κάνει εγγραφή.
- Τα ανοικτά σε εγγραφή είναι τα μαθήματα στα οποία μπορεί να έχει πρόσβαση ένας χρήστης μόνο εφόσον έχει κάνει εγγραφή στην πλατφόρμα και διαθέτει κάποιον κωδικό πρόσβασης.
- Κλειστά θεωρούνται τα μαθήματα στα οποία μπορεί να εγγραφεί ένας χρήστης που διαθέτει κωδικό πρόσβασης στην πλατφόρμα, μόνο αν του το επιτρέψει ο υπεύθυνος καθηγητής.

Τώρα όσον αφορά την κατανομή και τον διαχωρισμό των χρηστών η συνήθης πρακτική είναι να οι χρήστες να χωρίζονται στους εκπαιδευόμενους (learners), τους εκπαιδευτές (instructors) και τους διαχειριστές (administrators), η δε προσπέλαση στο σύστημα καθορίζεται από το διακριτό ρόλο που τους έχει ανατεθεί και που καθορίζει πλήρως τα δικαιώματα και τις δυνατότητές τους. Πιο συγκεκριμένα:

- Ο διαχειριστής έχει τη δυνατότητα να επέμβει σε κάθε διαδικασία του συστήματος (διαχειριστική ή και εκπαιδευτική).
- Ο εκπαιδευτής-καθηγητής είναι ο υπεύθυνος του μαθήματος (διαδικασίας και υλικού). Από αυτόν απαιτείται η δημιουργία και ενημέρωση των τάξεων και του υλικού που φιλοξε- νείται στον κεντρικό εξυπηρετητή (server), η προσωπική επικοινωνία με τους εκπαιδευόμενους, η διαχείριση των ομάδων συζήτησης και του πίνακα ανακοινώσεων και η παρακολούθηση της εργασίας και των επιδόσεων των εκπαιδευομένων. Ειδικότερα, ο εκπαιδευτής που αναλαμβάνει την ανάπτυξη και υποστήριξη μαθημάτων μέσα σε ένα ΣΔΜ οφείλει .
 - Να οργανώσει και να «ανεβάσει» το εκπαιδευτικό του υλικό με τέτοιο τρόπο και δομή ώστε να επιτυγχάνονται οι μαθησιακοί στόχοι που έχουν τεθεί.
 - Να εντάξει στο μάθημα υλικό σε διάφορες μορφές ώστε να το καταστήσει ελ-κυστικό και αποτελεσματικό.
 - Να εμπλουτίσει το εκπαιδευτικό υλικό με ασκήσεις, παραδείγματα, μελέτες περίπτωσης κ.λπ., επιδιώκοντας την ενεργοποίηση των εκπαιδευομένων. Να επιμεληθεί τους τρόπους επικοινωνίας με τους εκπαιδευόμενους για την υποστήριξή τους και την αύξηση της αλληλεπίδρασης.
 - Να προσπαθεί να βρει τρόπους κινητοποίησης των εκπαιδευόμενων-φοιτητών και παράλληλα προβολής του διαδικτυακού εκπαιδευτικού περιβάλλοντος της πλατφόρμας.
- Οι εκπαιδευόμενοι είναι οι εγγεγραμμένοι χρήστες, που έχουν τη δυνατότητα παρακολούθησης ενός μαθήματος, συμμετέχοντας στις δραστηριότητες που το αποτελούν. Έχουν πρόσβαση στο εκπαιδευτικό υλικό του μαθήματος, δυνατότητα αυτοαξιολόγησης μέσα από τεστ αυτόματης διόρθωσης και δυνατότητες επικοινωνίας με τον εκπαιδευτή και άλλους εκπαιδευόμενους. Όλοι οι χρήστες χρειάζεται να έχουν ένα

πρόγραμμα-πελάτη (client), ώστε να αλληλεπιδρούν με το υλικό και τις υπηρεσίες που φιλοξενούνται στον εξυπηρετητή. Συνήθως αυτό είναι ένας κοινός browser και σπανιότερα κάποιο ειδικό πρόγραμμα. Κάποιες φορές απαιτείται η χρήση συμπληρωματικών προγραμμάτων, π.χ. για ζωντανή τηλεδιάσκεψη.

Στην εξέλιξη των ΣΔΜ φαίνεται ότι οι λειτουργίες που διατίθενται δείχνουν να συγκλίνουν παρουσιάζοντας κοινά χαρακτηριστικά, όπως ανοικτή αρχιτεκτονική, συμβατότητα με όλους τους φυλλομετρητές, υποστήριξη HTML, μαθησιακά και διαχειριστικά εργαλεία για τον εκπαιδευτή και τον εκπαιδευόμενο, προσφέροντας ενιαίο λογισμικό–περιβάλλον για όλους (κατασκευαστές, εκπαιδευτικούς, μαθητές) και υποστηρίζοντας δραστηριότητες με σκοπό την ευελιξία τόπου και χρόνου κ.ά.

Αναφέρονται κάποια από τα κοινότερα γνωρίσματα

- Αναγνώριση χρήστη με μοναδικούς κωδικούς για λόγους ασφαλείας και οργάνωσης.
- Οργανόγραμμα μαθήματος, που περιλαμβάνει σημαντικά σημεία της διαδικασίας του μαθήματος.
- Ημερολόγιο για τις σημαντικές ημερομηνίες του προγράμματος.
- Λίστα τάξης και προσωπικές σελίδες συμμετεχόντων.
- Ηλεκτρονικό ταχυδρομείο.
- Πίνακες Ανακοινώσεων.
- Εργαλεία για σύγχρονη επικοινωνία (chat, teleconference, whiteboards)
- Μεταδεδομένα, για λόγους κατηγοριοποίησης και ταξινόμησης, π.χ. λέξεις κλειδιά.
- Εργασίες, που προετοιμάζονται από τον διδάσκοντα με δυνατότητα συλλογής των αποτελεσμάτων.

- Αξιολόγηση – Βαθμολόγιο
- Περιοχή αποθήκευσης αρχείων
- Εργαλεία αναζήτησης
- Υπηρεσία σελιδοδείκτη
- Μοντέλο πλοήγησης στο περιβάλλον εργασίας
- Καταγραφή στατιστικών στοιχείων
- Γλωσσάρι
- Βοήθεια

Αυτή την στιγμή στην κυκλοφορία υπάρχουν ΣΔΜ ανοιχτού κώδικα , εμπορικά, άλλα που κατασκευάζονται κατά παραγγελία.εν πάσει περιπτώσει η πληθώρα αυτή παρέχει μεγάλη δυνατότητα επιλογής, ανάλογα με τις ανάγκες αλλά και παράγοντες όπως το κόστος λειτουργίας και συντήρησης της υπηρεσίας ή των πόρων που αυτή απαιτεί για την ικανοποιητική και αποδοτική λειτουργία της. Για τον χώρο των εκπαιδευτικών ιδρυμάτων που το κόστος αποτελεί έναν σημαντικό παράγοντα εφόσον πέρα από την αρχική δαπάνη πρέπει να υπολογιστούν και τα κόστη για τις συντηρήσεις αλλά και πιθανόν κόστη για τις ανανεώσεις της άδειας χρήσης. Για αυτό το λόγο είναι συνηθισμένο η πλειοψηφία των φορέων που έχουν την δυνατότητα (άτομα με το απαραίτητο γνωστικό επίπεδο στον χώρο της πληροφορικής ,υποδομές) είτε να αποκτούν ΣΔΜ ανοιχτού κώδικα είτε να αναπτύσσουν τα δικά τους και παράλληλα να αναλαμβάνουν οι ίδιοι και την συντήρηση.

Γενικά η παραπάνω προσέγγιση έχει πολλά σημαντικά πλεονεκτήματα:

- Το ανοικτό λογισμικό εξελίσσεται πιο γρήγορα και ισορροπημένα, διότι πολλά άτομα και ομάδες δουλεύουν παράλληλα,

επιτυγχάνοντας ταχύτερη πρόοδο από ό,τι μπορεί να καταφέρει μια ομάδα μεμονωμένα.

- Οι ανάγκες των χρηστών καλύπτονται γρήγορα, καθώς το μοντέλο ανάπτυξης ανοικτού λογισμικού επιτρέπει την άμεση ενσωμάτωση της συλλογικής τεχνογνωσίας και συνεισφοράς.
- Νέες εκδόσεις του λογισμικού διατίθενται στους χρήστες του σε τακτά χρονικά διαστήματα.
- Η ομάδα ανάπτυξης του λογισμικού αποτελείται συνήθως από πολλούς γεωγραφικά διασκορπισμένους «εθελοντές» με εξειδίκευση σε διαφορετικούς τομείς και με διαφορετικές εμπειρίες.
- Δεν τίθενται σοβαρά θέματα ασφαλείας, επειδή ο κώδικας είναι ελεύθερα διαθέσιμος.

1.2 Πλεονεκτήματα-Μειονεκτήματα

Ως πλεονεκτήματα των Συστημάτων Διαχείρισης Μάθησης αναφέρονται πιο συγκεκριμένα τα παρακάτω:

- Ελαστικότητα, προσβασιμότητα, εξυπηρέτηση. Οι χρήστες μπορούν να μάθουν μέσα από ένα εκπαιδευτικό πρόγραμμα και επίσης να προσπελάσουν αυτή την εκπαίδευση ότι ώρα θέλουν και για όση ώρα θέλουν.
- Ανεξαρτησία πλατφόρμας (cross platform). Η διαδικασία μπορεί να πραγματοποιηθεί web browsing λογισμικό σε οποιαδήποτε πλατφόρμα: Windows, Mac, Unix κλπ. Μπορούμε να παραδώσουμε το εκπαιδευτικό πρόγραμμα σε οποιαδήποτε μηχανή μέσω του Internet χωρίς να χρειάζεται να προγραμματίζουμε πολλές φορές για όλες τις πλατφόρμες.
- Τα λογισμικά των browser και οι συνδέσεις Internet είναι ευρέως διαθέσιμες. Οι περισσότεροι χρήστες υπολογιστών έχουν τη δυνατότητα να χρησιμοποιούν browser όπως τον Internet explorer και να συνδέονται στο διαδίκτυο μέσω διαφόρων εταιριών.
- Φθηνή διανομή παγκοσμίως. Δεν χρειάζονται ξεχωριστοί μηχανισμοί διανομής. Η ηλεκτρονική μάθηση είναι εφικτή σε κάθε υπολογιστή οπουδήποτε στον κόσμο διατηρώντας τα κόστη διανομής χαμηλά.
- Ευκολία αναβάθμισης. Εάν χρειαστεί να γίνουν αλλαγές στο πρόγραμμα ύστερα από τη γνήσια υλοποίησή του, μπορούν να γίνουν στον server όπου υπάρχει αποθηκευμένο το πρόγραμμα και έτσι οποιοσδήποτε στον κόσμο μπορεί αμέσως να πάρει την αναβάθμιση.
- Κόστη ταξιδιών και εξοικονόμηση χρόνου. Δεν υπάρχουν κόστη ταξιδιών για τη μεταφορά απομακρυσμένων υπαλλήλων στα κεντρικά γραφεία των επιχειρήσεων γιατί το διαδίκτυο είναι εφικτό από τον Η/Υ οπουδήποτε. Επιπλέον ο πραγματικός

χρόνος που απαιτείται για την εκπαίδευση μέσω υπολογιστή είναι το 50% του χρόνου των μαθημάτων με καθηγητή και έτσι μειώνονται τα κόστη.

Στα μειονεκτήματα συγκαταλέγονται τα εξής:

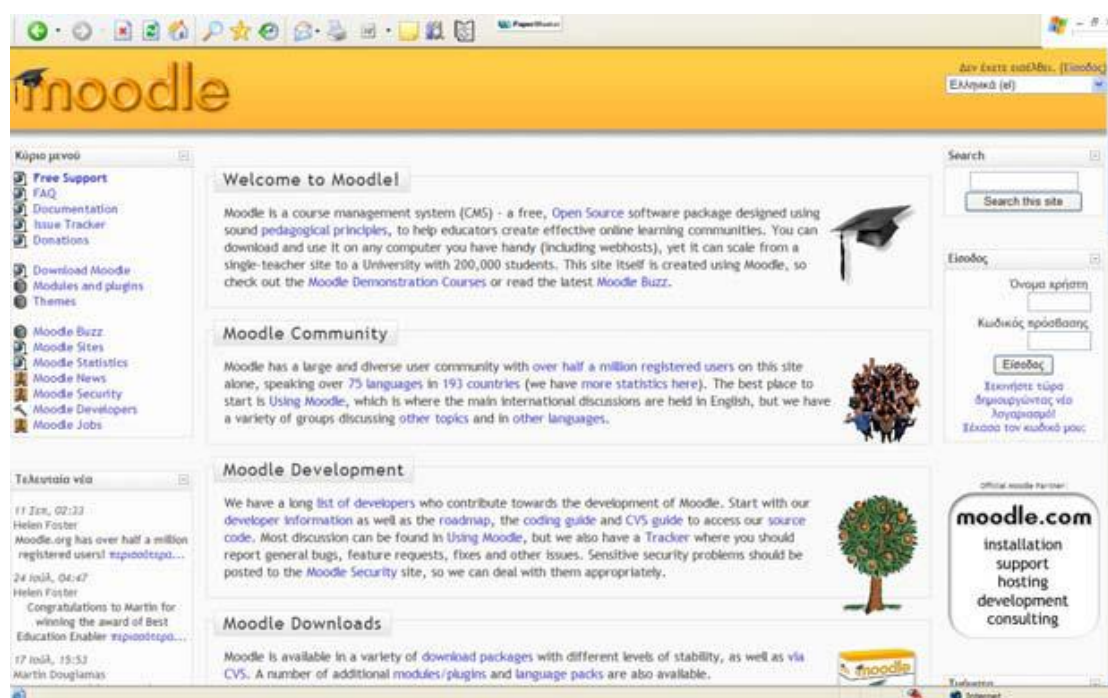
- Περιορισμοί στο εύρος ζώνης. Ο περιορισμός στο εύρος ζώνης σημαίνει χαμηλότερη απόδοση ήχου, βίντεο και ζωντανών γραφικών, προκαλώντας μεγάλες αναμονές στο downloading που μπορούν να επηρεάσουν την ευκολία της διαδικασίας εκμάθησης. Το πρόβλημα είναι μεγαλύτερο στο παγκόσμιο δίκτυο, το Internet όπου πολύ περισσότερα 'κυκλοφοριακά' προβλήματα προκύπτουν και μικρότερο στα τοπικά intranets των εταιριών όπου έχουν συνήθως μεγαλύτερο bandwidth. Οι τεχνολογίες του μέλλοντος θα βοηθήσουν στην επίλυση του προβλήματος αυτού. Βέβαια αυτό το φαινόμενο σήμερα είναι πολύ σπάνιο.
- Μπορούν οι υπολογιστές να αντικαταστήσουν την ανθρώπινη επαφή; Υπάρχει ένα γενικό θέμα, ότι όσο προχωράμε προς τη μεγαλύτερη χρήση των υπολογιστών, ένα τερματικό θα αντικαθιστά ένα φιλικό πρόσωπο.
- Χρειάζεται περισσότερο χρόνο και κόστος για να αναπτυχθεί από το αναμενόμενο. Η εκμάθηση και υλοποίηση της νέας τεχνολογίας απαιτεί περισσότερες πηγές από τις αναμενόμενες. Μπορούμε να το κάνουμε πιο εύκολα ξεκινώντας από ένα απλό πρόγραμμα και χτίζοντας νέες δυνατότητες στη συνέχεια. Μην ξεχνάμε επίσης ότι το μεγαλύτερο μέρος του κόστους που σχετίζεται με τα ΣΔΜ είναι αυτό της αρχής τους. Τα προγράμματα στη συνέχεια μπορούν να παραδοθούν και να επαναχρησιμοποιηθούν με χαμηλότερα κόστη με τις παραδοσιακές μεθόδους.
- Δεν παραδίδονται όλα τα μαθήματα ικανοποιητικά από τους υπολογιστές. Μερικά εκπαιδευτικά θέματα δεν παραδίδονται με τον καλύτερο τρόπο από τους υπολογιστές και απαιτούν ανθρώπινη παρέμβαση. Ομαδικές δραστηριότητες και συναισθηματικά θέματα έρχονται στην επιφάνεια. Τα ΣΔΜ και οι υπόλοιπες εκπαιδευτικές

τεχνολογίες είναι κυρίως για να βοηθούν τη διαδικασία της εκμάθησης και δεν αντικαθιστούν μεθόδους που ήδη λειτουργούν καλά.

1.3 Δημοφιλείς πλατφόρμες

Σε αυτό το σημείο θα γίνει μία αναδρομή στις δημοφιλέστερες εφαρμογές Συστημάτων διαχείρισης Μάθησης καθώς και η παράθεση των πιο αξιόλογων γνωρισμάτων τους που χρησιμοποιούνται στις μέρες μας και ανακαλύφθηκαν μετά από αναζήτηση στο διαδίκτυο.

Moodle: Αναπτύχθηκε το 1999 από τον Αυστραλό Martin Dougiamas με τη γλώσσα προγραμματισμού PHP και χρησιμοποιεί ως σύστημα διαχείρισης βάσεων δεδομένων τη MySQL . Παγκοσμίως χρησιμοποιείται μεταφρασμένο σε 78 γλώσσες από εγγεγραμμένους χρήστες που ξεπερνούν το μισό εκατομμύριο σε 210 χώρες. Στην Ελλάδα έχει εγκατασταθεί σε περισσότερους από 40 φορείς εκπαίδευσης και οργανισμούς, με περισσότερους από 160 ιστοχώρους



Πλατφόρμα Moodle

Στις δυνατότητες του συγκεκριμένου ΣΔΜ συγκαταλέγονται τα εξής:

- Εγγραφή χρηστών με δημιουργία δικού τους λογαριασμού και επιβεβαίωση των διευθύνσεων email.
- Έλεγχος του διδάσκοντα στο περιεχόμενο του μαθήματος και ποικιλία στις δραστηριότητες.
- Συγκεντρωτική παρουσίαση βαθμολογίας.
- Πολύγλωσση υποστήριξη κειμένων.
- Έλεγχος των εργασιών από τον καθηγητή.

Blackboard: Πιθανά το πιο διαδεδομένο και το πιο πετυχημένο εμπορικά ΣΔΜ. Η εταιρεία που το αναπτύσσει ιδρύθηκε το 1997 και απασχολεί περισσότερους από χίλιους εργαζόμενους, υποστηρίζοντας με το λογισμικό της σε περισσότερα από 5.000 εκπαιδευτικά ιδρύματα και οργανισμούς. Στα προϊόντα της περιλαμβάνονται το Blackboard Academic Suite release 8, το Blackboard Academic Suite for K-12 re-release 8, το Blackboard Learn release 9.0, το Blackboard Connect και από την πρώην WebCT Inc, την οποία εξαγόρασε το 2006, τα CMS Blackboard Learning System Vista και Campus Edition .Αποτελεί μια ευέλικτη πλατφόρμα την οποία οι εκπαιδευτές μπορούν να προσαρμόσουν ανάλογα με τη θεωρία μάθησης ή το μοντέλο διδασκαλίας που θέλουν να χρησιμοποιήσουν, ενώ παρέχει δυνατότητα επέκτασης ανάλογα με τις ανάγκες του ιδρύματος/οργανισμού που προσφέρει τα εκπαιδευτικά προγράμματα, πολυγλωσσική υποστήριξη και αρχιτεκτονική που επιτρέπει την ενσωμάτωση άλλων εφαρμογών

Blackboard Learning System ML™

Καλώς Ήλθατε!

Καλώς Ήλθατε στην πολυγλωσσική έκδοση Blackboard Learning System ML™ – του πρωτοπόρου συστήματος διαχείρισης μαθημάτων της εταιρίας Blackboard Inc. Το Blackboard Learning System ML™ παραδίδει ένα σύστημα διαχείρισης μαθημάτων, προσαρμοσμένες Πύλες για Ιδρύματα, δικτυακές κοινότητες και μια προηγμένη αρχιτεκτονική που επιτρέπει δικτυακά βασισμένη ολοκλήρωση (Web-based integration) με άλλα συστήματα διαχείρισης.

Blackboard LMS

Πιό συγκεκριμένα με την χρήση του blackboard ο χρήστης έχει την δυνατότητα:

- Να δημιουργεί ή να εισάγει διδακτικό υλικό κατά μονάδες ή πακέτα, όπως πληροφορίες, ανακοινώσεις, σημειώσεις, διαλέξεις, ασκήσεις, βιβλιογραφία κ.ά., ώστε να βρίσκεται στη διάθεση των εκπαιδευομένων σε 24ωρη βάση ή σε καθορισμένο χρόνο.
- Να επικοινωνεί ηλεκτρονικά με τους εκπαιδευόμενους, να δέχεται ηλεκτρονικά τις εργασίες τους, να διοργανώνει συζητήσεις πάνω σε συγκεκριμένα θέματα μέσω discussion board, chat κ.ά.,

- Να δημιουργεί και να προσφέρει online τεστ στους εκπαιδευόμενους, με ερωτήσεις από τη διαθέσιμη «τράπεζα εξετάσεων» ή καινούριες, που να έχουν κάθε δυνατή μορφή, ενώ οι βαθμοί των εκπαιδευομένων καταχωρούνται αναλυτικά κατά δραστηριότητα και κατά μάθημα στο «Βαθμολόγιο» (Gradebook), με δυνατότητα απεικόνισης της συνολικής πορείας προόδου,
- Να εξασφαλίζει την πρόσβαση στο υλικό του μαθήματος μόνο σε συγκεκριμένους χρήστες.

JoomlaLMS: Πρόκειται για την διαμόρφωση του γνωστού και πολύ δημοφιλούς συστήματος διαχείρισης περιεχομένου (CMS) και η πρώτη του εμφάνιση έγινε το 2007 και απαιτεί και μία εγκατάσταση του παραπάνω CMS προκειμένου να λειτουργήσει. Τα συστατικά της Professional έκδοσης είναι ανοιχτού κώδικα και έτσι δίνεται η δυνατότητα τροποποίησης του πηγαίου κώδικα σε αυτόν που θα το αποκτήσει.

The screenshot displays the Joomla! LMS interface in a Mozilla Firefox browser window. The page title is "eLearning solutions | Joomla! learning management system (LMS) - Mozilla Firefox". The URL is "http://www.joomla-lms.com/lms-elearning-solutions/joomla-lms-demonstration-course.html".

The interface includes a navigation menu with options like News, Features, Demo, Compare LMS, Try&Buy, Tutorials, FAQ&Help, Forum, and Resellers. A search bar is located in the top right corner.

The main content area is titled "Joomla!MS 1.0.6 - Test eLearning solutions - Joomla!MS Demo | recommended course". It features a "Course details" section with a "Describe yourself homework" link, an "Announcements" section with a "Course announcement for students" link, and a "DropBox" section with a "Barber's Chair.jpg" link.

A welcome message states: "We are glad to welcome you at our first Joomla!MS Demonstration Course. Our aim is to show you all the main functions of Joomla!MS. During our lessons you will learn how to use Joomla!MS facilities and how to apply them to your course creation using a great number of tools and methods including such Rapid Authoring tools as WordForce, QuizForce and PowerPointForce. At the end of the course you will have a good command of Joomla!MS toolkit. Our course will show you how to create courses by means not only SCORM production but by means of native Documents, Quizzes and Learning Paths. It will be possible for you to enrich your courses with such teaching elements as Word Documents, Flash Quizzes, PowerPoint Presentations."

A "Don't delay!" section encourages users to "Study our Learning Path to get to know Joomla!MS features." Below this is a table of course elements:

#	Element name	Controls	Details	Description
1	Joomla!MS Basic Course Sections	▼	✓	-
2	Collaboration & File Exchange Tools	▲	✓	-
3	Tracking Students' Results	▲	✓	-

The interface also includes a "login" section with a "Hi User" message and a "Log out" button. The footer contains the text "LMS e-learning software by Joomla!MS Copyright © 2006 - 2009".

Στα χαρακτηριστικά του συγκαταλέγονται τα εξής:

- Ευκολία στην εγκατάσταση
- Επεξεργασία του περιεχομένου με απλή γνώση word.
- Συλλογές πολυμέσικών αρχείων.
- Δυνατότητα ηλεκτρονικού εμπορίου,
- Δυνατότητα για δημιουργία blogs, newsletters, email.
- Συλλογή δεδομένων και δημιουργία αναφορών.
- Χώροι για φόρουμ συζητήσεις.
- Δημιουργία καταλόγων περιεχομένου.

1.4 SQLZoo το έναυσμα για αυτή την εφαρμογή

Όπως έχει ήδη αναφερθεί ο λόγος ανάπτυξης αυτού του συστήματος “Διαδικτυακή διαθεσιμότητα Περιβάλλοντος Εξάσκησης SQL” η διευκόλυνσης που θα παρέχει η ύπαρξη ενός ιστοτόπου ώστε η εξάσκηση φοιτητών τμημάτων πληροφορικής στην γλώσσα αιτημάτων προς ένα σύστημα διαχείρισης βάσεων δεδομένων να γίνεται με ταχύτητα και ευκολία (φυσικά και για την πλευρά των καθηγητών) αλλά πηγή έμπνευσης αποτελεί ο ιστοτόπος <http://sqlzoo.net/> .

Πρόκειται για ένα εγχείρημα του Andrew Cumming καθηγητή στο πανεπιστήμιο του Εδιμβούργου. Το συγκεκριμένο site προσφέρει μία συλλογή απλών παραδειγμάτων που συγκροτούν τις βασικές γνώσεις πάνω στην SQL, σε διάφορα DBMS. Αυτή την στιγμή υποστηρίζονται τα εξής:

- MySQL 5
- Ingres
- PostgreSQL
- Microsoft SQL Server
- IBM DB2
- Oracle

Βέβαια ανα εξαιρέσουμε το κομμάτι της εκτέλεσης κάποιων sql αιτημάτων και την απόδοση της απόκρισης από την πλευρά του server (την αντίστοιχη βάση δεδομένων ανάλογα με την περίπτωση) το υπόλοιπο περιεχόμενο είναι στατικό. Δεν υπάρχει δυνατότητα ή τουλάχιστον δεν είναι εύκολο να γίνει ανανέωση των ασκήσεων που είναι διαθέσιμες και επίσης δεν είναι δυνατόν να εμπλουτίσουμε το ήδη υπάρχον υλικό με παραπάνω. Παράλληλα η πρόσβαση στις βάσεις δεδομένων δεν απαιτεί πιστοποιήσεις από την μεριά του χρήστη κάτι το οποίο στην περίπτωση μας δεν είναι επιθυμητό.

Παρ όλα αυτά η διαδικασία εκτέλεσης των ερωτημάτων και η οπτική μορφή των ασκήσεων της πτυχιακής αυτής εργασίας εμπνεύστηκε σε μεγάλο βαθμό απο το SQLZoo.

Quick Ref.

SELECT

Functions

SELECT WHERE

SELECT GROUP BY

SELECT JOIN

SELECT SELECT

INSERT VALUES

INSERT SELECT

UPDATE

DELETE

CREATE TABLE

CREATE VIEW

CREATE INDEX

DROP

ALTER table

nobel Nobel Laureates

We continue practicing simple SQL queries on a single table.

This tutorial is concerned with a table of Nobel prize winners:

nobel(yr, subject, winner)

Exercises

Using the **SELECT** statement.

1a. Change the query shown so that it displays Nobel prizes for 1950.

```
SELECT yr, subject, winner
FROM nobel
WHERE yr = 1960
```

nobel		
yr	subject	winner
1960	Chemistry	Willard F. Libby
1960	Literature	Saint-John Perse
1960	Medicine	Sir Frank Macfarlane Burnet
1960	Medicine	Peter Medawar

Wrong Answer. The correct answer has 8 row(s).

yr	subject	winner
1960	Chemistry	Willard F. Libby
1960	Literature	Saint-John Perse
1960	Medicine	Sir Frank Macfarlane Burnet
1960	Medicine	Peter Medawar
1960	Peace	Albert Lutuli
1960	Physics	Donald A. Glaser

Εκτέλεση Ερωτήματος στο SQLZoo

Κεφάλαιο 2°

Εργαλεία-Τεχνολογίες που χρησιμοποιήθηκαν

Σε αυτό το κεφάλαιο θα παρουσιάσουν οι τεχνολογίες και τα πακέτα λογισμικού που χρησιμοποιήθηκαν για την ανάπτυξη αυτής της εφαρμογής. Επίσης θα κάνουμε και μία ανασκόπηση στα εργαλεία που βοήθησαν στην συγγραφή του κώδικα (editors, περιβάλλοντα ανάπτυξης ,εργαλεία αποσφαλμάτωσης κλπ). Θα δωθί βάση στις διαδικασίες απόκτησης και εγκατάστασης τους. Η προσέγγιση η οποία έχει αυτή η παρουσίαση είναι ο διαχωρισμός αυτών των τεχνολογιών ανάλογα με τον σκοπό που εξυπηρετούν, δηλαδή αρχικά θα ξεκινήσουμε με όσες τεχνολογίες λειτουργούν στην πλευρά του διακομιστή, θα συνεχίσουμε με τις γλώσσες προγραμματισμού και τα πακέτα τα οποία χρησιμοποιούνται στην πλευρά του χρήστη και τέλος θα γίνει η παρουσίαση των εργαλείων. Επίσης θα δωθούν και οι πληροφορίες εγκατάστασης όπου κρίνεται απαραίτητο.

2.1 Η πλευρά server

2.1.1 Η γλώσσα προγραμματισμού

Όσον αφορά την πλευρά του διακομιστή της εφαρμογής χρησιμοποιήθηκε η γλώσσα προγραμματισμού PHP. Πρόκειται για ένα πακέτο ανοιχτού λογισμικού και χρησιμοποιείται όπως είπαμε και παραπάνω από την πλευρά του server για την δημιουργία δυναμικού περιεχομένου στις ιστοσελίδες και υποστηρίζονται όλες οι πλατφόρμες λειτουργικών συστημάτων (Linux, Windows, Macintosh κλπ), αρκεί να υπάρχει εγκατεστημένος και ο κατάλληλος διακομιστής που στην συγκεκριμένη περίπτωση είναι εξ' ορισμού ο Apache που θα συζητηθεί παρακάτω.

Η ιστορία αυτής της γλώσσας προγραμματισμού ξεκινάει το 1995 από τον τότε φοιτητή Rasmus Lerdorf και οι μετέπειτα εκδόσεις είναι βασισμένες στην

γλώσσα C. Στις μέρες μας χρησιμοποιείται για να καλύψει την πλευρά του διακομιστή εκατομμυρίων site και η τελευταία έκδοση που βρίσκεται σε κυκλοφορία είναι η 6.

Ο μηχανισμός έχει ως εξής: αφού εκτελεστεί κάποιο αρχείο PHP στην πλευρά του server παράγεται HTML κώδικας ο οποίος στέλνεται τελικά στον browser του χρήστη για απεικόνιση.

Σχετικά τώρα με την απόκτηση και εγκατάσταση της PHP θα πρέπει να ακολουθήσουμε τα εξής βήματα:

1) Αρχικά θα πρέπει να επισκεφτούμε τον σχετικό δικτυακό τόπο <http://php.net/downloads.php> ώστε να προμηθευτούμε την τελευταία έκδοση συγκεκριμένα θα επιλέξουμε ένα από τα πακέτα 5.*.*.zip.

2) Όσον αφορά την εγκατάσταση σε πλατφόρμα Windows τα πράγματα είναι πολύ εύκολα τό μόνο που έχουμε να κάνουμε είναι να αποσυμπιέσουμε το αρχείο που αποκτήσαμε σε έναν από τους αρχικούς καταλόγους του σκληρού μας δίσκου με όνομα php.

3) Τώρα μένει να εντοπιστούν τα αρχεία php.ini-dist και php5ts.dll όπου το πρώτο θα πρέπει να μετονομαστεί σε php.ini

Και να μεταφερθεί στον κατάλογο C:\WINDOWS \ και το δεύτερο θα πρέπει να μεταφερθεί στον κατάλογο που βρίσκονται τα αρχεία dll του λειτουργικού μας συστήματος. Σε αυτό το σημείο θα πρέπει να τονίσουμε ότι θα χρειαστεί να εγκατασταθούν και οι σχετικές επεκτάσεις για σύνδεση με βάση δεδομένων μέσω odbc αλλά συνήθως αυτό γίνεται εξ ορισμού.

2.1.2 Ο διακομιστής

Για αυτόν τον ρόλο ήταν μονόδρομος η επιλογή του Apache Web Server. Όπως και το παραπάνω πακέτο και αυτο πρόκειται για ανοιχτό λογισμικό και είναι από τα δημοφιλέστερα σε αυτόν τον τομέα ,μεγάλη βαρύτητα σε αυτό έχει το γεγονός ότι είναι συμβατός με όλα τα υπάρχοντα λειτουργικά συστήματα.

Συντηρείται τώρα από μια κοινότητα ανοικτού κώδικα με επιτήρηση από το Ίδρυμα Λογισμικού Apache (Apache Software Foundation).

Η πρώτη του έκδοση, γνωστή ως NCSA HTTPd, δημιουργήθηκε από τον Robert McCool και κυκλοφόρησε το 1993. Θεωρείται ότι έπαιξε σημαντικό ρόλο στην αρχική επέκταση του παγκόσμιου ιστού. Ήταν η πρώτη βιώσιμη εναλλακτική επιλογή που παρουσιάστηκε απέναντι στον εξυπηρετητή http της εταιρείας Netscape και από τότε έχει εξελιχθεί στο σημείο να ανταγωνίζεται άλλους εξυπηρετητές βασισμένους στο Unix σε λειτουργικότητα και απόδοση. Σχετικά με την εγκατάσταση και την χρήση του τα πράγματα είναι λίγο περίπλοκα αλλά σε καμία περίπτωση δύσκολα εφόσον ο χρήστης αποκτήσει μία σχετική εξοικείωση.

Τα βήματα είναι τα εξής:

1) Για την εγκατάσταση σε περιβάλλον Windows τα πράγματα είναι αρκετά εύκολα λόγω της ύπαρξης του οδηγού εγκατάστασης.Αφού λοιπόν επισκευτούμε τον δικτυακό τόπο:

<http://www.apache.org/>

Επιλέγουμε να κατεβάσουμε την Win32 έκδοση από τον αντίστοιχο σύνδεσμο .

2) Ξεκινώντας λοιπόν τον οδηγό εγκατάστασης και αφού γίνει αποδοχή των όρων χρήσης βρισκόμαστε στην παρακάτω οθόνη



Εγκατάσταση Apache

Εδώ θα ζητηθεί η παροχή κάποιων λεπτομερειών σχετικά με τις ρυθμίσεις του διακομιστή όπως το όνομα του το όνομα του domain μας ο λογαριασμός ηλεκτρονικού ταχυδρομείου του διαχειριστή κλπ. Σε περίπτωση που δεν είναι ακόμα γνωστά είτε ο χρήστης βρίσκεται σε επίπεδο αρχαρίου μπορεί να αφήσει τις εξ'ορισμού τιμές οι οποίες μπορούν να αλλάξουν αργότερα.

1) Εφόσον ολοκληρωθεί ο οδηγός εγκατάστασης μένουν ακόμα μερικές απαραίτητες ρυθμίσεις οι οποίες θα πρέπει να λάβουν χώρα αφενός για την λειτουργία του ίδιου του διακομιστή αλλά και την αλληλεπίδραση του με την PHP.

2) Θα πρέπει να σημειωθεί ότι το κύριο αρχείο ρυθμίσεων είναι το httpd.conf και οι ρυθμίσεις που αναφέραμε θα αποθηκευτούν σε αυτό και συνήθως βρίσκεται στον εξής κατάλογο C:\Program Files\Apache Group\Apache2\conf\l. Εδώ μπορούμε να αλλάξουμε τις ρυθμίσεις που αναφέρθηκαν παραπάνω.

3) Τώρα είμαστε πλέον έτοιμοι να ξεκινήσουμε την λειτουργία του διακομιστή. Αυτό μπορεί να γίνει από το μενού έναρξης των windows όπου αφού θα επιλέξουμε τον apache θα εμφανιστεί ένα τερματικό το οποίο θα τερματιστεί από μόνο του.

Πλέον μπορούμε να πληκτρολογήσουμε την εξ ορισμού IP (αν δεν έχει τροποποιηθεί προηγουμένως) 127.0.0.1 στο παράθυρο ενός web browser. Αν το αποτέλεσμα είναι αυτό της επόμενης εικόνας τότε όλα έχουν πάει καλά!



Εγ κατάσταση του Web Server

4) Αυτό που μένει είναι να ρυθμιστεί ο Web Server για αλληλεπίδραση με την PHP. Εντός του αρχείου httpd.conf θα πρέπει να εισάγουμε τα εξής:

1. LoadModule php5_module
2. c:/php// php5apache2.dll

Στην περιοχή του κειμένου που φαίνεται παρακάτω:

```

# Example:
# LoadModule foo_module modules/mod_foo.so
#
LoadModule access_module modules/mod_access.so
LoadModule actions_module modules/mod_actions.so
LoadModule alias_module modules/mod_alias.so
LoadModule asis_module modules/mod_asis.so
LoadModule auth_module modules/mod_auth.so
#LoadModule auth_anon_module modules/mod_auth_anon.so
#LoadModule auth_dbm_module modules/mod_auth_dbm.so
LoadModule auth_digest_module modules/mod_auth_digest.so
LoadModule autoindex_module modules/mod_autoindex.so
#LoadModule cern_meta_module modules/mod_cern_meta.so
LoadModule cgi_module modules/mod_cgi.so
#LoadModule dav_module modules/mod_dav.so
#LoadModule dav_fs_module modules/mod_dav_fs.so
LoadModule dir_module modules/mod_dir.so
LoadModule env_module modules/mod_env.so
#LoadModule expires_module modules/mod_expires.so
#LoadModule file_cache_module modules/mod_file_cache.so
#LoadModule headers_module modules/mod_headers.so
LoadModule imap_module modules/mod_imap.so
LoadModule include_module modules/mod_include.so
#LoadModule info_module modules/mod_info.so
LoadModule isapi_module modules/mod_isapi.so
LoadModule log_config_module modules/mod_log_config.so
LoadModule mime_module modules/mod_mime.so
#LoadModule mime_magic_module modules/mod_mime_magic.so
#LoadModule proxy_module modules/mod_proxy.so
#LoadModule proxy_connect_module modules/mod_proxy_connect.so
#LoadModule proxy_http_module modules/mod_proxy_http.so
#LoadModule proxy_ftp_module modules/mod_proxy_ftp.so
LoadModule negotiation_module modules/mod_negotiation.so
#LoadModule rewrite_module modules/mod_rewrite.so
LoadModule setenvif_module modules/mod_setenvif.so
#LoadModule spelling_module modules/mod_spelling.so
#LoadModule status_module modules/mod_status.so
#LoadModule unique_id_module modules/mod_unique_id.so
LoadModule userdir_module modules/mod_userdir.so
#LoadModule usertrack_module modules/mod_usertrack.so
#LoadModule vhost_alias_module modules/mod_vhost_alias.so
#LoadModule ssl_module modules/mod_ssl.so

```

Πλέον είμαστε έτοιμοι ώστε να δοκιμάσουμε ένα απλό παράδειγμα στην γλώσσα PHP συντάσσοντας το σε οποιοδήποτε επεξεργαστή κειμένου.

2.2 Το σύστημα της Βάσης Δεδομένων

Όπως έχει αναφερθεί αλλά και είναι δεδομένο για να εξασφαλίσουμε την ύπαρξη δυναμικού περιεχομένου σε μία ιστοσελίδα θα πρέπει στο παρασκήνιο τα συστατικά λογισμικού στον διακομιστή να αλληλεπιδρούν με μία βάση δεδομένων. Εδώ επιλέχθηκε σύστημα DB2 της IBM. Πρόκειται για το μόνο πακέτο κλειστού κώδικα που χρησιμοποιήθηκε στην συγκεκριμένη εφαρμογή αν και κυκλοφορεί πληθώρα εκδόσεων που μπορούν να αποκτηθούν χωρίς χρέωση.

2.2 Αλληλεπίδραση με την βάση δεδομένων

Για την σύνδεση με την βάση δεδομένων της εφαρμογής η γλώσσα προγραμματισμού PHP παρέχει αρκετούς τρόπους εκ των οποίων η καλύτερη πρακτική είναι να χρησιμοποιηθεί το πακέτο μεθόδων της db2. Δυστυχώς όμως οι συγκεκριμένες μέθοδοι δεν αποτελούν μέρος μίας εξ ορισμού εγκατάστασης της PHP αλλά είναι επεκτάσεις. Προκειμένου να χρησιμοποιηθούν απαιτείται να γίνει μία νέα εγκατάσταση της php για να συμπεριληφθούν και αυτές οι επεκτάσεις.

Για αυτόν τον λόγο αναγκαστικά επιλέχθηκε η λύση της αλληλεπίδρασης μέσω ODBC σύνδεσης. Οι επεκτάσεις για την σύνδεση με βάση δεδομένων μέσω odbc είναι ενεργοποιημένες εξ ορισμού σε οποιαδήποτε εγκατάσταση της php και το μόνο που απαιτείται για σύνδεση με οποιοδήποτε σύστημα βάσεων δεδομένων είναι να έχουμε μία εγκατάσταση στην μηχανή στην οποία εκτελείται ο php κώδικας προκειμένου να έχει συμπεριληφθεί το σχετικό πρόγραμμα οδήγησης στις odbc services του λειτουργικού συστήματος.

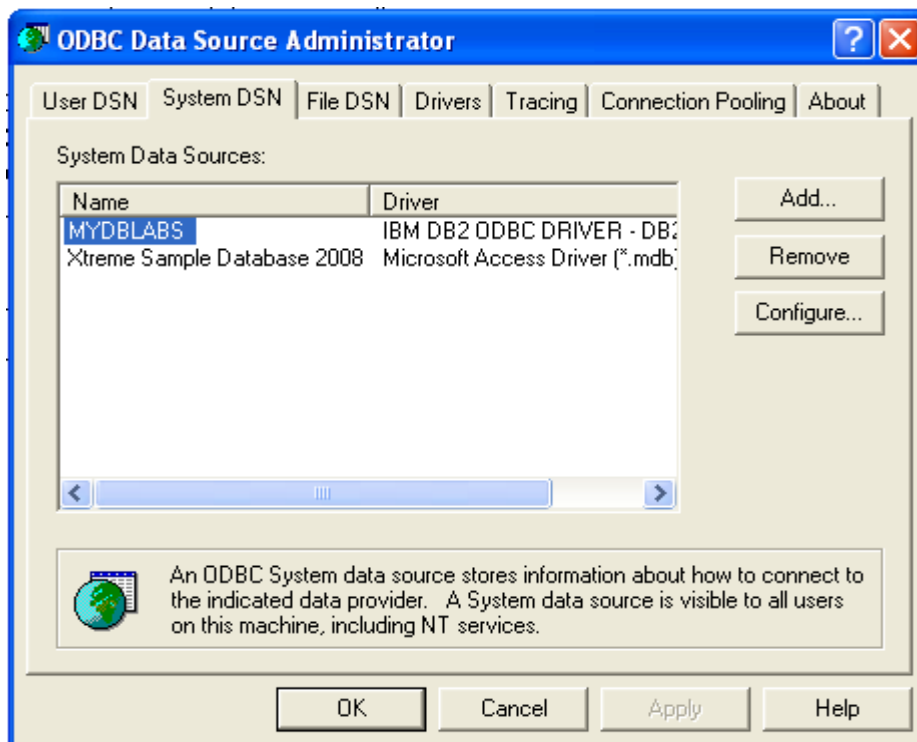
2.2.1 ODBC σύγκριση με άλλες τεχνικές

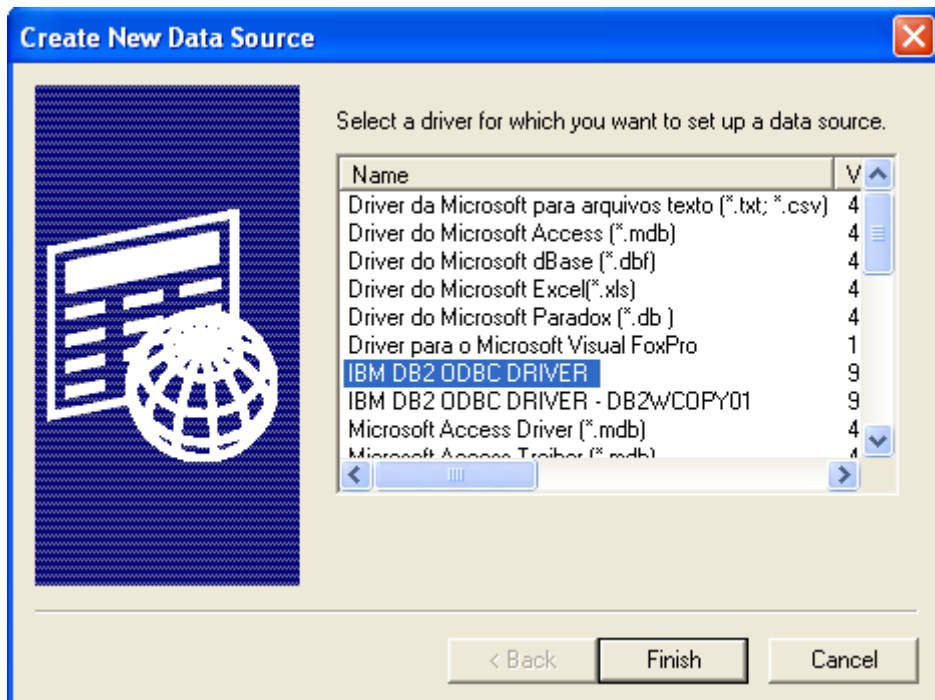
Το πρότυπο ODBC (Open Database Connectivity) παρέχει τις μεθόδους που είναι απαραίτητες προκειμένου να επιτευχθεί η επικοινωνία μεταξύ διαφόρων πλατφορμών λογισμικού. Οι κατασκευαστές του είχαν ως σκοπό την δημιουργία ενός προτύπου το οποίο θα είναι ανεξάρτητο από οποιαδήποτε γλώσσα προγραμματισμού, σύστημα διαχείρισης βάσεων δεδομένων ή λειτουργικό σύστημα.

Σήμερα κυκλοφορεί μία πληθώρα υλοποιήσεων του ODBC προτύπου το που περιέχουν προγράμματα οδήγησης για όλα τα γνωστά συστήματα βάσεων δεδομένων όπως DB2, Oracle, PostgreSQL, Sybase, MySQL κλπ. Ας εξετάσουμε λοιπόν τι απαιτείται προκειμένου να γίνει η σύνδεση και να επιτευχθεί η αλληλεπίδραση με ένα πρόγραμμα PHP.

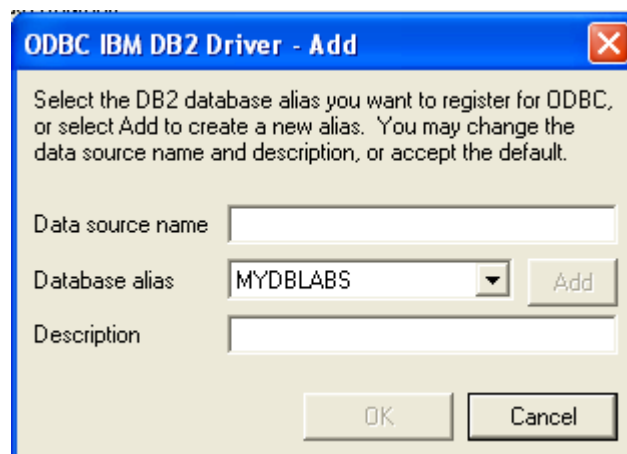
Για αυτόν τον σκοπό αυτό θα πρέπει να δημιουργήσουμε έναν οδηγό με την βοήθεια του διαχειριστή αρχείων και προέλευσης δεδομένων του λειτουργικού μας συστήματος.

- Αφού μεταβούμε στον διαχειριστή συστήματος επιλέγουμε τις πηγές δεδομένων.
- Από εκεί επιλέγουμε την καρτέλα DSN συστήματος και την προσθήκη:





- Εδώ επιλέγουμε τον κατάλληλο οδηγό στην συγκεκριμένη περίπτωση τον IBM DB2 ODBC Driver.
- Αυτό που μένει τώρα είναι να αντιστοιχίσουμε το όνομα της πηγής στο όνομα μίας τοπικής βάσης ή αυτό που χρειάζεται στην δική μας περίπτωση το τοπικό ψευδώνυμο της απομακρυσμένης βάσης δεδομένων.



2.3 Η πλευρά client.

2.3.1 Τεχνολογία AJAX

Σε αυτήν την εφαρμογή έχει μεγάλη σημασία η διαδικασία αποστολής ενός ερωτήματος στην πλευρά του server και αντίστοιχα λήψης και επεξεργασίας της απόκρισης, να μην απαιτεί την επαναφόρτωση ολόκληρης της ιστοσελίδας αλλά μόνο ενός συγκεκριμένου τμήματος πχ το πεδίο στο οποίο θα εμφανίζονται τα αποτελέσματα από ένα ερώτημα προς την βάση δεδομένων. Για αυτό τον λόγο ήταν αναγκαία η δημιουργία και χρήση ενός μηχανισμού Ajax.

Ajax είναι τα αρχικά των Asynchronous Javascript and XML και ο πυρήνας του είναι το αντικείμενο XMLHttpRequest που αποτελεί μέρος του Μοντέλου Αντικειμένων Εγγράφου, πρόκειται για μία αποτελεσματική συλλογή γλωσσών που όταν συγκεντρωθούν δημιουργούν εξαιρετικά περιβάλλοντα για τον χρήστη και διαδραστικές δυνατότητες στην πλευρά του client (όπως αναφέρθηκε και παραπάνω). Προφανώς δεν έχουν όλες οι εφαρμογές ανάγκη για την Ajax αλλά πολλά τμήματα μίας εφαρμογής μπορούν να βελτιωθούν επωφελοόμενες των πλεονεκτημάτων της εφόσον δεν γίνονται υπερβολές.

Επίσης προσφέρει έναν θαυμάσιο τρόπο για να κάνουμε συνδέσεις με προγράμματα που εκτελούνται σε έναν διακομιστή και στην συνέχεια σε μία βάση δεδομένων, να γίνεται καθορισμός του τι θα παρουσιαστεί στον χρήστη να μας επιστρέφονται κωδικοί κατάστασης HTTP και όλα αυτά χωρίς να ανανεωθεί ποτέ η σελίδα.

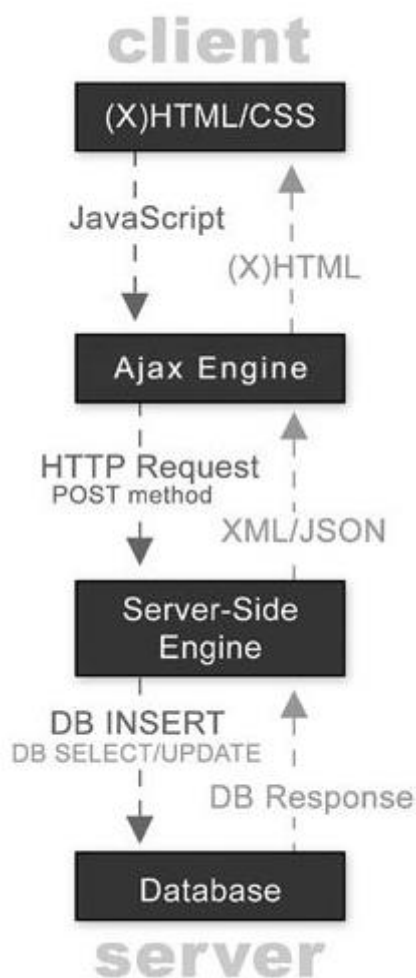
Βέβαια για να γίνουν δυνατά όλα τα παραπάνω χρειάζονται και υποστηρικτικές τεχνολογίες όπως XML και Javascript για την απόδοση των δεδομένων.

2.3.2 Το αντικείμενο XMLHttpRequest

Όπως αναφέρθηκε και πριν το αντικείμενο XMLHttpRequest αποτελεί τον πυρήνα του μηχανισμού AJAX. Είναι το αντικείμενο που επιτρέπει σε μία σελίδα να παίρνει δεδομένα ή να στέλνει δεδομένα στον server και έτσι να επιτρέπει σε μία web εφαρμογή να ενεργεί σαν μία εφαρμογή desktop. Το XMLHttpRequest εξαλείφει την ανάγκη αναμονής για την απόκριση του server σε μία νέα σελίδα για μία αίτηση και επιτρέπει στους χρήστες να συνεχίζουν να αλληλεπιδρούν με την σελίδα ενώ οι αιτήσεις δρουν στο παρασκήνιο.

Για τις αιτήσεις αυτές χρησιμοποιούνται οι μέθοδοι GET και POST και λειτουργούν όπως μία τυπική αίτηση HTTP. Η απόκριση τώρα μπορεί να είναι οποιαδήποτε τυποποιημένη μορφή αλλά οι συνηθέστερες πρακτικές είναι οι XML και Json. Η post μέθοδος είναι ιδιαίτερα χρήσιμη όταν τα δεδομένα που αποστέλλονται δεν υπερβαίνουν τα 512 bytes (ένας όγκος τον οποίο δεν μπορεί να χειριστεί η GET). Αφού ληφθεί μία απόκριση η εφαρμογή μπορεί να συμπληρωθεί με νέα δεδομένα από τον server τα οποία μπορούν να αποδοθούν με τον συνδυασμό Javascript, HTML και CSS.

Όλες οι αιτήσεις στο αντικείμενο XMLHttpRequest χειρίζονται από την Javascript . Η javascript δημιουργεί το αντικείμενο αυτό και υποβάλλει την αίτηση στον server . Στο σχήμα που ακολουθεί θα ρίξουμε μία ματιά στην διαδικασία αυτή:



Στο παραπάνω σενάριο συμβαίνουν τα εξής: Η αποστολή στην βάση δεδομένων ξεκινά μέσω του XHR με μία POST ή μία GET μέθοδο, προς μία γλώσσα ή ένα μηχανισμό πλευράς server που στην δική μας περίπτωση είναι η PHP. Αφού ο server λάβει την αίτηση αναλύει το XML ή το απλό ζεύγος κλειδιού/τιμής που εστάλη και ενημερώνει την βάση δεδομένων αναλόγως. Αυτό το μοντέλο αίτησης ενημερώνει την βάση δεδομένων με βάση την διαδραστική ενέργεια του χρήστη χωρίς και πάλι την ανανέωση του προγράμματος περιήγησης. Αυτή η περίπτωση θα μπορούσε να είναι το πάτημα του κουμπιού αποθήκευσης μίας desktop εφαρμογής.

2.3.3 Η απόκριση μέσω XML

Η XML (Extensible Markup Language) είναι μία δημοφιλής επιλογή για τα XHR's αντικείμενα που χρησιμοποιούνται σε αυτή την εφαρμογή επειδή αποτελεί την τυπική ενδιάμεση γλώσσα την οποία όλες οι γλώσσες προγραμματισμού , όλες οι συλλογές τεχνολογιών και όλα τα συστήματα βάσεων δεδομένων μπορούν να μοιραστούν και αυτό το γεγονός την καθιστά την μακράν πιο ευέλικτη λύση .Πρόκειται ουσιαστικά για μία προσαρμοσμένη δομή βασισμένη σε ετικέτες που ορίζει ο ίδιος ο προγραμματιστής. Η βασισμένη σε ετικέτες δομή της XML είναι παρόμοια με αυτή της HTML με την μόνη διαφορά ότι στην HTML οι ετικέτες είναι προκαθορισμένες και αναπαριστούν την δομή μίας ιστοσελίδας όπως η κεφαλίδα, το σώμα τους πίνακες κλπ.

Η XML μπορεί να χρησιμοποιηθεί επίσης για την επικοινωνία μεταξύ πολλαπλών γλωσσών προγραμματισμού. Το να υπάρχει αυτή η κοινή γλώσσα προσφέρει στον προγραμματιστή εξαιρετική δύναμη. Μας επιτρέπει να δημιουργούμε άμεσες συνδέσεις από το γραφικό περιβάλλον μίας εφαρμογής σε μία γλώσσα server και τελικώς αν θέλουμε σε μία βάση δεδομένων. Θα πρέπει να προσθέσουμε ότι αυτή η προσέγγιση επιτρέπει τον πλήρη διαχωρισμό των 2 επιπέδων της εφαρμογής. Έτσι τα ξεχωριστά τμήματα παραμένουν διαχωρισμένα και η διαχείριση γίνεται ευκολότερη αφού η αλλαγές σε ένα συγκεκριμένο επίπεδο δεν επιφέρουν αναγκαστικά αλλαγές και στα υπόλοιπα.

ΑΠΑΙΤΗΣΕΙΣ ΧΡΗΣΤΩΝ

3.1 Παρουσίαση των εμπλεκόμενων προσώπων

Όσον αφορά τα εμπλεκόμενα πρόσωπα στην εφαρμογή επιλέχθηκε να χωριστούν σε 2 κατηγορίες

A) **Φοιτητές** : Στην περίπτωση μας αποτελούν απλούς χρήστες οι οποίοι έχουν την δυνατότητα απλής πρόσβασης στο περιεχόμενο της εφαρμογής, και επίλυσης των υπαρχόντων παραδειγμάτων.

B) **Καθηγητές** : Που αντιπροσωπεύουν τους διαχειριστές του συστήματος και μπορούν να αλλάξουν το υπάρχον περιεχόμενο ,να προσθέσουν νέο κλπ.

Οι σχετικές λειτουργικές απαιτήσεις για αυτές τις ομάδες χρηστών παρουσιάζονται στην επόμενη ενότητα

3.2 Παρουσίαση απαιτήσεων των ομάδων χρηστών

Για την ομάδα των φοιτητών (απλών χρηστών) έχουμε:

Φοιτητές:

- A) Δυνατότητα εγγραφής στο σύστημα χρησιμοποιώντας τον λογαριασμό ηλεκτρονικού ταχυδρομείου του τμήματος πληροφορικής του ΤΕΙ Θεσ/νίκης.
- B) Επιλογή κάποιας συλλογής ασκήσεων από την διαθέσιμη λίστα και επίλυση των αντιστοίχων ασκήσεων που περιέχονται σε αυτήν.

- Γ) Εμφάνιση των αποτελεσμάτων ενός ερωτήματος που έγινε σε κάποια βάση δεδομένων.
- Δ) Παροχή περαιτέρω πληροφοριών για κάποια συλλογή ασκήσεων μέσω κάποιου συνδέσμου σε εξωτερική πηγή είτε μέσω ανάκτησης ενός σχετικού αρχείου κειμένου.

Καθηγητές:

- A) Είσοδος στο σύστημα μέσω του λογαριασμού του .
- B) Δημιουργία μιας νέας συλλογής ασκήσεων –ενότητας
- Γ) Δημιουργία και ενσωμάτωση ασκήσεων στην επιθυμητή ενότητα.
- Δ) Δυνατότητα οι ασκήσεις να αλληλεπιδρούν με παραπάνω από ένα συστήματα διαχείρισης βάσεων δεδομένων.
- E) Καθορισμός των χρονικών ορίων για τα οποία θα είναι διαθέσιμη (ορατή) στο κοινό μία ενότητα.
- ΣΤ) Διαγραφή ασκήσεων,ενοτήτων.
- Z) Ανανέωση του περιεχομένου και γενικά όλων των χαρακτηριστικών μίας ενότητας (ημερομηνία- ώρα έναρξης και λήξης ,υπάρχουσες ασκήσεις κλπ).
- I) Αποθήκευση συμπληρωματικού ενημερωτικού υλικού όπως σύνδεσμοι σε εξωτερικές πηγές , ανέβασμα αρχείων κειμένου.

Κεφάλαιο 4^ο

Σενάρια Χρήσης

Σε αυτό το κεφάλαιο θα παρουσιαστεί υπό μορφή κειμένου αλλά και με την βοήθεια διαγραμμάτων τι μπορεί να συμβεί κατά την επίσκεψη ενός χρήστη και των 2 προαναφερθέντων ομάδων στον δικτυακό τόπο της εφαρμογής.

Να σημειώσουμε ότι στο παρόν σημείο δεν μας απασχολεί η τεχνική πλευρά της εφαρμογής αλλά απλώς θα πρέπει να εξετάσουμε αν οι περιπτώσεις χρήσης

Αρχικά ας εξετάσουμε την περίπτωση χρήσης της εισόδου ενός φοιτητή στο σύστημα μας.

4.1 Περίπτωση Χρήσης 1: Είσοδος φοιτητή

ΠΧ1:

1) Ο φοιτητής εισέρχεται στον διαδικτυακό τόπο της εφαρμογής και εμφανίζεται η οθόνη εισαγωγής του ονόματος χρήστη και του κωδικού πρόσβασης.

1.1) Ο φοιτητής δεν είναι εγγεγραμμένος στο σύστημα οπότε συμπληρώνει στην σχετική φόρμα τα απαραίτητα στοιχεία και αυτά αποστέλλονται για έλεγχο

1.1.2) Σε περίπτωση που τα στοιχεία δεν είναι αποδεκτά επιστρέφουμε στην αρχική οθόνη και εμφανίζεται η σχετική πληροφορία

1.1.3) Τα στοιχεία είναι αποδεκτά η εγγραφή ολοκληρώνεται

1.1.4) Ο χρήστης πρέπει να ελέγξει τα εισερχόμενα του λογαριασμού ηλεκτρονικού ταχυδρομείου του για να πληροφορηθεί για το όνομα χρήστη και τον κωδικό πρόσβασης που έχουν δημιουργηθεί ειδικά για αυτόν.

1.2) Ο χρήστης εισάγει τα επιθυμητά στοιχεία στην φόρμα και εισέρχεται στο σύστημα.

2) Βρισκόμαστε στην αρχική οθόνη του συστήματος όπου εμφανίζονται οι διαθέσιμες-ενεργές ενότητες ασκήσεων.

2.1) Ο φοιτητής επιλέγει κάποια ενότητα από την λίστα και εμφανίζονται οι αντίστοιχες λεπτομέρειες όπως κάποιο ενημερωτικό αρχείο ή κάποιος σύνδεσμος σε εξωτερική πηγή.

2.2) Εμφανίζονται επίσης οι ασκήσεις που περιέχει η επιλεγμένη ενότητα. Μαζί με τις λεπτομέρειες τους όπως ο αριθμός της άσκησης η εκ'φωνηση κλπ.

2.3) Αφού επιλέξει κάποιο dbms και εισάγει το ερώτημα που θέλει να κάνει προς την βάση ο φοιτητής εκτελεί το ερώτημα.

2.4) Εμφανίζεται ένα μήνυμα αναμονής και εφόσον τα δεδομένα είναι έτοιμα έχουμε την απάντηση υπό μορφή ενός πίνακα.

3) Έξοδος του φοιτητή από το σύστημα και επιστροφή στην αρχική σελίδα.

4.2 Περίπτωση Χρήσης 2 Είσοδος Καθηγητή

ΠΧ2:

- 5) Ο καθηγητής εισέρχεται στον διαδικτυακό τόπο της εφαρμογής και εισάγει το όνομα και τον κωδικό χρήστη στα σχετικά πεδία.
- 6) Τα στοιχεία είναι αποδεκτά και έτσι μεραφέρεται στην αρχική σελίδα της εφαρμογής.

2.1)Εμφανίζονται οι διαθέσιμες ενότητες ασκήσεων

- 3) Ο χρήστης επιλέγει μία από τις διαθέσιμες ενότητες και έτσι εμφανίζονται οι σχετικές λεπτομέρειες για αυτήν, καθώς και οι ασκήσεις που περιέχει.
- 4) Ο χρήστης επιλέγει να τροποποιήσει τα χαρακτηριστικά τις επιλεγμένης ενότητας και τα πεδία που πριν περιείχαν τις λεπτομέρειες διαμορφώνονται ώστε να δεχθούν τα νέα δεδομένα.

4.1) Ο χρήστης εισάγει τα επιθυμητά χαρακτηριστικά στα σχετικά πεδία και επιλέγει την αποθήκευση τους.

4.2) Ο χρήστης επιλέγει την εισαγωγή κάποιας άσκησης στην συγκεκριμένη ενότητα.

4.2.1)Εμφανίζεται η φόρμα εισαγωγής ασκήσεων όπου εισάγονται τη εκφώνηση της άσκησης και τα σωστά ερωτήματα για άθε DBMS.

4.2.2)Μετά από την επιλογή αποθήκευσης της νέας άσκησης ο χρήστης πληροφορείται για το αν όλα πήγαν καλά.

- 5) Ο χρήστης επιλέγει να διαγράψει κάποια άσκηση μέσω του κουμπιού διαγραφής που αντιστοιχεί σε αυτήν.

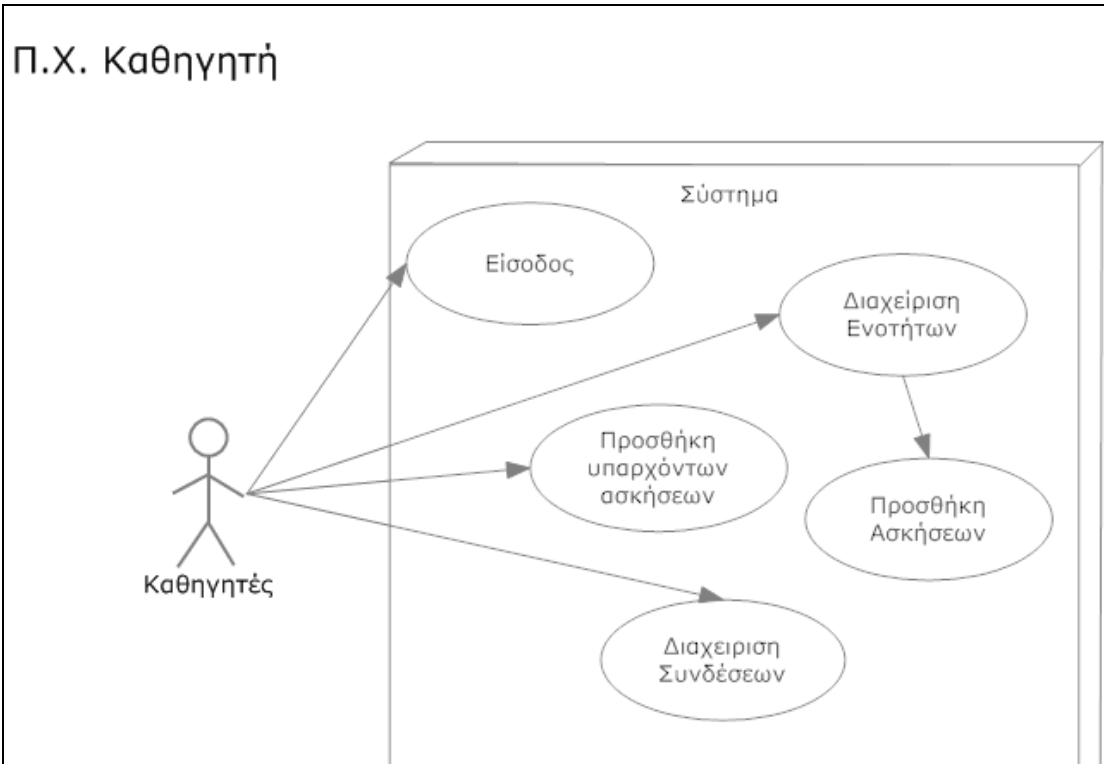
5.1)Αφού το σύστημα εκτελέσει με επιτυχία την συγκεκριμένη ενέργεια εμφανίζεται το σχετικό μήνυμα πληροφόρησης.

6) Ο καθηγητής επιθυμεί να εισάγει μία νέα ενότητα και μετά την επιλογή του εμφανίζεται η σχετική φόρμα.

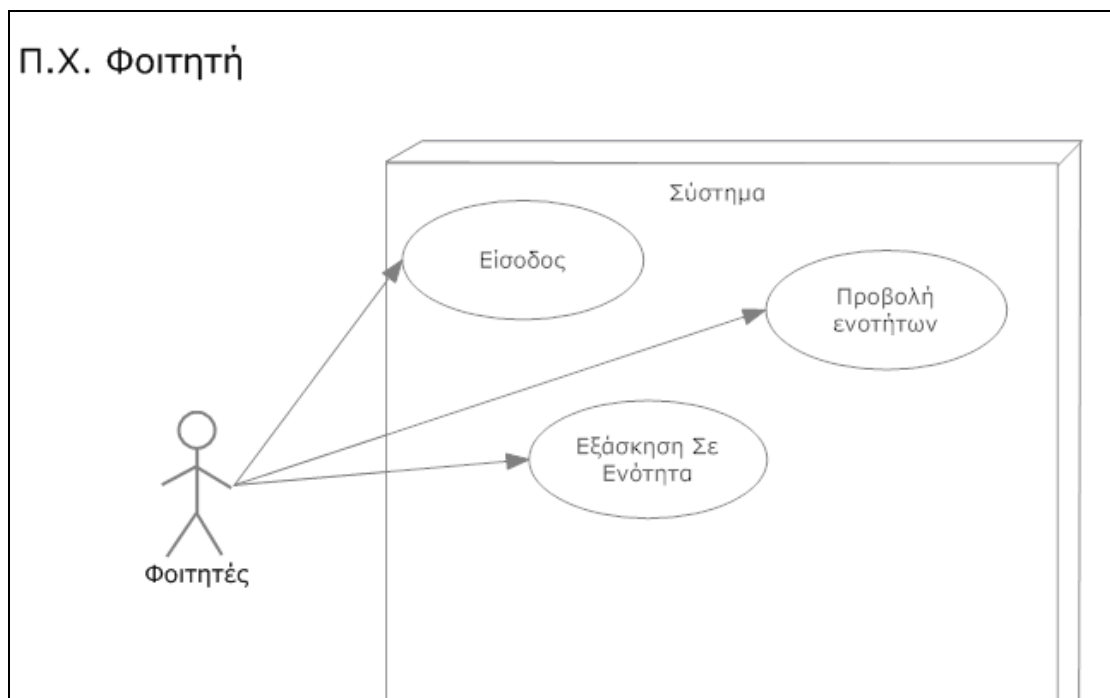
6.1)Σε αυτό το σημείο μπορούν να εισαχθούν το όνομα, οι ημερομηνίες έναρξης και λήξης της ενότητας ο καθορισμός ενός εξωτερικού συνδέσμου., καθώς και η προσκόμιση ενός αρχείου κειμένου που θα περιέχει τις απαραίτητες διευκρινίσεις για τις ασκήσεις της συγκεκριμένης ενότητας.

6.2)Τώρα μετά την επιλογή αποθήκευσης η νέα ενότητα έχει την δυνατότητα να εμπλουτιστεί με υλικό.

Παρακάτω παρουσιάζονται και με την βοήθεια διαγραμμάτων οι περιπτώσεις χρήσης που έχουμε εξετάσει μέχρι τώρα:



Π.Χ. Φοιτητή

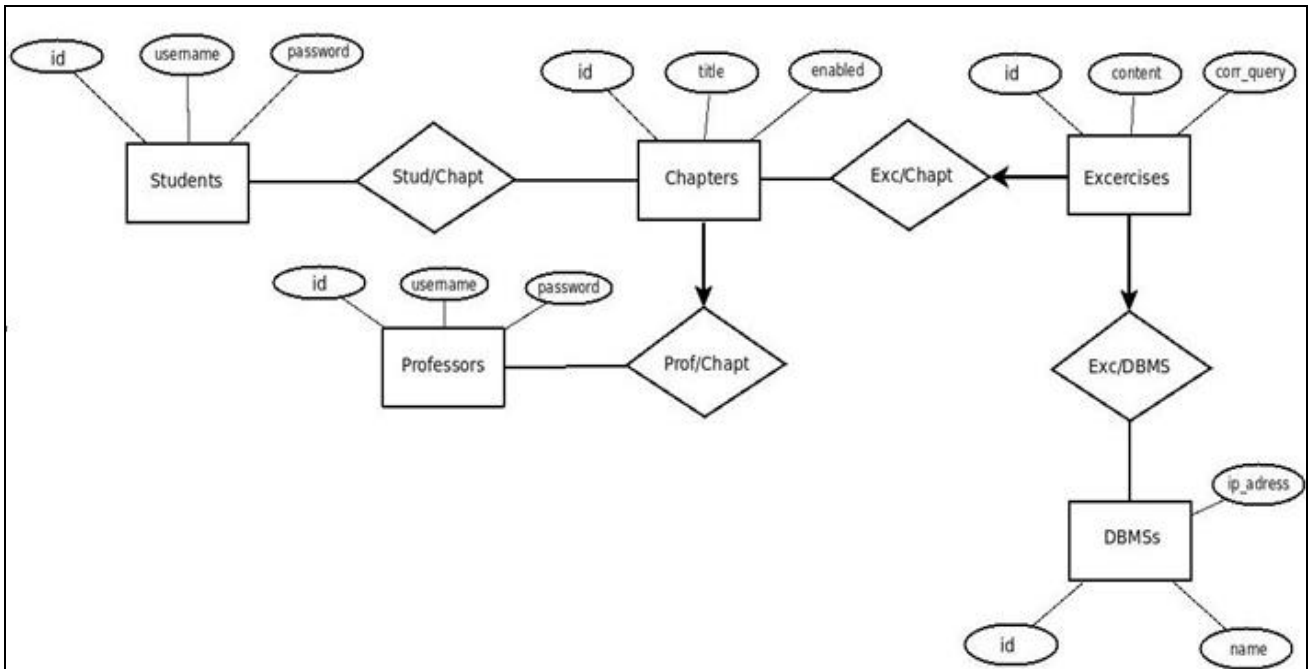


Π.Χ. Φοιτητή

Προδιαγραφές συστήματος

5.1 Το σχεσιακό σχήμα της βάσης δεδομένων

Η εικόνα που φαίνεται παρακάτω παρουσιάζει το σχεσιακό σχήμα της βάσης δεδομένων της εφαρμογής. Σε αυτήν την ενότητα θα εξηγήσουμε τις σχέσεις μεταξύ των πινάκων καθώς και τα πεδία που περιέχουν:



Το ER της εφαρμογής

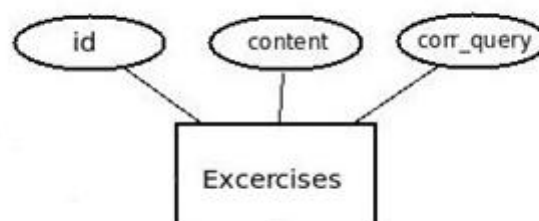
Σε πρώτη φάση θα περιγράψουμε τους πίνακες των καθηγητών , εργαστηρίων , ασκήσεων (εκπαιδευτικού υλικού).



Εδώ έχουμε τον πίνακα που αποθηκεύονται τα στοιχεία των καθηγητών με κύριο κλειδί το id (οποίο έχει οριστεί να παράγεται αυτόματα όπως και όλα τα πεδία id-κύρια κλειδιά όλων των πινάκων).



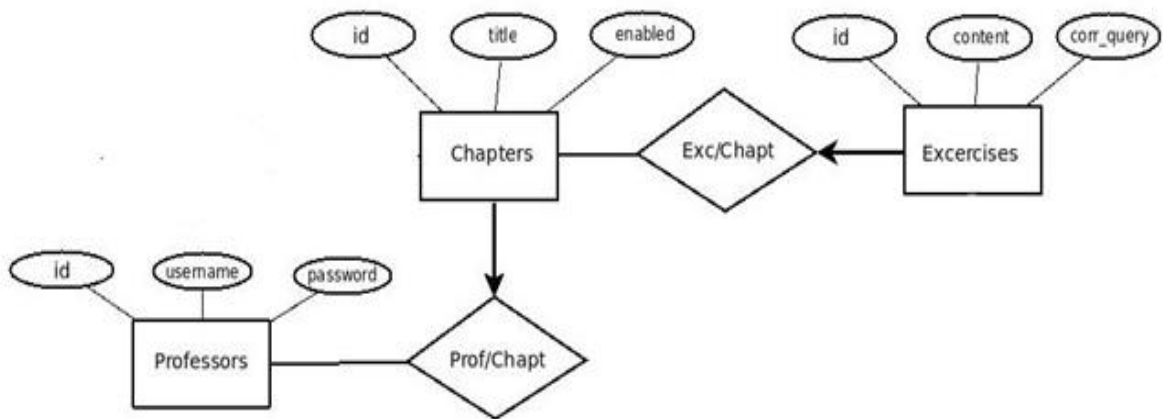
Ο πίνακας των εργαστηρίων (εκπαιδευτικών ενοτήτων) καταχωρεί τον τίτλο του εργαστηρίου, (πεδίο title) , επίσης το πεδίο enabled θα χρησιμοποιείται για να καθορίζεται αν το εργαστήριο θα είναι διαθέσιμο στους φοιτητές.(Σε περίπτωση που είναι ακόμα υπο κατασκευή).Ως κύριο κλειδί χρησιμοποιείται το πεδίο id.



Τα πεδία του πίνακα των ασκήσεων περιγράφουν την εκφώνηση της άσκησης και το σωστό ερώτημα. Επίσης εδώ καταχωρούνται και

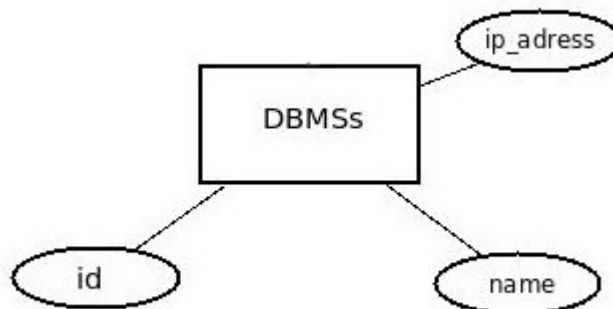
χαρακτηριστικά απο το επιστρεφόμενο set αποτελεσμάτων για να ελέγχεται η ορθότητα των εντολών που καταχωρούν οι φοιτητές.

Οι συσχετίσεις μεταξύ αυτών των πινάκων φαίνονται στο παρακάτω σχήμα:



Βλέπουμε ότι στον πίνακα Exercises έχει δοθεί ολική συμμετοχή στην συσχέτιση του με τον πίνακα Chapters διότι οι ασκήσεις που είναι καταχωρημένες υποχρεωτικά θα ανήκουν σε κάποια ενότητα και σε περίπτωση διαγραφής κάποιας ενότητας θα πρέπει να διαγράφονται όλες οι ασκήσεις που περιλαμβάνει.

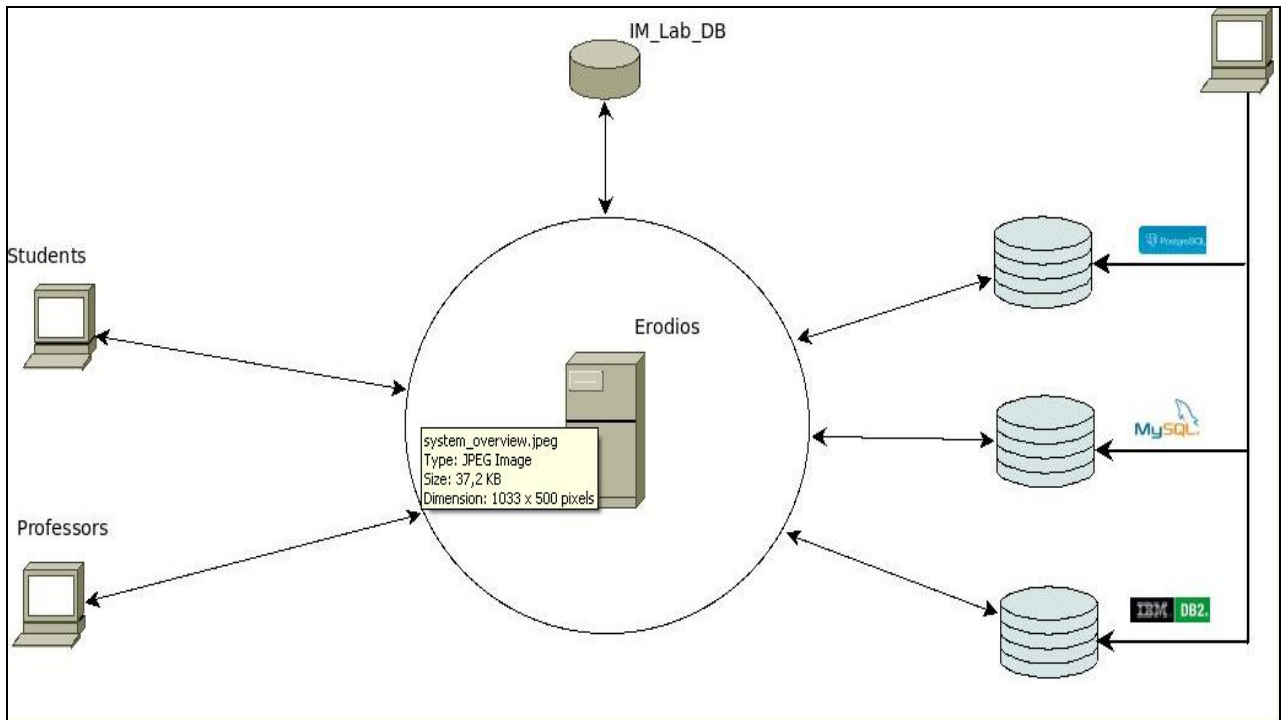
Ο επόμενος πίνακας που έχουμε είναι αυτός στον οποίο αποθηκεύουμε τις πληροφορίες προκειμένου να γίνεται η σύνδεση σε κάποια απομακρυσμένη βάση δεδομένων:



Το πεδίο IP address ουσιαστικά περιέχει μία συμβολοσειρά την οποία εισάγουμε στις μεθόδους που πραγματοποιούν μία σύνδεση μέσω odbc.

5.2 Αρχιτεκτονική Γενική Μορφή Συστήματος

Ας ρίξουμε τώρα μία ματιά στο πως είναι καταναμημένοι οι διάφοροι πόροι που θα χρησιμοποιεί η εφαρμογή:

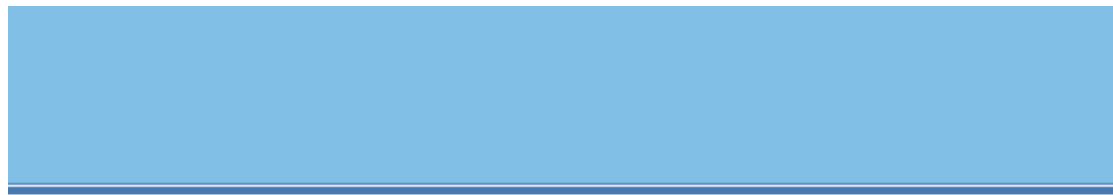


Βλέπουμε οτι ως κεντρικό διακομιστή χρησιμοποιούμε το μηχάνημα εροδιός το οποίο φιλοξενεί τα αρχεία πηγαίου κώδικα της εφαρμογής. Τα τερματικά αριστερά αναπαριστούν την δυνατότητα απομακρυσμένης σύνδεσης των χρηστών με τον εροδιό. Επίσης η βάση δεδομένων της εφαρμογής βρίσκεται και αυτή σε άλλο μηχάνημα και η σύνδεση και αλληλεπίδραση από τον εροδιό γίνεται μέσω του προτύπου ODBC. Οι υπόλοιπες βάσεις δεδομένων που βλέπουμε σε αυτό το διάγραμμα μας υποδεικνυουν την δυνατότητα σύνδεσης και αλληλεπίδρασης με οποιοδήποτε σύστημα διαχείρισης βάσεων δεδομένων υποστηρίζει το πρότυπο ODBC και έχει οριστεί από τους διεχειριστές του συστήματος είτε τους καθηγητές.

Λειτουργική Παρουσίαση Συστήματος

Έχοντας εξετάσει κάποια εισαγωγικά θέματα σχετικά με την φύση και τις απαιτήσεις της εφαρμογής ήρθε η ώρα δούμε κάποιες από τις διαθέσιμες λειτουργίες της όπως θα τις αντιλαμβάνεται και ο τυπικός χρήστης. Ας ξεκινήσουμε λοιπόν από την πλευρά του καθηγητή:

Βρισκόμαστε στην αρχική οθόνη όπου ο χρήστης μπορεί να εισέλθει στο σύστημα δίνοντας το όνομα του και τον κωδικό του:



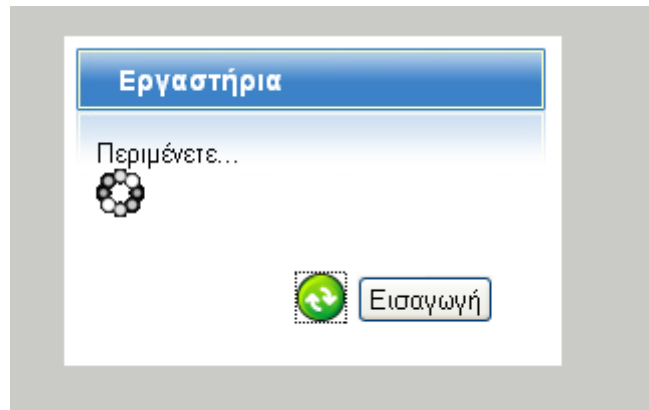
Login Here

Username

Password

Οθόνη Εισόδου.

Σε περίπτωση που τα στοιχεία είναι εσφαλμένα επιστρέφουμε σε αυτήν την οθόνη. Διαφορετικά μεταφερόμαστε στην αρχική σελίδα της εφαρμογής, όπου θα δούμε και τα διαθέσιμα εργαστήρια τις διαθέσιμες ασκήσεις κλπ.



Αρχικά αναμένουμε για ένα μικρό διάστημα προκειμένου να φτάσουν τα δεδομένα των διαθέσιμων εργαστηρίων μόλις γίνει αυτό εμφανίζεται η σχετική λίστα:



Στην παραπάνω εικόνα βλέπουμε επίσης οτι παρέχονται και 2 κουμπιά προκειμένου να ανανεώσουμε το υλικό σε περίπτωση που έγιναν κάποιες τροποποιήσεις και να εισάγουμε μία νέα ενότητα. Για να εισάγουμε μία νέα ενότητα εμφανίζεται η σχετική φόρμα:

Ενσωματωμένα Δοχεία

Δημιουργία Νέου Εργαστηρίου

Τίτλος:

Ημερομηνία/Ωρα Ενεργοποίησης:

Ημερομηνία/Ωρα Λήξης:

Καταχώρηση Κλείσιμο

Εδώ μπορούμε να δημιουργήσουμε μία νέα ενότητα με τα επιθυμητά χαρακτηριστικά καθώς και να ορίσουμε το χρονικό διάστημα και απο ποιά ημερομηνία και μετά θα είναι διαθέσιμη η συγκεκριμένη ενότητα για τους φοιτητές:

Επιλεγμένο

May 2010

May 2010						
Mo	Tu	We	Th	Fr	Sa	Su
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31						

16 : 43 : 37

Δημιουργία Νέου Εργαστηρίου

Τίτλος:

Ημερομηνία/Ωρα Ενεργοποίησης:

Ημερομηνία/Ωρα Λήξης:

Καταχώρηση Κλείσιμο

Εφόσον οι αλλαγές πραγματοποιηθούν με επιτυχία εμφανίζεται το σχετικό μήνυμα για να μας πληροφορήσει και αφού κάνουμε μία ανανέωση στο πεδίο των εργαστηρίων το νέο είναι έτοιμο να δεχθεί υλικό.

Επιλέγοντας ένα εργαστήριο εμφανίζονται οι λεπτομέρειες του καθώς και οι διαθέσιμες ασκήσεις του και ενεργοποιείται η δυνατότητα εισαγωγής νέας άσκησης:

Επιλεγμένο Εργαστήριο :

Εργαστήριο 2

Ημερομηνία/Ωρα Έναρξης	2010-04-05 21:05:07
Ημερομηνία/Ωρα Λήξης	2010-04-30 12:12:10
Ενημερωτικό Αρχείο	Δεν υπάρχει αρχείο
Σύνδεσμος Σε Εξωτερική Πηγή	Δεν υπάρχει κάποιος σύνδεσμος

Τροποποίηση Αποθήκευση Ακύρωση

Βλέπουμε ότι δεν έχει οριστεί-μεταφορτωθεί ακόμα κάποιο αρχείο που να περιέχει ενημερωτικό υλικό σχετικά με αυτήν την ενότητα επιλέγοντας λοιπόν τροποποίηση τα πεδία των λεπτομερειών παίρνουν την κατάλληλη μορφή για εισαγωγή νέων στοιχείων.

Επιλεγμένο Εργαστήριο :

Ημερομηνία/Ωρα Έναρξης	<input style="width: 150px; height: 20px;" type="text"/>
Ημερομηνία/Ωρα Λήξης	<input style="width: 150px; height: 20px;" type="text"/>
Ενημερωτικό Αρχείο	File: <input style="width: 100px;" type="text"/> Αναζήτηση... Upload
Σύνδεσμος Σε Εξωτερική Πηγή	<input style="width: 150px; height: 20px;" type="text"/>

Τροποποίηση Αποθήκευση

Τώρα πλέον μπορούμε να αλλάξουμε το όνομα της επιλεγμένης ενότητας να ορίσουμε άλλες ημερομηνίες έναρξης και λήξης καθώς και να μεταφορτώσουμε κάποιο αρχείο και να ορίσουμε έναν σύνδεσμο σε κάποια άλλη πηγή.

Αξίζει να σημειωθεί ότι για την μεταφόρτωση κάποιου αρχείου δεν απαιτείται η ανανέωση της σελίδας αλλά απλώς ανανεώνεται το συγκεκριμένο τμήμα μόνο αυτό επιτυγχάνεται με την χρήση ενός κρυφού πεδίου και η τεχνική αυτή θα παρουσιαστεί στο επόμενο κεφάλαιο.Ακόμα πρέπει να πούμε ότι επιτρέπονται μόνο αρχεία κειμένου και συγκεκριμένα μόνο κατάληξης *.doc ή *.pdf.Η διαδικασία μεταφόρτωσης φαίνεται παρακάτω:

Επιλεγμένο Εργαστήριο :

Ημερομηνία/Ωρα Έναρξης

Ημερομηνία/Ωρα Λήξης

Ενημερωτικό Αρχείο File: C:\Documents and Settings\... Αναζήτηση...

Σύνδεσμος Σε Εξωτερική Πηγή

Σε περίπτωση που παρουσιαστεί για κάποιο λόγο σφάλμα πχ γιατί ο τύπος του αρχείου δεν είναι αποδεκτός είτε για κάποιον λόγο του σφάλματος του δικτύου εμφανίζεται το κατάλληλο μήνυμα για να μας πληροφορήσει σχετικά:

Υπήρξε ένα σφάλμα κατά την επεξεργασία του αρχείου

Ενημερωτικό Αρχείο

File: C:\Documents and Settings\Αναζήτηση... Upload

Σφάλμα κατά την μεταφότωση

Παρακάτω με χρήση της σχετικής επιλογής μπορούμε να προσθέσουμε μία νέα άσκηση στην επιλεγμένη ενότητα με την βοήθεια της φόρμας εισαγωγής ασκήσεων:

Ημερομηνία/Ωρα Λήξης
Ενημερωτικό Αρχείο

Εισα...

Προσθήκη Άσκησης

Εκφώνηση:

Σωστό Ερώτημα:
DB2:

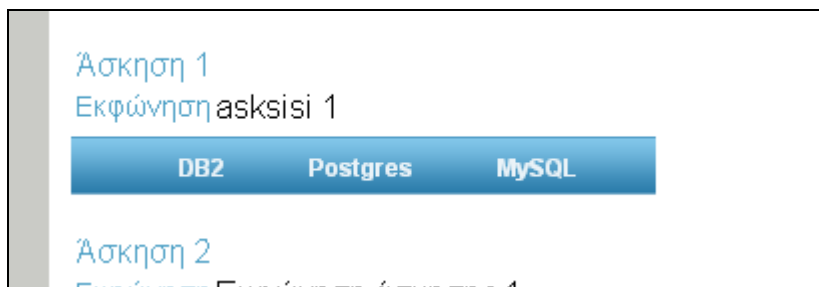
PostgreSQL:

MySQL:

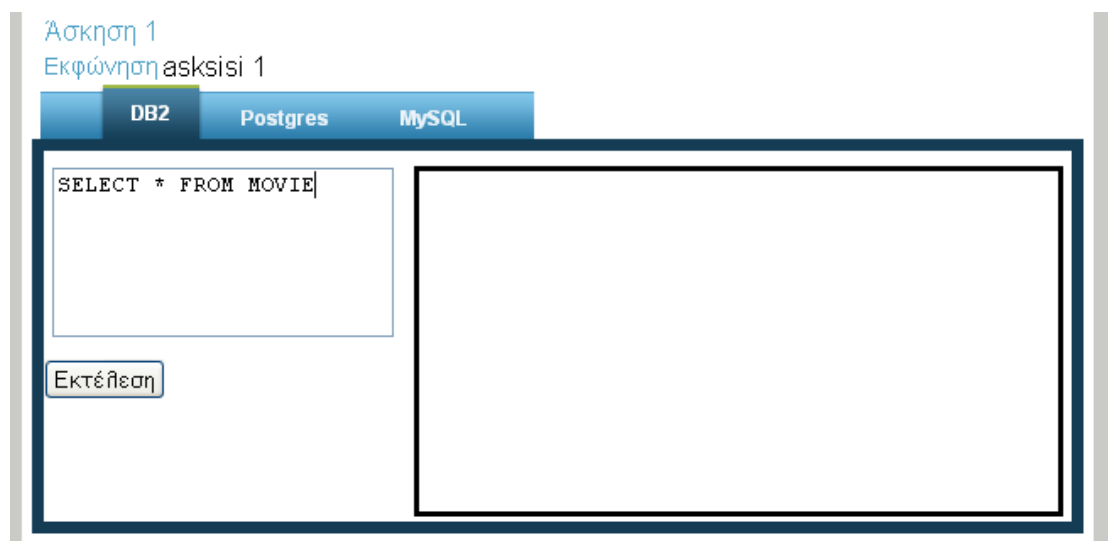
Καταχώρηση Κλείσιμο

Εδώ δίνουμε την επιθυμητή εκφώνηση καθώς και τα ερωτήματα που επιστρέφουν το ζητούμενο αποτέλεσμα για αυτήν την άσκηση στα 3 DBMS που υποστηρίζονται:

Παρακάτω από τις λεπτομέρειες της ενότητας εμφανίζεται η λίστα των διαθέσιμων ασκήσεων:

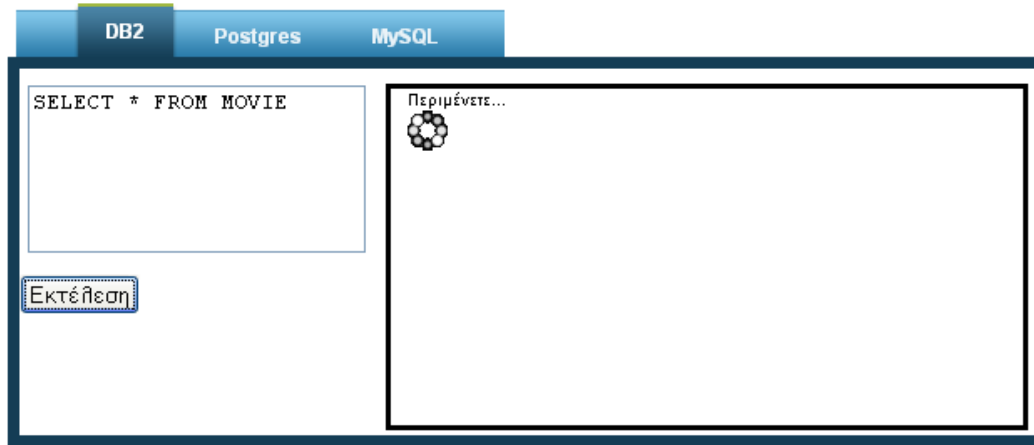


Εδώ βλέπουμε την εκφώνηση των ασκήσεων και μπορούμε επιλέγοντας κάποια από τις καρτέλες να δούμε το σχετικό ερώτημα και να το εκτελέσουμε προκειμένου να αποθηκευθεί ο επιστρεφόμενος αριθμός σειρών και στηλών του σετ αποτελεσμάτων για να χρησιμοποιηθεί στην σελίδα των φοιτητών όσον αφορά την ορθότητα των ερωτημάτων που εκτελούν.

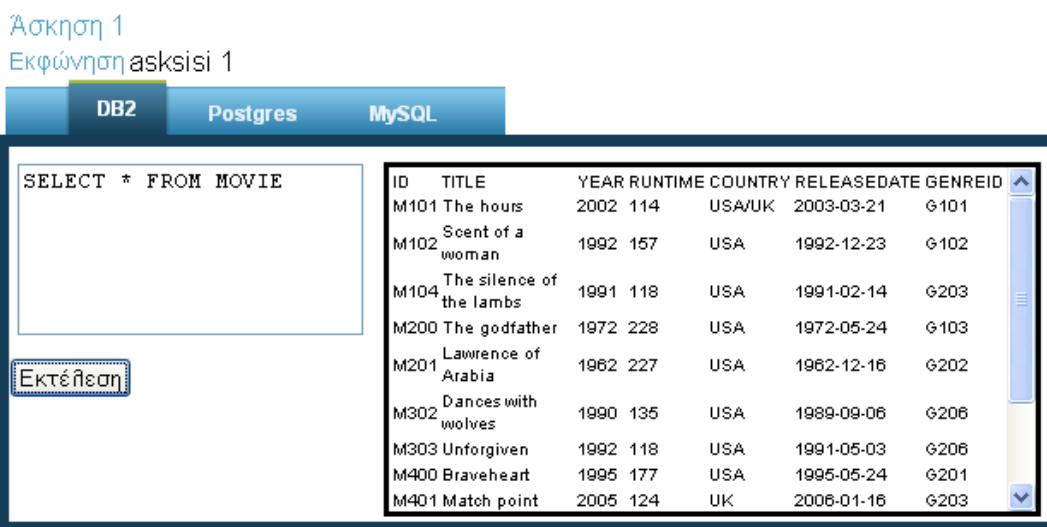


Εκτελώντας λοιπόν το συγκεκριμένο ερώτημα εμφανίζεται ένα μήνυμα αναμονής για να μας δείξει ότι η διαδικασία έχει ξεκινήσει :

Άσκηση 1
Εκφώνηση asksisi 1



Όταν τα αποτελέσματα έχουν επιστρέψει και είναι έτοιμα παρουσιάζονται με την μορφή πίνακα:



ID	TITLE	YEAR	RUNTIME	COUNTRY	RELEASEDATE	GENREID
M101	The hours	2002	114	USA/UK	2003-03-21	G101
M102	Scent of a woman	1992	157	USA	1992-12-23	G102
M104	The silence of the lambs	1991	118	USA	1991-02-14	G203
M200	The godfather	1972	228	USA	1972-05-24	G103
M201	Lawrence of Arabia	1962	227	USA	1962-12-16	G202
M302	Dances with wolves	1990	135	USA	1989-09-06	G206
M303	Unforgiven	1992	118	USA	1991-05-03	G206
M400	Braveheart	1995	177	USA	1995-05-24	G201
M401	Match point	2005	124	UK	2006-01-16	G203

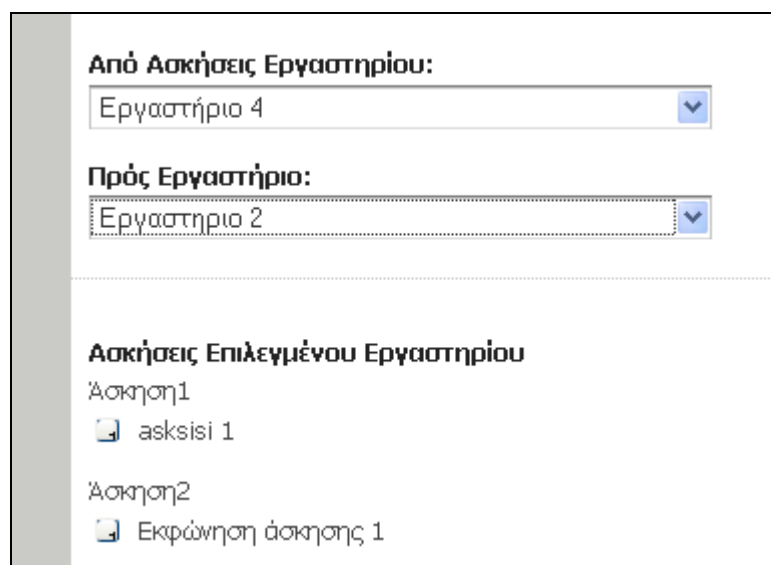
Μία ακόμη δυνατότητα που θα εξετάσουμε είναι ο εμπλουτισμός μίας ενότητας από τις ήδη υπάρχουσες ασκήσεις μίας άλλης. Χρησιμοποιώντας τον κατάλληλο σύνδεσμο:



Μεταβαίνουμε στην σελίδα που περιέχει την κατάλληλη φόρμα για αυτήν την λειτουργία:

A screenshot of a web form titled 'Εισαγωγή διαθέσιμων ασκήσεων από άλλη ενότητα'. Below the title is a subtitle: 'Επιλέξτε την ενότητα από την οποία θέλετε να μεταφέρετε ασκήσεις και αυτήν που θέλετε να τις προσθέσετε.' The form contains two dropdown menus. The first is labeled 'Από Ασκήσεις Εργαστηρίου:' and has 'Εργαστήριο 1' selected. The second is labeled 'Πρός Εργαστήριο:' and also has 'Εργαστήριο 1' selected. Below these dropdowns is a section titled 'Ασκήσεις Επιλεγμένου Εργαστηρίου' and a button labeled 'Προσθήκη'.

Εδώ τα 2 combo-box φορτώνονται με τα ονόματα των διαθέσιμων ενοτήτων και σε περίπτωση που στο πρώτο έχουμε αλλαγή τιμής εμφανίζονται οι ασκήσεις που περιέχει στο δεύτερο combo box επιλέγουμε το εργαστήριο το οποίο θα δεχθεί τις ασκήσεις που θέλουμε:



The screenshot shows a web interface with the following elements:

- Από Ασκήσεις Εργαστηρίου:** A dropdown menu with the selected value "Εργαστήριο 4".
- Πρός Εργαστήριο:** A dropdown menu with the selected value "Εργαστήριο 2".
- Ασκήσεις Επιλεγμένου Εργαστηρίου:** A list of exercises for the selected laboratory, including:
 - Άσκηση1
 - asksisi 1
 - Άσκηση2
 - Εκφώνηση άσκησης 1

Αυτό που μένει τώρα είναι η επιλογή κάποιας άσκησης για προσθήκη μέσω του κουμπιού που βρίσκεται δίπλα στην εκφώνηση της.

Τεχνική Παρουσίαση Συστήματος

7.1 Επισκόπηση

Σε αυτό το κεφάλαιο θα ασχοληθούμε με τον τρόπο που λειτουργούν τα βασικότερα και πιο ενδιαφέροντα τμήματα της εφαρμογής σε επίπεδο πηγαίου κώδικα. Θα συζητηθεί ο τρόπος κατασκευής των αντικειμένων javascript στην μεριά του χρήστη που αναλαμβάνουν να πραγματοποιούν τις αιτήσεις προς τον server, πώς γίνεται η επεξεργασία αυτών των αιτήσεων στην μεριά του server και πως η ρηρ αλληλεπιδρά με την βάση δεδομένων της εφαρμογής. Ακόμα θα ερευνήσουμε την μορφή των δεδομένων που αποστέλλονται πίσω στο πρόγραμμα περιήγησης του χρήστη και πως γίνεται η απόδοση τους μέσω των CSS. Ακόμα θα δούμε πως υλοποιείται τελικά το σχεσιακό σχήμα της βάσης δεδομένων της εφαρμογής που είδαμε στο κεφάλαιο “Προδιαγραφές Συστήματος».

7.2 Δημιουργία του μηχανισμού Ajax

Για την δημιουργία και διαχείριση των αιτήσεων προς τον server ,αλλά και για τον χειρισμό και την απόδοση των επιστρεφόμενων δεδομένων εντός της ιστοσελίδας συντάχθηκαν ορισμένα αντικείμενα με την βοήθεια της γλώσσας Javascript. Ας τα αναφέρουμε αρχικά τα 2 σημαντικότερα:

- **Ajax:** Αυτό το αντικείμενο αναλαμβάνει τον χειρισμό όλων των αιτήσεων προς τον server.
- **AjaxUpdater:** Αντικείμενο το οποίο περιλαμβάνει τις διάφορες μεθόδους στις οποίες γίνεται προσπέλαση από την υπόλοιπη εφαρμογή. Χρησιμοποιείται κυρίως ώστε να προσδώσουμε ένα επίπεδο αφαιρετικότητας/

Για να δημιουργήσουμε το αντικείμενο Ajax χρειάζεται πρώτα να το δομήσουμε με τον εξής τρόπο:

```
Ajax = {};
```

Εφόσον το αντικείμενο δημιουργηθεί μπορούμε πλέον να προσαρτήσουμε τις μεθόδους που χρειαζόμαστε. Δηλαδή μεθόδους για δημιουργία και αποστολή των αιτήσεων , έλεγχος της κατάστασης μίας αίτησης , και προσπέλασης στα αποτελέσματα. Αρχικά για την μέθοδο δημιουργίας και αποστολής του αντικειμένου XHR:

```
Ajax.makeRequest= function(method,url,callbackMethod)
{
    this.request=(window.XMLHttpRequest)? new XMLHttpRequest():new ActiveXObject("MSXML.XMLHTTP");
    this.request.onreadystatechange=eval(callbackMethod);
    this.request.open(method,url,true);
    this.request.send(url);
}
```

Εδώ αρχικά εξετάζουμε τον τύπο του browser που χρησιμοποιεί ο χρήστης και αναλόγως δημιουργούμε ένα αντικείμενο XHR με τον κατάλληλο τρόπο. Μετά θέτουμε το συμβάν "onreadystatechange" στην μέθοδο ανάκλησης (της μεθόδου που θα εκτελεστεί σε περίπτωση που η απόκριση είναι έτοιμη) και είναι το τρίτο όρισμα αυτής της μεθόδου. Τα επόμενα στάδια είναι η χρήση των μεθόδων open και send. Η μέθοδος open ξεκινά την αίτηση, καθορίζει την πηγή που θα την χειριστεί στην πλευρά του server και θέτει μία παράμετρο Boolean σε true. Αυτή η παράμετρος καθρίζει αν η μετάδοση αυτή θα είναι σύγχρονη ή ασύγχρονη. Προφανώς στην περίπτωση μας δεν θέλουμε να υποβάλλουμε ασύγχρονες αιτήσεις αφού θα παγώνουν την εφαρμογή μας μέχρι να ληφθεί η απόκριση. Η μέθοδος send τώρα χρησιμοποιείται για να στείλει την αίτηση HTTP στον server και να περιμένει για να λάβει την απόκριση.

Για να ελέγξουμε πότε μία αίτηση έχει ληφθεί από τον server και είναι έτοιμη για διαχείριση χρησιμοποιείται η ιδιότητα `readyState`. Παραπάνω αναφέραμε το συμβάν “`onreadystatechange`” και ότι θέτουμε σε αυτό το ξεκίνημα κάποιας ενέργειας. Αυτό το συμβάν πυροδοτείται από τις αλλαγές της κατάστασης της αίτησης HTTP και αυτές τις καταστάσεις τις περιέχει η ιδιότητα `readyState`. Έτσι για να ελέγχεται η κατάσταση της αίτησης χρησιμοποιούμε την παρακάτω μέθοδο

```
Ajax.checkReadyState= function()  
{  
    switch(this.request.readyState)  
    {  
        case 1:  
            break;  
        case 2:  
            break;  
        case 3:  
            break;  
        case 4:  
            AjaxUpdater.isUpdating = false;  
            return HTTP.status(this.request.status);  
    }  
}
```

Εδώ εξετάζεται κάθε κατάσταση από την οποία περνάει η αίτηση μας προς τον διακομιστή. Μπορούμε ανάμεσα στις δηλώσεις συνθήκης να τροφοδοτούμε ένα στοιχείο της σελίδας ώστε να παρέχουμε μία οπτική πληροφορία στον χρήστη για την πορεία της αίτησης του. Αργότερα σε αυτό το κεφάλαιο κατά την επέκταση αυτού του μηχανισμού εξηγείται πώς μπορούμε να αντιστοιχίσουμε αυτές τις εκτελέσεις υπό συνθήκη με τους κωδικούς κατάστασης HTTP.

Τέλος απομένει να δούμε την μέθοδο που μας επιστρέφει την απόκριση από τον διακομιστή. Πρέπει να σημειώσουμε ότι η απόκριση δεν είναι απαραίτητο να καταφθάνει σε μορφή κειμένου οπότε θα πρέπει να γίνεται και ένας έλεγχος σχετικά με τον τύπο της.

```

Ajax.getResponse = function()
{
    if(this.request.getResponseHeader('Content-Type').indexOf('xml') != -1)
    {
        return this.request.responseXML.documentElement;
    }
    else
    {
        return this.request.responseText;
    }
}
.

```

Το μόνο που έχουμε να κάνουμε προκειμένου να προσπελάσουμε την απόκριση μίας αίτησης είναι να καλέσουμε την παραπάνω μέθοδο. Έχουμε και μία συνθήκη η οποία εξετάζει την φύση του περιεχομένου της απόκρισης.Γίνεται ένας έλεγχος στην κεφαλίδα και σε περίπτωση που βρεθεί ότι υπάρχει ο δείκτης ακολουθίας χαρακτήρων XML (Content-Type = XML) επιστρέφεται στην μέθοδο που θα την χειριστεί με τον κατάλληλο τρόπο ώστε να υπάρχει συμβατότητα.διαφορετικά περνάει στην περνάει ως απλό κείμενο.

Πλέον απομένει να δημιουργηθεί ακόμα ένα αντικείμενο το οποίο θα εξυπηρετεί στο να κρατήσουμε ένα επίπεδο αφαιρετικότητας το AjaxUpdater. Σε αυτό το αντικείμενο θα γίνεται η προσπέλαση από τα διάφορα συστατικά της εφαρμογής μας και επίσης θα περιέχει τις μεθόδους με τις οποίες θα πραγματοποιούμε την απόδοση των δεδομένων σε οπτικά στοιχεία για τον χρήστη.Αρχικά θα πρέπει να δομηθεί με τον κλασσικό τρόπο.

AjaxUpdater = {};

Θα πρέπει επίσης να του αποδώσουμε και κάποιες αρχικές τιμές ώστε να είναι δυνατή η προσπέλαση του από άλλα στοιχεία.:

```

AjaxUpdater.initialize = function()
{
    AjaxUpdater.isUpdating = false;
}

AjaxUpdater.initialize();

```

Τώρα θα πρέπει να δηλώσουμε και μία μέθοδο η οποία θα αλληλεπιδρά με το αντικείμενο Ajax και θα χειρίζεται τις επιθυμητές αιτήσεις. Ουσιαστικά χρειαζόμαστε 3 παραμέτρους. Την μέθοδο που θα χρησιμοποιούμε προκειμένου να υποβάλλουμε τις παραμέτρους (GET, POST), την δήλωση της υπηρεσίας που θα χρησιμοποιηθεί στον server δηλαδή το URL του αρχείου server-side κώδικα το οποίο θα χειριστεί την αίτηση και τέλος προεραϊτικά μία μέθοδο που θα χειριστεί την απόκριση στην πλευρά του client. Προαιρετικά γιατί θα δηλώσουμε και μία εξ ορισμού μέθοδο χειρισμού της απόκρισης αν δεν δηλωθεί κάποια άλλη μέθοδος ανάκλησης.

```
AjaxUpdater.Update = function(method, service, callback)
{
    if(callback == undefined || callback == "")
    {
        callback = AjaxUpdater.onResponse;
    }
    Ajax.makeRequest(method, service, callback);
    AjaxUpdater.isUpdating = true;
}
```

Η τελευταία μέθοδος φαίνεται παρακάτω και είναι η default μέθοδος ανάκλησης. Ουσιαστικά σε περίπτωση που δεν έχουμε δηλώσει κάποια άλλη γίνεται κλήση κλήση αυτής και η μόνη λειτουργία που πραγματοποιεί είναι να εξετάζει την ιδιότητα "checkReadyState" αντικειμένου Ajax και σε περίπτωση που η απόκριση έχει ληφθεί να γνωστοποιεί ότι η διαδικασία έχει τελειώσει.

```
AjaxUpdater.onResponse = function()
{
    if(Ajax.checkReadyState('loading') == "OK")
    {
        AjaxUpdater.isUpdating = false;
    }
}
```

Μέχρι εδώ έχουμε εξετάσει όλα τα απαιτούμενα του Ajax μηχανισμού που υπάρχει σε αυτήν την εφαρμογή παρακάτω θα δούμε πως λειτουργεί στην πράξη μέσω κάποιων επιλεγμένων τμημάτων της εφαρμογής και επίσης πως επεκτάθηκε αυτός ο μηχανισμός για να βοηθήσει στον χειρισμό των αποκρίσεων. Προφανώς όμως θα πρέπει πρώτα να συζητηθούν και κάποια τμήματα κώδικα που η εκτέλεση τους γίνεται από την μεριά του server.

7.3 Επέκταση του μηχανισμού Ajax.

Μέχρι τώρα είδαμε τις απαιτήσεις σε επίπεδο κώδικα προκειμένου να αποστέλλονται οι αιτήσεις προς τον server και να γίνεται ο χειρισμός των αποκρίσεων στην πλευρά του client. Όμως υπάρχουν ακόμα 2 αντικείμενα τα οποία δημιουργήθηκαν προκειμένου να παρέχουν κάποια βοήθεια σχετικά αφ ενός με τον χειρισμό των δεδομένων της απόκρισης αφετέρου σχετικά με την πληροφόρηση της πορείας μίας αίτησης προς τον server. Αυτά τα αντικείμενα είναι:

- **Utilities:** Με το συγκεκριμένο αντικείμενο γίνεται ο χειρισμός κάποιων συνηθισμένων λειτουργιών που πρέπει να λάβουν χώρα στην μεριά του client όπως η δημιουργία HTML στοιχείων ,και η απόδοση ιδιοτήτων σε αυτά ,η προσάρτηση κάποιων στοιχείων σε κάποιο τμήμα της εφαρμογής και η αφαίρεση κάποιων τμημάτων από την εφαρμογή.
- **HTTP:** Το οποίο χρησιμοποιείται για τον χειρισμό κωδικών κατάστασης HTTP προκειμένου να παρέχεται στον χρήστη πληροφόρηση για την πορεία των αιτήσεων του.

Ξεκινώντας με το αντικείμενο Utilities δημιουργούμε κλασικά ένα στιγμιότυπο του.

```
function Utilities(){};
```

Αρχικά θα δούμε την μέθοδο η οποία χρησιμοποιείται για την δημιουργία HTML στοιχείων:

```
Utilities.createElement = function(e,obj)
{
    var element=document.createElement(e)

    for(prop in obj)
    {
        element[prop]=obj[prop];
    }

    return element;
}
```

Αυτή η μέθοδος δέχεται 2 ορίσματα. Το πρώτο είναι το όνομα του στοιχείου που επιθυμούμε να δημιουργηθεί και το δεύτερο ένας πίνακας ο οποίος περιέχει τις ιδιότητες τις οποίες θέλουμε να περιέχει το νέο στοιχείο..Σε δεύτερο στάδιο χρησιμοποιείται η global μέθοδος createElement η οποία δημιουργεί στοιχεία δυναμικά.αυτά τα μπορεί να είναι οποιαδήποτε HTML στοιχεία όπως div,img,li,a κλπ. Τέλος γίνεται ένα σκανάρισμα του πίνακα που περιέχει τις επιθυμητές ιδιότητες και γίνεται η ανάθεση τους στο νέο στοιχείο. Ακολουθεί η μέθοδος με την οποία γίνεται προσάρτηση απογόνων (child elements) σε ένα στοιχείο.


```

Utilities.appendChild = function()
{
    if(this.appendChild.arguments.length>1)
    {
        var a=this.appendChild.arguments[0];
        for(i=1;i<this.appendChild.arguments.length;i++)
        {
            a.appendChild(this.appendChild.arguments[i]);
        }
        return a;
    }
    else
    {
        return null;
    }
}

```

Η μέθοδος appendChild χρησιμοποιεί την ιδιότητα της Javascript arguments η οποία πρόκειται για έναν πίνακα που περιέχει όλες τις παραμέτρους εισόδου για την τρέχουσα μέθοδο και μας επιτρέπει να δημιουργούμε μεθόδους που μπορούν να αλλάξουν με βάση τις παραμέτρους. Επίσης έχει δημιουργηθεί μία μέθοδος για την διαγραφή των στοιχείων απογόνων από ένα επιλεγμένο στοιχείο:

```

Utilities.removeChild = function(node)
{
    if(node===null)
    {
        return;
    }

    while (node.hasChildNodes())
    {
        node.removeChild(node.firstChild);
    }
}

```

Η μέθοδος removeChildren παίρνει ως όρισμα ένα στοιχείο και με την χρήση ενός βρόχου while αφαιρεί όλα τα στοιχεία απογόνους του.

Τέλος για το αντικείμενο Utilities έχουμε μία πολύ χρήσιμη μέθοδο η οποία προσαρτά σε ένα επιλεγμένο στοιχείο listeners για διάφορα συμβάντα.

```
Utilities.addListener = function(obj, eventName, listener)
{
    if(obj.attachEvent)
    {
        var id = obj.uniqueID
        obj.attachEvent("on"+eventName,
            function() {
                return listener.call(document.all[id], event)
            }
        )
    }
    else if(obj.addEventListener)
    {
        obj.addEventListener(eventName, listener, false);
    }
    else
    {
        return false;
    }
    return true;
}
```

Ως ορίσματα εδώ έχουμε το στοιχείο στο οποίο θέλουμε να προσθέσουμε έναν listener (obj) το όνομα του συμβάντος για το οποίο θέλουμε να πραγματοποιείται η παρακολούθηση (eventName) και τέλος την παράμετρο listener η οποία αντιπροσωπεύει την μέθοδο που θέλουμε να εκτελείται. Μετά εξετάζεται ποιά μέθοδος προσάρτησης υποστηρίζεται από τον browser του χρήστη attachEvent ή addEventListener (οι 2 πιο δημοφιλείς browser IE και Firefox ακολουθούν διαφορετικές υλοποιήσεις). Τέλος ανάλογα με τον τρόπο που υποστηρίζεται προσαρτάται και ο listener για το αντίστοιχο συμβάν.

Τώρα όσον αφορά το αντικείμενο HTTP δημιουργήθηκε για να εξυπηρετήσει 2 σκοπούς. Πρώτον είναι χρήσιμο να έχουμε γνώση των επιστρεφόμενων κωδικών κατάστασης στο παρασκήνιο για τους σκοπούς της ανάπτυξης, αλλά προσφέρουν και στον χρήστη πληροφορίες σχετικά με την κατάσταση των ενεργειών τους. Οι κωδικοί κατάστασης HTTP παρουσιάζονται παρακάτω:

- Πληροφοριακό: 1xx.
- Επιτυχές: 2xx.
- Ανακατεύθυνση: 3 xx.
- Σφάλμα Client: 4 xx.
- Σφάλμα Server : 5 xx.

Τα xx μετά από κάθε αριθμό αναπαριστούν άλλα ψηφία που υπάρχουν σε κάθε κατηγορία και χρησιμοποιούνται για να καθορίσουν κάποιους συγκεκριμένους τύπους σφάλματος. Ας δείξουμε λοιπόν τις μεθόδους που περιλαμβάνει το αντικείμενο HTTP:

Αρχικά έχουμε την μέθοδο κατάστασης η οποία λαμβάνει την παράμετρο status και εντοπίζει την αντίστοιχη μέθοδο με βάση τον αριθμό κατηγορίας της. Για παράδειγμα αν η μέθοδος λάβει τον κωδικό 200 θα εντοπίσει τον αριθμό 2 και θα μεταβιβάσει τον κωδικό σε αυτή την μέθοδο.

```
HTTP.status = function(_status)
{
    var s = _status.toString().split("");
    switch(s[0])
    {
        case "1":
            return this.getInformationalStatus(_status);
            break;
        case "2":
            return this.getSuccessfulStatus(_status);
            break;
        default:
            return "An unexpected error has occurred.";
    }
}
```

Για να συνεχίσουμε το παράδειγμα αυτό θα αναφερθεί μόνο η μέθοδος χειρισμών κωδικών επιτυχούς κατάστασης. Αυτή η κατηγορία κωδικών αποτελείται από 6 κωδικούς κατάστασης:

- OK : 200

- Created:201
- Accepted:202
- Non Authorative Information:203
- NO Content:204
- Reset Content: 205
- Partial Content : 206

Όπως είπαμε η μέθοδος παίρνει ως όρισμα την κατάσταση του αντικειμένου Ajax και εκτυπώνει το κατάλληλο μήνυμα.την έξοδο της μπορούμε να την χειριστούμε σε κάποια άλλη μέθοδο είτε να την παρουσιάσουμε στον χρήστη.

```

HTTP.getSuccessfulStatus = function(_status)
{
    //Χειρισμός των κωδικών επιτυχίας
    switch(_status)
    {
        case 200:
            return "OK";
            break;
        case 201:
            return "Created";
            break;
        case 202:
            return "Accetped";
            break;
        case 203:
            return "Non-Authorative Information";
            break;
        case 204:
            return "No Content";
            break;
        case 205:
            return "Reset Content";
            break;
        case 206:
            return "Partial Content";
            break;
        default:
            return "An unexpected error has occured";
    }
}

```

7.5 Ο server-side κώδικας

Εξετάζοντας τώρα την μεριά του server θα πρέπει αρχικά να τονίσουμε ότι τα αρχεία php χωρίζονται σε 3 λογικές κατηγορίες:

- Τις κλάσεις που αναπαριστούν τις ομάδες των χρηστών.
- Τους controllers. Με αυτόν τον όρο αναφέρομαι στα αρχεία php που αναλαμβάνουν να γεφυρώσουν το χάσμα ανάμεσα στον client side κώδικα και τις κλάσεις που αναφέρθηκαν παραπάνω. Δηλαδή να παραλάβουν μία αίτηση από τον Ajax μηχανισμό και να την προωθήσουν στο κατάλληλο αντικείμενο που έχει δημιουργηθεί ανάλογα με την φύση του χρήστη.
- Τέλος υπάρχουν κάποια βοηθητικά αρχεία που περιέχουν πληροφορίες για τις συνδέσεις με την βάση δεδομένων της εφαρμογής, για την είσοδο ενός χρήστη στο σύστημα κλπ.

7.5.1 Οι κλάσεις των ομάδων χρηστών.

Ανάλογα με την φύση του χρήστη (κάτι το οποίο κρίνεται κατά την είσοδο του στο σύστημα) δημιουργείται ένα αντικείμενο που καθορίζει τις επιτρεπτές ενέργειες του. Σε περίπτωση που αυτός ο χρήστης είναι καθηγητής-διαχειριστής την στιγμή που θα πραγματοποιήσει το login δημιουργείται ένα στιγμιότυπο της κλάσης Professor. Αυτή η κλάση περιέχει μεθόδους σχετικές με την δημιουργία και διαχείριση των ενοτήτων, δημιουργία ασκήσεων κλπ. Αντίθετα σε περίπτωση που έχουμε έναν απλό χρήστη (έναν φοιτητή) δημιουργείται ένα στιγμιότυπο της κλάσης Student το οποίο παρέχει απλώς μεθόδους σχετικές με την εκτέλεση των παραδειγμάτων. Σε αυτό το στάδιο θα παρουσιάστούν μόνο οι μέθοδοι δόμησης των 2 αυτών αντικείμενων και στην πορεία θα αποκλυφθούν ολόκληρα (λόγω του όγκου τους κρίνω καλύτερο να τα παρουσιάσω μέσω παραδειγμάτων).

```

require_once 'class.db.php';
class Professor
{

    function Professor($username, $password, $id)
    {
        $this->username=$username;
        $this->password=$password;
        $this->id=$id;
    }
    function getUsername()
    {
        return $this->username;
    }

    function getPassword()
    {
        return $this->password;
    }
}

```

Η κλάση Professor

```

require_once 'class.db.php';

class Student
{
    private $username;
    private $password;

    function Student($username, $password)
    {
        $this->username=$username;
        $this->password=$password;
    }
}

```

Η κλάση Student

7.5.3 Σύνδεση με την Βάση Δεδομένων της Εφαρμογής

Για την σύνδεση με την βάση δεδομένων της εφαρμογής αλλά και με τις υπόλοιπες βάσεις δεδομένων που θα χρειαστεί να εκτελεστούν τα ερωτήματα κάποιων ασκήσεων τα παραπάνω αντικείμενα δημιουργούν ένα στιγμιότυπο του αντικειμένου db. Το συγκεκριμένο αντικείμενο περιέχει τις πληροφορίες που είναι απαραίτητες προκειμένου να επιτευχθεί μία σύνδεση ODBC με οποιαδήποτε βάση δεδομένων όπως το όνομα της βάσης, η πόρτα που θα χρησιμοποιηθεί στο μηχανήμα που είναι εγκατεστημένο κάποιο DBMS, τα σχετικά username και password κλπ.

```
class db
{
    private $username="IMLabAdmin";
    private $password="*****";

    function dbConnect()
    {
        $database = sprintf("DRIVER={IBM DB2 ODBC DRIVER};DATABASE=IMLabDB;HOSTNAME=192.168.16.99;PORT=50");
        $connection=odbc_connect($database,$this->username,$this->password);
        return $connection;
    }
}
```

Αντικείμενο DB για σύνδεση με βάσεις δεδομένων

7.5.4 Απαγόρευση προσπέλασης με χρήση μεταβλητών Session

Ως γνωστόν σε αυτόν τον δικτυακό τόπο έχουμε περιεχόμενο στο οποίο δεν θέλουμε να επιτρέπουμε την πρόσβαση. Αυτό επιτυγχάνεται εν μέρει μέσω της ταυτοποίησης του χρήστη κατά την είσοδο του στο σύστημα αλλά θα πρέπει να δημιουργηθούν και άλλα επίπεδα ασφαλείας προκειμένου να κρατήσουμε τον διαχωρισμό στο επίπεδο και στις δυνατότητες των χρηστών. Αυτός ο σκοπός έχει επιτευχθεί με την βοήθεια Session μεταβλητών. Ένα παράδειγμα του πως μπορεί να γίνει αυτό είναι το εξής:

- Τοποθέτηση ενός αρχείου index.php σε κάθε κατάλογο της εφαρμογής.
- Η προσπάθεια προσπέλασης σε έναν φακελο οδηγεί εξ ορισμού στο συγκεκριμένο αρχείο (σε περίπτωση που υπάρχει σε έναν κατάλογο)
- Στα αρχεία index.php γίνεται έλεγχος αν έχει αρχικοποιηθεί μία μεταβλητή πχ. Authenticated .
- Η μεταβλητή αυτή παίρνει μία τιμή πχ ok κατά την είσοδο ενός χρήστη στο σύστημά .
- Στην δική μας περίπτωση εφόσον τα στοιχεία του χρήστη είναι σωστά δημιουργείται ένα στιγμιότυπο αντικειμένου χρήστη πχ Professor και γίνεται έλεγχος για αυτό.

Ένα παράδειγμα του πως μπορεί να γίνει αυτό φαίνεται παρακάτω


```

<?php
if (isset($_POST['user']) && $_POST['user']=='Username'
    && isset($_POST['pass']) && $_POST['pass']=='secret')
{
    session_start();
    $_SESSION['authorized']=='ok';
    $url = (isset($_GET['url'])) ? $_GET['url']:
        'index.php';
    header("Location:$url");
}
?>

```

Στο παραπάνω παράδειγμα ελέγχονται τα στοιχεία που απεστάλησαν από μία φόρμα και στην συνέχεια αν δεν είναι αληθή το script επιστρέφει τον χρήστη στο url από το οποίο είχε μεταβεί εκεί αρχικά διαφορετικά θέτει την τιμή της session μεταβλητής Authorized σε “ok”.

7.5.5 Διαδικασία Εγγραφής

Στο προηγούμενο κεφάλαιο “Λειτουργική Παρουσίαση Συστήματος” έγινε μία περιγραφή της διαδικασίας εγγραφής εδώ θα δούμε με ποιόν τρόπο υλοποιήθηκε.

Για την αποστολή των email που περιέχουν το όνομα χρήστη και των κωδικό πρόσβασης για το σύστημα ως smtp server χρησιμοποιήθηκε αυτός της google. Για αυτόν τον λόγο δημιουργήθηκε ένας λογαριασμός ηλεκτρονικού ταχυδρομείου στο domain gmail , με όνομα χρήστη DBLabs. Τώρα ο κώδικας που απαιτείται για την υλοποίηση αυτής της διαδικασίας βρίσκεται σε 2 αρχεία. Το SignUp.php στο οποίο βρίσκεται ο κώδικας για την δημιουργία του κειμένου του email και απευθύνεται η φόρμα εγγραφής και το gmail.php μία κλάση στην οποία δηλώνεται ένα αντικείμενο που περιέχει τις

απαραίτητες πληροφορίες για τις σχετικές δηλώσεις (smtp server, smtp port , κλπ) .

Σχετικά με την διαδικασία εγγραφής τώρα αρχικά αφού ληφθεί ο λογαριασμός ηλεκτρονικού ταχυδρομείου που απαιτείται για την εγγραφή ελέγχεται αν το domain του αντιστοιχεί σε αυτό του τμήματος πληροφορικής.

```
$username=${_POST['newid']};  
$domain=substr($username, '@');
```

Πρώτα εξάγουμε το τμήμα της διεύθυνσης που βρίσκεται μετά το @ και μετά κάνουμε τον έλεγχο που αναφέραμε

```
if($domain!="@it.teithe.gr")  
{  
    echo "<p>Εισάγετε μία διεύθυνση email με domain του τμήματος πληροφορικής</p>";  
    exit;  
}
```

Σε περίπτωση που η διεύθυνση που δόθηκε δεν είναι αποδεκτή η διαδικασία σταματάει και τυπώνεται το σχετικό μήνυμα. Διαφορετικά δημιουργείται ένα νέο password που ουσιαστικά είναι μία ακολουθία χαρακτήρων:

```
$username_db=substr($username, '@', true);  
$newpass = substr(md5(time()), 0, 6);
```

Και είμαστε πλέον έτοιμοι να αποστείλουμε τον mail στην διεύθυνση που δόθηκε από τον χρήστη:

```

$gmail = new gmail;
//Ορίζουμε την ηλεκτρονική διεύθυνση του παραλήπτη
$gmail->setReceiver($username);

$headers = array(
    'to:name' . $gmail->receiver, //To header 'to:' πρέπει να πάρει το ίδιο e-
    'subject:Registration DBLabs', //To θέμα του e-mail
    'Content-type: text/plain;charset=utf-8', //Ο τύπος του e-mail. (Για αλλαγ
);

$text= sprintf("Τα στοιχεία σας είναι τα εξής:
                username=%s
                password=%s", $username, $newpass);

if ($gmail->sendMail($usernamegmail, $password, $headers, $text)) {
    echo "<p>ok</p>";
}
else
{
    echo "<p></p>";
}

```

Βλέπουμε ότι έχουμε μία δημιουργία ενός αντικειμένου τύπου gmail (αυτό το αντικείμενο θα αναλάβει να θέσει την διεύθυνση του παραλήπτη και να αποστείλει το email) και σε δεύτερη φάση την σύνταξη του email και αποστολή του με την μέθοδο sendMail η οποία παίρνει ως ορίσματα τον λογαριασμό gmail ο οποίος σε αυτή την περίπτωση έχει τον ρόλο του SMTP username και το αντίστοιχο password τις κεφαλίδες που περιέχουν το θέμα και τον παραλήπτη και τέλος το κείμενο που περιέχει. Το σώμα της μεθόδου sendMail φαίνεται παρακάτω:

```

function sendMail($user, $pass, $headers, $text) {

    if (!$this->checkCode(220))
        die('Σφάλμα: Αδυναμία σύνδεσης με τον smtp server του gmail');

    if (!$this->sendGreet())
        die('Σφάλμα: Πρόβλημα επικοινωνίας με τον smtp server του gmail');

    if (!$this->auth($user, $pass))
        die('Σφάλμα: Λανθασμένα στοιχεία λογαριασμού gmail');

    $this->sendData('MAIL FROM:<>' . $this->lineterm);
    if (!$this->checkCode(250))
        return false;

    $this->sendData('RCPT TO:<' . $this->receiver . '>' . $this->lineterm);
    if (!$this->checkCode(250))
        return false;

    $this->sendData('DATA' . $this->lineterm);
    if (!$this->checkCode(354))
        return false;

    $this->sendData(implode($this->lineterm, $headers) . $this->lineterm);
    $this->sendData($this->lineterm);
    $this->sendData($text . $this->lineterm);
    $this->sendData('.') . $this->lineterm);
    if (!$this->checkCode(250))
        return false;
}

```

Βλέπουμε ότι γίνονται οι απαραίτητοι έλεγχοι σχετικά με την δυνατότητα σύνδεσης με τον smtp server, αν τα στοιχεία smtp username και smtp password είναι έγκυρα κλπ. Και σε περίπτωση που κάτι απ' όλα αυτά πάει στραβά η εκτέλεση σταματάει και μας επιστρέφεται μία false Boolean μεταβλητή ώστε να πληροφορηθεί ο χρήστης ότι η αποστολή απέτυχε.

7.5.5.1 Ολοκλήρωση Εγγραφής

Όπως είπαμε και πριν στην περίπτωση που το email που παρέχει ο χρήστης είναι αποδεκτό παράγεται ένα password και αποστέλλεται στην διεύθυνση ηλεκτρονικού ταχυδρομείου που έδωσε. Αν τώρα η διεύθυνση ανταποκρίνεται σε μία πραγματική μόνο έτσι θα μπορέσει να δει το password και το username τα οποία αποθηκεύτηκαν στην βάση δεδομένων και τελικά να πραγματοποιήσει είσοδο στο σύστημα.

```
$statement= sprintf("INSERT INTO STUDENTS (username,password) VALUES ('%s','%s')", $username_db, $newpass);  
odbc_exec($conn,$statement);
```

Ημερομηνία: Sat, 15 May 2010 04:22:52 -0700 [15/05/2010 14:22:52 EEST]

Από: dblabsteithe@gmail.com

Προς: nametharmen@it.teithe.gr

Θέμα: Registration DBLabs

Τα στοιχεία σας είναι τα εξής:

username=tharmen@it.teithe.gr
password=cfd61b

Το mail που τελικά λαμβάνει ο χρήστης

γεγονός ότι το εργαστήριο που έχει επιλεγθεί συσχετίζεται με το όνομα του αρχείου που πρόκειται να μεταφορτωθεί.

```
$query=sprintf("UPDATE CHAPTERS SET UPLOADED_FILE='%s' where ID=%d",$_FILES['myfile']['name'],$selected_lab);  
  
$statement=odbc_prepare($conn,$query);
```

Μετά ορίζουμε τον κατάλογο που θα αποθηκευτεί το επιθυμητό αρχείο ως τον κατάλογο που βρίσκεται το αρχείο upload.php και ελέγχουμε ότι ο τύπος του είναι αποδεκτός:

```
$target_path = $destination_path . basename( $_FILES['myfile']['name']);  
  
if((($_FILES['myfile']['type']=="application/msword")||($_FILES['myfile']['type']=="application/pdf"))  
{
```

Σε περίπτωση που αυτός ο έλεγχος είναι επιτυχής μπορούμε πλέον να μεταφορτώσουμε το αρχείο και να κάνουμε την σχετική εγγραφή στην βάση δεδομένων:

```
if(@move_uploaded_file($_FILES['myfile']['tmp_name'], $target_path))  
{  
    odbc_exec($conn,$query);  
  
    printf( "<script type='text/javascript' src='../page_scripts.js'></script>  
        <script>  
            function stop(msg)  
            {  
                parent.document.getElementById('upload_target').onload=stopUpload(msg);  
            }  
  
            window.onload=stop('ok'); </script>");  
}
```

Βλέπουμε ότι το παραγόμενο μήνυμα προς επιστροφή περιέχει μία javascript μέθοδο που θα ξεκινήσει την διαδικασία τερματισμού της μεταφόρτωσης και θα πληροφορήσει τον χρήστη ότι όλα πήγαν καλά. Η

μεθοδος αυτη stopUpload παρνει ως ορισμα την επιστρεφόμενη τιμή από τον ρηρ κώδικα και τα περιεχόμενα της παρουσιάζονται παρακάτω:

```
function stopUpload(results)
{
    if(results=="ok")
    {
        parent.document.getElementById('result').innerHTML = '<span class="msg">Το αρχείο αποθηκεύτηκε επιτυχώ
    }
    else
    {
        parent.document.getElementById('result').innerHTML = '<span class="emsg">Υπήρξε ένα σφάλμα κατά την επ
    }
    parent.document.getElementById('f1_upload_process').style.visibility = 'hidden';
}
```

Εδώ ανάλογα με την τιμή του επιστρεφόμενου μηνύματος φορτώνουμε σε ένα span στοιχείο το κατάλληλο περιεχόμενο για να πληροφορήσουμε τον χρήστη σχετικά με την έκβαση της διαδικασίας μεταφόρτωσης.