

1. Εισαγωγή

Τα τελευταία χρόνια, ο χρόνος έχει εισβάλει δυναμικά στη ζωή μας. Κάθε μέρα που περνάει έχουμε την απαίτηση να κάνουμε περισσότερες δουλειές με το λιγότερο χάσιμο χρόνου. Στην αρχή χρησιμοποιήθηκε ο Ηλεκτρονικός Υπολογιστής αλλά αυτό δεν ήταν αρκετό. Έτσι λοιπόν ξεκίνησε να αναπτύσσεται ραγδαία ο κυβερνοχώρος. Στις μέρες μας τα δεδομένα έχουν αλλάξει. Χιλιάδες εργασίες πραγματοποιούνται μπροστά από έναν υπολογιστή καθώς και με την βοήθεια του Διαδικτύου, και έτσι γίνονται ευκολότερες.

Τα συστήματα αυτόματων συνδιαλλαγών και πληροφόρησης από απόσταση έχουν αρχίσει να αυξάνονται ταχύτατα και να γίνονται μέρος της καθημερινότητας μας. Η προσπέλαση στα συστήματα αυτά γίνεται συνήθως είτε μέσω τηλεφώνου είτε μέσω διαδικτύου. Σκοπός της πτυχιακής μας εργασίας είναι να αναλύσουμε τις απαιτήσεις για τη δημιουργία ενός ολοκληρωμένου συστήματος διαδικτυακής πύλης για το ΤΕΙ Κρήτης και να υλοποιήσουμε ένα μέρος από αυτό.

Η διαδικτυακή πύλη είναι ένα σύστημα με πολλαπλή λειτουργικότητα και ιεραρχική δομή που ενσωματώνει ένα σύνολο εφαρμογών. Η πρόσβαση σε αυτήν γίνεται μέσω Διαδικτύου. Η πύλη του ΤΕΙ Κρήτης έχει ως κύριο στόχο τη διαχείριση των μαθημάτων, την ενημέρωση των φοιτητών και άλλων πληροφοριών που αφορούν το ΤΕΙ Κρήτης. Στα πλαίσια της πτυχιακής μας εργασίας συνδυάζουμε τεχνολογίες αιχμής από τους τομείς των δικτύων υπολογιστών, την πληροφορική και τις τηλεπικοινωνίες για τη δημιουργία ενός συστήματος πληροφόρησης από απόσταση για το ΤΕΙ Κρήτης. Οι τεχνολογίες αυτές εφαρμόζονται κατάλληλα για τη δημιουργία ενός ενιαίου συστήματος παροχής πληροφοριών μέσω του Διαδικτύου που είναι εντελώς απαραίτητο για την αναβάθμιση και βελτιστοποίηση της ποιότητας σπουδών στο ΤΕΙ.

Το Διαδίκτυο αποτελεί ένα διαρκώς ανερχόμενο μέσο επικοινωνίας με τα ακόλουθα πλεονεκτήματα:

- Αναζήτηση συγκεκριμένης πληροφορίας σε ένα ευρύ σύνολο πηγών.
- Δυνατότητα ταυτόχρονης επικοινωνίας πολλών ατόμων μαζί.
- Πολύ μεγάλη ταχύτητα ανταλλαγής εγγράφων, ιδεών, προγραμμάτων που οδηγεί σε διευκόλυνση πάσης φύσεως συναλλαγών.
- Ικανοποιητική διείσδυση σε οργανισμούς, εταιρίες και οικίες.

1.1 Βασική Ιδέα για την δημιουργία του προγράμματος

Κατά την διάρκεια της φοίτησης στο Τεχνολογικό Ίδρυμα Κρήτης, πρέπει κάθε αρχή ακαδημαϊκού εξαμήνου οι σπουδαστές να προβούν στην ανανέωση της εγγραφής τους. Αυτό πολλές φορές ήταν αρκετά κουραστικό

και χρονοβόρο, αφού θα έπρεπε να παρουσιαστούν στην γραμματεία της σχολής μια συγκεκριμένη ημέρα και να καταθέσουν τα απαραίτητα δικαιολογητικά δηλαδή την δήλωση τους. Αυτό όμως πολλές φορές δεν είναι τόσο εφικτό. Δε πρέπει να ξεχνάμε πως το συγκεκριμένο εκπαιδευτικό ίδρυμα εδρεύει στην νησιωτική Ελλάδα, με αποτέλεσμα πολλές φορές να μην είναι εύκολη η πρόσβαση μας σε αυτό λόγω διαφόρων αιτιών (π.χ. καιρός, εργασία). Εάν ένας σπουδαστής δεν παρουσιαστεί για την ανανέωση της εγγραφής την συγκεκριμένη ημέρα όπου ορίζει η γραμματεία τότε πρέπει να παρουσιαστεί μπροστά στο συμβούλιο των καθηγητών για να πάρει την έγκριση ώστε να ανανεώσει την εγγραφή του. Υπάρχει βέβαια και η περίπτωση της εξουσιοδότησης όπου πρέπει να εξουσιοδοτήσεις κάποιο σπουδαστή για να προβεί αυτός στην γραμματεία και να ανανεώσει την εγγραφή. Έτσι λοιπόν ο εκάστοτε σπουδαστής ο οποίος κατοικεί κάπου εκτός Κρήτης θα πρέπει να αφιερώσει κάποιες ημέρες ώστε να ανανεώσει την εγγραφή του. Για τους σπουδαστές οι οποίοι κατοικούν εντός Κρήτης τα πράγματα είναι κάπως ευκολότερα αφού για να προβούν σε αυτή την διαδικασία θα πρέπει να αφιερώσουν μόνο λίγες ώρες. Βάσει αυτών των γεγονότων η ιδέα μίας διαδικτυακής εφαρμογής η οποία θα δίνει την δυνατότητα σε κάθε σπουδαστή να ανανεώνει την εγγραφή από οποιαδήποτε σημείο της χώρας ήταν αναμενόμενη.

Σκοπός λοιπόν για την εκπόνηση αυτής της πτυχιακής εργασίας είναι η διευκόλυνση των σπουδαστών και η άμεση εξυπηρέτησή τους. Βέβαια θα πρέπει να αναφερθούμε και στην γραμματεία της σχολής, αφού όπως πιστεύεται με την βοήθεια αυτής της εφαρμογής θα μειωθεί κατά πολύ η εργασία της και θα έχει περισσότερο χρόνο έτσι ώστε να παρέχει στο ΤΕΙ μια πλήρη γραμματειακή υποστήριξη χωρίς να χάνουν οι εργαζόμενοι χρόνο για τυπικές διαδικασίες όπως είναι η ανανέωση των εγγραφών. Εννοείται πως σε καμία περίπτωση η εφαρμογή αυτή δεν αποσκοπεί να καταργήσει το ανθρώπινο δυναμικό του ΤΕΙ αλλά μόνο να το βοηθήσει, καθώς πιστεύεται ότι θα γίνει ένα χρήσιμο εργαλείο στα χέρια του διοικητικού προσωπικού του ΤΕΙ Κρήτης.

1.2 Απαιτήσεις - Προδιαγραφές του συστήματος

Οι δυνατότητες αυτού του Συστήματος είναι πολύ μεγάλες φυσικά μέσα από αυτήν την εργασία θα δούμε μερικές μόνο εφαρμογές οι οποίες είναι και οι βασικότερες. Αυτές είναι η αυτόματη εγγραφή μέσω Διαδικτύου και η έκδοση αναλυτικής βαθμολογίας ανά πάσα στιγμή.

Κάθε σπουδαστής θα έχει την δυνατότητα να γίνει χρήστης αυτής της εφαρμογής. Το μόνο όπου θα πρέπει να κάνει είναι μία αίτηση στην γραμματεία από όπου και θα παραλαμβάνει και το όνομα χρήστη μαζί με τον κωδικό επιβεβαίωσής του. Εννοείται ότι τα στοιχεία αυτά ο σπουδαστής θα μπορεί να τα μεταποιεί. Η εφαρμογή αυτή θα είναι διαθέσιμη όλο τον χρόνο και έτσι ο σπουδαστής θα μπορεί να λαμβάνει τις πληροφορίες όπου θέλει, χωρίς να χρειάζεται να μπει στην διαδικασία να πάει στην γραμματεία να παραλάβει κάποιο έγγραφο.

Όλα τα στοιχεία που θα έχει στην διάθεσή του ο εκάστοτε σπουδαστής θα προέρχονται από την γραμματεία και συγκεκριμένα από τον διαχειριστή του προγράμματος χωρίς φυσικά να μπορεί να τα μεταποιήσει. Ενώ κάθε χρήστης θα έχει ένα μοναδικό όνομα επιβεβαίωσης έτσι ώστε να μην υπάρχει σύγχυση, επίσης, όπως θα δούμε και παρακάτω, για καλύτερη ασφάλεια σε κάθε ενέργεια που θα προβαίνει ο χρήστης θα υπάρχει και επιβεβαίωση των στοιχείων του.

Κύριο μέλημά μας τώρα είναι η σχεδίαση της εφαρμογής αυτής, παρέχοντας περισσότερες δυνατότητες τόσο στον χρήστη όσο και στον διαχειριστή. Μία επέκταση της διαχείρισης του προγράμματος μέσω του Διαδικτύου σίγουρα θα έλυνε τα χέρια του διαχειριστή αλλά του εκάστοτε χρήστη. Ολόκληρη η εφαρμογή θα ήταν ακέραια και όλοι θα εργαζόταν με την βοήθεια του κυβερνοχώρου από τον χώρο τους και ότι ώρα θα ήθελαν, χωρίς να έχουν τα γνωστά προβλήματα.

1.3 Μια γρήγορη ματιά στην παρούσα εργασία

Το κύριο τμήμα αυτής της πτυχιακής εργασίας δομείται σε πέντε κεφάλαια. Εκτός από το παρόν κεφάλαιο, που αποτελεί μια σύντομη εισαγωγή, στο δεύτερο κεφάλαιο παραθέτονται συστηματοποιημένα οδηγίες για τα προγράμματα (περιβάλλοντα) που χρησιμοποιήσαμε κατά την ανάπτυξη της εφαρμογής. Τα περιεχόμενα αυτού του κεφαλαίου έχουν αντληθεί από διάφορα βιβλία ή εγχειρίδια (manuals) που αναφέρονται στις συγκεκριμένες εφαρμογές. Επίσης, μελετάμε και αναλύουμε τις παρακάτω έννοιες: δυναμική σελίδα, πρωτόκολλου TCP/IP και ODBC. Στο τρίτο κεφάλαιο παρουσιάζουμε τα διάφορα στάδια της σχεδίασης-ανάπτυξης της εφαρμογής (σχεδίασης βάσης δεδομένων, ανάπτυξη ιστοσελίδων). Το τέταρτο κεφάλαιο αποτελεί μία παρουσίαση του τρόπου χρήσης του προγράμματος μας τόσο από την οπτική γωνία του χρήστη όσο και από αυτήν του διαχειριστή του προγράμματος. Στο πέμπτο και τελευταίο κεφάλαιο συνοψίζουμε τα συμπεράσματα στα οποία καταλήξαμε κατά την ενασχόλησή μας με το παρόν θέμα. Στη συνέχεια, παραθέτουμε την σχετική βιβλιογραφία που συμβουλευτήκαμε κατά την εκπόνηση της παρούσας εργασίας, η οποία αποτελείται τόσο από τίτλους βιβλίων όσο και από ιστοχώρους. Τέλος, υπάρχει παράρτημα το οποίο παρουσιάζει αυτούσια τμήματα κώδικα από επιλεγμένα σημεία της εφαρμογής μας.

2. Εξοικείωση με τα αναπτυξιακά περιβάλλοντα

2.1 Γενικά

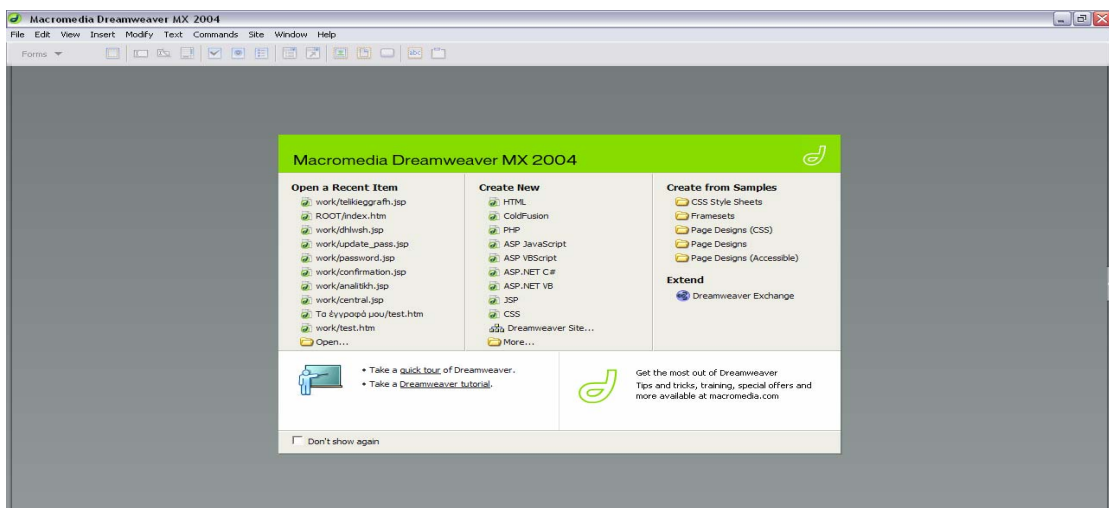
Για την σχεδίαση και ολοκλήρωση αυτής της πτυχιακής εργασίας χρησιμοποιήσαμε κάποιες εφαρμογές. Οι εφαρμογές σχετιζόταν είτε με το κομμάτι της δημιουργίας σελίδων jsp είτε με την διαχείριση των δεδομένων είτε με την εμφάνιση των δεδομένων. Ονομαστικά τα προγράμματα που χρησιμοποιήθηκαν είναι:

- α → Macromedia Dreamweaver MX 2004
- β → Jakarta Tomcat
- γ → Microsoft ODBC

2.2 Macromedia Dreamweaver MX 2004

Το Macromedia Dreamweaver MX 2004 είναι επαγγελματική εφαρμογή για την σύνταξη και τον σχεδιασμό HTML σελίδων, την κωδικοποίηση, και την ανάπτυξη των διαφόρων ιστοσελίδων καθώς και πολλών εφαρμογών που αναφέρονται στο Διαδίκτυο. Εάν ο χρήστης επιθυμεί να δημιουργήσει ένα χειρόγραφο κομμάτι (χειρονακτικά) ή προτιμά να εργαστεί σε ένα γραφικό περιβάλλον η έκδοση του Dreamweaver του παρέχει όλα τα χρήσιμα σε αυτόν εργαλεία για την δημιουργία των σελίδων του. Τα οπτικά χαρακτηριστικά γνώρισμα έκδοσης σε Dreamweaver μας αφήνουν να δημιουργήσουμε γρήγορα τις σελίδες χωρίς την πληκτρολόγηση καμιάς γραμμής κώδικα. Μπορείτε να δείτε άμεσα όλα τα μειονεκτήματα ή τα προτερήματα της σελίδας σας και να τα επεξεργαστείτε προτού να δημοσιεύσετε την ιστοσελίδα σας. Μπορείτε να βελτιώσετε τη ροή της δουλειάς με την ανάπτυξη και τη δημιουργία διαφόρων κινουμένων εικόνων (flash) μέσω του Macromedia Flash MX 2004 ή κάποιας εφαρμογής γραφικής απεικόνισης διαφόρων εικόνων, εισάγοντας τα έπειτα άμεσα στο Dreamweaver, ή με την προσθήκη των αντικειμένων Macromedia Flash MX 2004. Το Dreamweaver παρέχει επίσης ένα πλήρες περιβάλλον για εγγραφή του κώδικα και περιλαμβάνει τα εργαλεία για την σύνταξη του κώδικά μας. Το Dreamweaver έχει την δυνατότητα να επεξεργαστεί σελίδες της μορφής HTML, JavaScript, γλώσσα σήμανσης ColdFusion (CFML), ενεργές σελίδες κεντρικών υπολογιστών της Microsoft (ASP), και σελίδες JavaServer (JSP). Η μετ' επιστροφής τεχνολογία HTML Macromedia εισάγει το χειρόγραφο έγγραφο HTML χωρίς να μορφοποιεί των κώδικα και στη συνέχεια να τον διαφοροποιήσουμε για να πάρουμε τα δικά μας επιθυμητά αποτελέσματα. Το Dreamweaver επιτρέπει επίσης την δημιουργία δυναμικών σελίδων αντλώντας δεδομένα από διάφορες βάσεις δεδομένων όπου υποστηρίζονται από τις ιστοσελίδες οι οποίες χρησιμοποιούν τις τεχνολογίες κεντρικών υπολογιστών όπως CFML, ASP.NET, ASP, JSP. Το Dreamweaver MX το 2004 έχει μια νέα σχεδίαση και εντελώς πρωτοποριακή σε σχέση με τα υπάρχοντα προγράμματα

σχεδίασης ιστοσελίδων. Αυτά τα νέα προτερήματα θα τα παρατηρήσουμε στη δημιουργία είτε εάν εργαζόμαστε στο γραφικό περιβάλλον είτε εργαζόμαστε στο περιβάλλον εγγραφής κώδικα. Η βελτιωμένη σχεδίαση και ανάπτυξη της ιστοσελίδας δια μέσου του Dreamweaver είναι πιο προσιτή για να μας βοηθήσει να βελτιώσουμε την παραγωγικότητα και της ποιότητα εργασίας μας. Μία από τις πιο σημαντικές δυνατότητες που μας παρέχει το Dreamweaver αναφέρεται στις γνωστές σε όλους μας εφαρμογές Microsoft Word και Microsoft Excel όπου μας δίνει την δυνατότητα να αντιγράψουμε και να εισάγουμε εντολές από το Microsoft Word ή από ένα έγγραφο Excel άμεσα στο Dreamweaver. Όταν κάνουμε επικόλληση ενός εγγράφου του Word ή Excel, το Dreamweaver δημιουργεί τη μορφοποίηση και παράγει μια ποιοτική και ασφαλή HTML σελίδα. Μετά την έκδοση αυτής της σελίδας γίνεται μία ανατροφοδότηση από το πηγαίο κομμάτι του κώδικα ενώ μας επιτρέπει να δούμε άμεσα τα αποτελέσματά μας. Η αρχική σελίδα μας επιτρέπει να έχουμε πρόσβαση στα πρόσφατα χρησιμοποιημένα αρχεία και τους πόρους Dreamweaver (σχήμα 2.2.1) ή να δημιουργήσουμε νέα αρχεία. Η σελίδα έναρξης εμφανίζεται είτε όταν αρχίζει το Dreamweaver ή όταν θέλουμε να επεξεργαστούμε ένα υπάρχον αρχείο. Σώζοντας το επιλεγμένο αρχείο σε ένα υπολογιστή έχουμε την δυνατότητα να το ανοίξουμε και να το επεξεργαστούμε εκ νέου. Επιπλέον η εφαρμογή του Dreamweaver μας δίνει την δυνατότητα να γράψουμε κώδικα και να περάσουμε όλες τις δυνατές κωδικοποιήσεις που υποστηρίζονται στον χώρο του Διαδικτύου. Η ιδιότητα της κωδικοποίησης δεν αναφέρεται μόνο σε σελίδες τις οποίες δημιουργούμε εμείς αλλά και σε υπάρχουσες που θέλουμε να επεξεργαστούμε, και εμείς με την σειρά μας θα την σώσουμε σε οποιαδήποτε μορφή θέλουμε.



Σχήμα 2.2.1 Κεντρική εικόνα του Dreamweaver MX 2004

Η καινούργια εφαρμογή του Dreamweaver (Σχήμα 2.2.2) μας παρέχει ένα περιβάλλον γραφικού σχεδίου με ενισχυμένα χαρακτηριστικά γνωρίσματα, όπου κάνουν ευκολότερη την χρήση όλων των διαθέσιμων εργαλείων ενώ τα αποτελέσματα έχουμε την δυνατότητα να τα δούμε άμεσα μέσα από ένα απλό προσομοιωτή λειτουργίας HTML σελίδων. Η δυναμική επικύρωση που μας παρέχει το Dreamweaver ελέγχει αυτόματα το τρέχον έγγραφο για τα ζητήματα συμβατότητας και μας το ανοίγει ή όχι ανάλογα με το εάν έχουμε

δυνατότητα να το επεξεργαστούμε. Ελέγχει και μας ενημερώνει αυτόματα εάν αντιληφθεί κάποιο συντακτικό λάθος στον κώδικά μας και μας προτείνει διάφορες μεθόδους έτσι ώστε να εντοπίσουμε το λάθος μας και να προχωρήσουμε στην επιμέρους επεξεργασία του κώδικά μας. Εάν επιλέξουμε οποιοδήποτε κομμάτι κώδικα έχουμε τη δυνατότητα να διευκρινίσουμε τις ιδιότητες της συγκεκριμένης εντολής ενώ η εμφάνιση συγκεκριμένων συμβόλων μας αφήνει να γνωρίζουμε τις ιδιότητες συγκεκριμένων εντολών και να μας υπενθυμίζει τις συγκεκριμένες ιδιότητες εάν έχουν αγνοηθεί από εμάς, ενώ επιτόπου έχουμε την έκδοση της σελίδας μας, επιτρέποντας μας να δούμε τις όποιες ατέλειες που δεν γίνονται αντιληπτές κατά την διάρκεια της σύνταξης του κώδικά μας. Η βελτιωμένη έκδοση του Dreamweaver μας επιτρέπει να σχεδιάσουμε περίπλοκα σχεδιαγράμματα, με τα εργαλεία που μας παρέχει, ή να επεξεργαστούμε οποιαδήποτε μορφή εικόνας που επιθυμούμε χωρίς να μας δημιουργήσει κάποιο πρόβλημα αφού έχουμε την δυνατότητα να εισάγουμε οποιαδήποτε εικόνα εμείς θέλουμε (.jpeg, .gif, .tif κ.α.). Η βελτιωμένη έκδοση μας παρέχει επιπλέον και πολλές επιλογές σε ένα τρέχον έγγραφο, μπορούμε άμεσα να εφαρμόσουμε διάφορες εφαρμογές σε ένα κώδικα ο οποίος έχει παραχθεί από άλλο χρήστη να τον σώσουμε και να τον δημοσιεύσουμε με κάποιο άλλο όνομα.

```

1 <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
2 "http://www.w3.org/TR/html4/loose.dtd">
3 <html>
4 <head>
5 <title>Κεντρική Σελίδα</title>
6 <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-7">
7 </head>
8 <body>
9
10 <div align="center">
11 <p><strong>ΤΕΙ ΚΡΗΤΗΣ</strong></p>
12 <p><strong>ΠΑΡΑΡΤΗΜΑ ΧΑΝΙΩΝ</strong></p>
13 <p><strong>ΕΦΑΡΜΟΤΗ ΔΗΜΟΣΙΑΣ ΔΙΑΧΕΙΡΙΣΗΣ</strong></p>
14
15 <p>&nbsp;</p>
16 <div align="center">
17
18
19 <table width="370" border="1">
20 <tr>
21 <td width="137" height="33">Επιλέξτε Τμήμα :</td>
22 <td width="217"><span class="tabletext">
23 <SELECT name="tblArea" class="searchformarea" >
24 <OPTION value="1"> Ηλεκτρονική </OPTION>
25 <OPTION value="2"> Φυσικών Πόρων </OPTION>
26 </SELECT>
27 </span></td>
28 </tr>
29 <tr>
30 <td colspan="2"><form name="form1" method="post" action="work/central.jsp" encoding="iso-8859-7">
31 <input type="button" value="Αναζήτηση" /></td>
32 </tr>
33 </table>
34 </div>
35 </body>
36 </html>

```

Σχήμα 2.2.2 Εικόνα από το κειμενογράφο του Dreamweaver

Τελειώνοντας αυτή την μικρή παρουσίαση του Dreamweaver θα πρέπει να αναφερθούμε και στη δυνατότητα που μας παρέχει στη δημιουργία δυναμικών σελίδων. Με τον όρο της δυναμικής σελίδας αναφερόμαστε στις σελίδες εκείνες όπου μας παρουσιάζουν δεδομένα αντλώντας και αυτές με την σειρά τους δεδομένα από άλλες εφαρμογές όπως π.χ. κάποια μορφή βάσης δεδομένων ή τα περιεχόμενα διαφόρων JavaBeans. Για να προσθέσουμε κάποια δυναμικά δεδομένα στη σελίδα μας δεν κάνουμε τίποτα άλλο από το να «σύρουμε» απλά μέσα στο περιβάλλον μας και το Dreamweaver θα δημιουργήσει μόνο του τον πηγαίο κώδικα που απαιτείται. Μπορούμε επίσης να επεξεργαστούμε πολλά δεδομένα τη φορά

και να μας εμφανίζονται σε μία ή σε πολλές σελίδες τη φορά, να δημιουργήσουμε συνδέσμους με προηγούμενες ή επόμενες σελίδες και να κινηθούμε όπως εμείς θέλουμε, να δημιουργήσουμε κάποιας μορφής μετρητή όπου θα αναφέρεται στην επισκεψιμότητα της σελίδας μας ή ακόμη και κάποια εφαρμογή όπου κάποιοι χρήστες θα μπορούν να αντλούν, να εισάγουν και γενικότερα να επεξεργάζονται διαφόρων μορφής αρχεία. Βάση της τελευταίας δυνατότητας στηρίζεται και η δικιά μας εφαρμογή όπου θα αναλυθεί παρακάτω. Η τελευταία δυνατότητα όπου μας παρέχει το Dreamweaver από πολλούς θεωρείτε η πιο δύσκολη και η πιο επώδυνη για το λόγο ότι ο κώδικας πρέπει να γραφεί εξολοκλήρου από το δημιουργό της εφαρμογής έχοντας μια αρκετά καλή γνώση της γλώσσας προγραμματισμού όπου επιθυμεί να χρησιμοποιήσει.

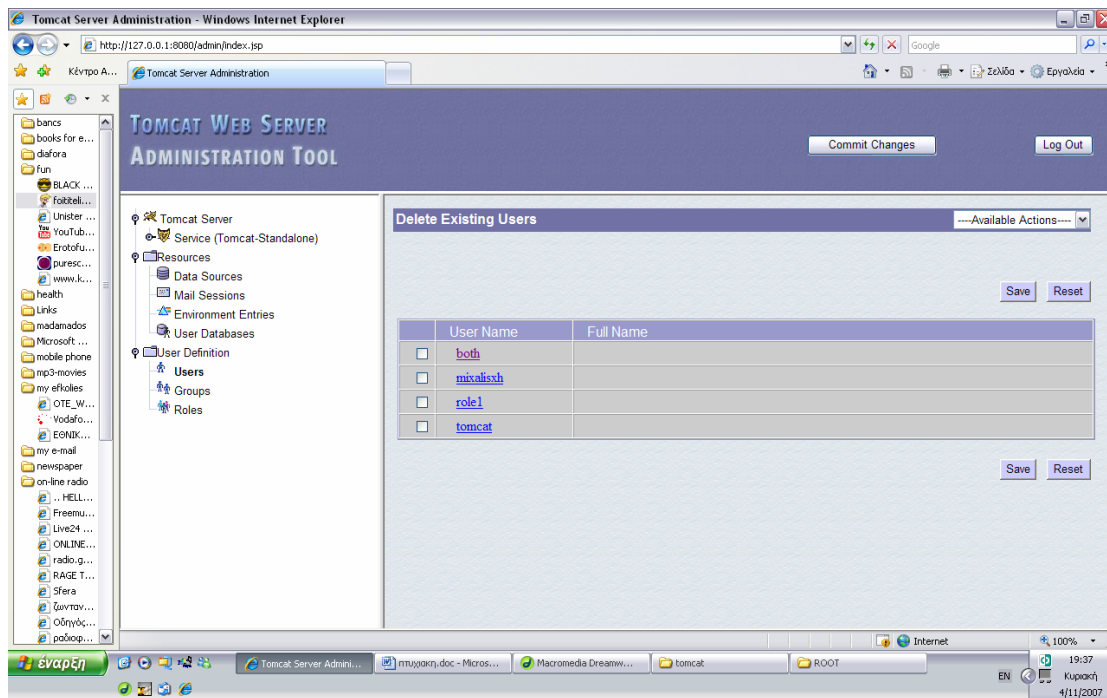
2.3 Εισαγωγή στον Jakarta Tomcat

Εάν οποιοσδήποτε θέλει να επεξεργαστεί εντολές servlets της java ή σελίδες JavaServer (JSPs), το πρόγραμμα το οποίο θα πρέπει να χρησιμοποιήσει είναι το Jakarta Tomcat.

Το jakarta Tomcat είναι μια ελεύθερη πλατφόρμα, και πλήρες εμπορευματοποιημένη του οποίου οι χρήστες έχουν την δυνατότητα να αναπτύξουν και να εξετάσουν τον κώδικά τους. Το jakarta Tomcat είναι μια εφαρμογή η οποία αναφέρεται σε servlet και JSPs, κάτι που σημαίνει ότι είναι εκμεταλλεύσιμο, συνεργάσιμο και υποστηρίζει 100% όλες τις εκδόσεις servlet και JSPs. Το jakarta Tomcat μια γνωστή πλατφόρμα για την δημοσίευση σελίδων στο διαδίκτυο. Πολλές είναι οι επιχειρήσεις τόσο στην ελληνική αγορά όσο και στην παγκόσμια που το χρησιμοποιούν στον κεντρικό τους υπολογιστή (server) λόγω της σταθερότητας του στις εφαρμογές που υποστηρίζει και λόγω του χαμηλού κόστους για την απόκτηση της άδειας εκμετάλλευσής του. Παρά τη δημοτικότητα του, ο jakarta Tomcat έχει και κάποια μειονεκτήματα και συγκεκριμένα αναφερόμαστε στην μη σωστή τεκμηρίωση και στην εμφάνιση κωδικών οι οποίοι δεν εξηγούν πλήρως κάποια πιθανά συντακτικά λάθη, τα οποία πιθανότατα βρίσκονται στον πηγαίο κωδικό. Μία στοιχειώδη επεξήγηση παρέχεται από το ίδιο το πρόγραμμα και μπορούμε να βρούμε στον παρακάτω σύνδεσμο: <http://jakarta.apache.org/tomcat/tomcat-4.0-doc>, ή σε ένα on-line βιβλίο του jakarta Tomcat το οποίο υπάρχει στο: <http://tomcatbook.sourceforge.net>. Παρόλα αυτά ακόμα και με αυτές τις δύο βασικές πηγές δεν θα μπορούσαμε να βρούμε όλες τις απαραίτητες πληροφορίες που θα θέλαμε. Ως εκ τούτου η δημιουργία αυτών των δύο βιβλίων-βοηθημάτων έχουν επιφέρει αρκετά καλά αποτελέσματα τόσο στην σωστότερη κατανόηση του jakarta Tomcat όσο και στην ανάπτυξη εφαρμογών Java τόσο σε μη-επαγγελματικό επίπεδο όσο και σε επαγγελματικό. Πολλοί διαχειριστές και συντάκτες επαγγελματικών εφαρμογών εμπιστεύονται το jakarta Tomcat σαν μια εφαρμογή η οποία θα ανταποκριθεί πλήρως στις επαγγελματικές απαιτήσεις όπου έχει μια εταιρία. Η περαιτέρω ανάλυση που θα ακολουθήσει θα μας βοηθήσει να κατανοήσουμε πλήρως τους όρους του servlet, και να περιηγηθούμε στις

άγνωστες πτυχές του jakarta Tomcat. Ένα από τα πιο διαδεδομένα βιβλία το οποίο αναφέρεται στο jakarta Tomcat είναι το jakarta Apache, όπου θεωρεί το jakarta Tomcat σαν ένα υπο-πρόγραμμα του κεντρικού υπολογιστή. Γενικότερα τα προγράμματα του jakarta θεωρούνται μία ομπρέλα κάτω από την οποία θα βρούμε την υποστήριξη πολλών υποπρογραμμάτων της Java. Κατά την διάρκεια της μελέτης που θα κάνουμε, θα δούμε να υπάρχει μια εντυπωσιακή πληθώρα από υποπρογράμματα. Μπορούμε να τα διαιρέσουμε σε τρεις διαφορετικές κατηγορίες: "Libraries, Tools, and APIs", "Frameworks and Engines", και "Server Applications ". Εμείς με την σειρά μας θα δώσουμε έμφαση στο Tomcat, όπου είναι εκείνη η πλατφόρμα που χρησιμοποιήσαμε για αυτή την πτυχιακή εργασία.

Το πρόγραμμα του jakarta Tomcat έχει την προέλευσή του από τις πρώτες ημέρες της τεχνολογίας servlet της java. Το 1999, η Sun έδωσε τα πρώτα δείγματα κώδικα με servlet και ASF, μετέπειτα τα δύο προγράμματα συγχωνεύθηκαν και δημιούργησαν ένα κεντρικό πρόγραμμα όπου έτρεχε σε ένα κεντρικό υπολογιστή. Αυτό το πρόγραμμα είναι το Tomcat. Σήμερα, το Tomcat χρησιμεύει ως η επίσημη εφαρμογή αναφοράς της Java Sun, τώρα πρώτο μέλημα της Java Sun είναι η ανάπτυξη της εφαρμογής jakarta Tomcat η οποία είναι πλήρως συμβατή με οποιαδήποτε έκδοση Servlet και έχει όλες τις προδιαγραφές για τη δημιουργία JSP σελίδων καθώς και την δημοσίευσή τους, μέσα από την επίσημη ιστοσελίδα της Java Sun. Το jakarta Tomcat, επίσης μπορεί να υποστηρίξει και σελίδες γραμμένες σε HTML κώδικα χωρίς να δημιουργεί προβλήματα συμβατότητας μιας και ο κώδικας HTML είναι μία γλώσσα προγραμματισμού η οποία δεν δημιουργήθηκε από την Java Sun. Ένα δεύτερο πλεονέκτημα όπου υπάρχει στην ανανεωμένη έκδοση του jakarta Tomcat είναι ότι η τεχνικοί του επιδιώκουν την εξάλειψη των μειονεκτημάτων, το οποίο σχετίζεται με την αποκωδικοποίηση των διαφόρων συντακτικών λαθών, που είχαμε αναφέρει παραπάνω. Η όλη προσπάθεια γίνεται για το λόγο ότι η εφαρμογή jakarta Tomcat θα ήταν καλύτερο να γίνει πιο φιλική και πιο προσιτή στους χρήστες του είτε αναφερόμαστε σε επαγγελματίες, είτε αναφερόμαστε σε μεμονωμένα άτομα, έτσι ώστε να μην χρειάζεται οι χρήστες να είναι σε μια συνεχής αναζήτηση κωδικών. Η αναφερόμενη πλατφόρμα τώρα είναι ελεύθερη και άμεσα διαθέσιμη στο διαδίκτυο. Η εγκατάσταση και η επεξεργασία της θα πρέπει να γίνεται στον κεντρικό υπολογιστή στον οποίο πρόσβαση θα πρέπει να έχουν μόνο συγκεκριμένοι χρήστες, η οποίοι θα πρέπει να γνωρίζουν τα ιδιαίτερα χαρακτηριστικά του jakarta Tomcat. Σύντομα περιμένουμε την ακόμη νεότερη έκδοση της εφαρμογής στην οποία πιθανότητα θα έχουν εξαλειφθεί όλα τα τρωτά σημεία των προηγούμενων εκδόσεων. Η πρώτη έκδοση Tomcat ήταν η 3.x σειρά, και χρησίμευσε ως η εφαρμογή αναφοράς Servlet 2,2 και είχε προδιαγραφές JSP 1,1. Το 2001, εμφανίζεται το Tomcat 4,0 (η με κωδικό Catalina) απελευθερώνοντας αρκετά από τα προβλήματα που είχε το προηγούμενο. Στην ουσία ήταν ένας πλήρης επανασχεδιασμός του Tomcat και η αρχιτεκτονική βασίστηκε σε μια νέα βάση πηγαίου κώδικα. Η 4.x σειρά Tomcat, που είναι τρέχουσα σε αυτή την διπλωματική εργασία μπορεί να υποστηρίξει Servlet 2,3 και έχει προδιαγραφές δημοσίευσης JSP 1,2 (σχήμα 2.3.1).



Σχήμα 2.3.1 κεντρική εικόνα του tomcat όπου χρησιμοποιούμε

Κατά την διάρκεια της σύνταξης της εφαρμογής μας χρησιμοποιήσαμε την πιο πρόσφατη έκδοση του Tomcat η οποία είναι προάγγελος της νέας έκδοσής του και θα υποστηρίζει Servlet 2,4 και προδιαγραφές JSP 2,0.

2.4 Μοντέλο Πελάτη – Εξυπηρετητή

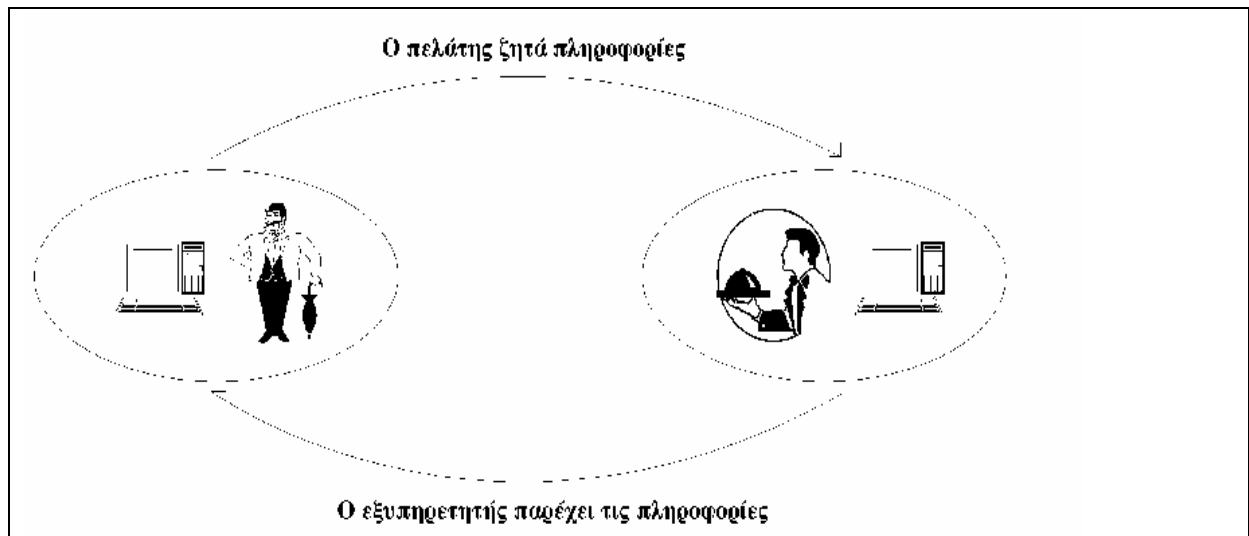
Προκειμένου να χρησιμοποιήσουμε μια υπηρεσία του Internet θα πρέπει:

- να έχουμε εγκατεστημένο στον υπολογιστή μας και να εκτελέσουμε το κατάλληλο πρόγραμμα για αυτή την υπηρεσία. Το πρόγραμμα αυτό ονομάζεται πελάτης (client). Μέσω του πελάτη, ζητάμε την παροχή της συγκεκριμένης υπηρεσίας.
- να έχουμε πρόσβαση (μέσω Internet) σε μηχανή που υποστηρίζει την αιτούμενη υπηρεσία. Σε αυτή τη μηχανή πρέπει να εκτελείται ένα πρόγραμμα που παρέχει τη συγκεκριμένη υπηρεσία, ο εξυπηρετητής (server).

Η παροχή των περισσότερων υπηρεσιών στο Internet βασίζεται στο μοντέλο πελάτη-εξυπηρετητή (client-server) (σχήμα 2.4.1) που λειτουργεί ως εξής:

Ο πελάτης ζητά από τον εξυπηρετητή πληροφορίες και ο τελευταίος εξυπηρετεί το αίτημα παρέχοντας του τις πληροφορίες αυτές. Αφού τελειώσει η διαδικασία, ο εξυπηρετητής περιμένει έως ότου κάποιος πελάτης υποβάλλει πάλι κάποια αίτηση για εξυπηρέτηση.

Κάθε υπηρεσία στο Internet έχει το δικό της ξεχωριστό πρωτόκολλο, δηλαδή το δικό της σύνολο από συμβάσεις που καθορίζουν το πώς γίνεται η "συνομιλία" του αντίστοιχου ζεύγους πελάτη-εξυπηρετητή. Έτσι, άλλο πρωτόκολλο χρησιμοποιεί η υπηρεσία WWW, άλλο η υπηρεσία FTP, άλλο η υπηρεσία E-mail, κ.ο.κ. Ένα σημείο που είναι σημαντικό να κατανοήσουμε είναι ότι μέσω του πελάτη δεν μπορούμε να ζητάμε την παροχή της συγκεκριμένης υπηρεσίας από οποιονδήποτε υπολογιστή του Internet. Θα πρέπει ο υπολογιστής στον οποίον απευθύνουμε την αίτησή μας να υποστηρίζει την υπηρεσία αυτή και να εκτελεί τον αντίστοιχο εξυπηρετητή.



Σχήμα 2.4.1 σχηματική παράσταση του μοντέλου Πελάτη – Εξυπηρετητή

Έστω π.χ. ότι θέλουμε να δούμε μια σελίδα του World Wide Web (WWW). Τρέχουμε στον υπολογιστή μας έναν πελάτη για WWW για να συνδεθούμε με έναν απομακρυσμένο υπολογιστή όπου είναι αποθηκευμένη η σελίδα και στον οποίον εκτελείται ένας εξυπηρετητής για WWW. Ο εξυπηρετητής στέλνει στον πελάτη μας τη σελίδα και ο πελάτης την εμφανίζει στην οθόνη του υπολογιστή μας. Σημειώνουμε ότι οι πελάτες για την υπηρεσία WWW ονομάζονται και browsers (αναζητητές). Επίσης, σε έναν υπολογιστή μπορούν να εκτελούνται ταυτόχρονα εξυπηρετητές για περισσότερες από μία υπηρεσίες π.χ. ένας εξυπηρετητής για WWW, ένας εξυπηρετητής για FTP, κι ένας εξυπηρετητής για E-mail. Έτσι, ο ίδιος υπολογιστής μπορεί να παρέχει περισσότερες από μία υπηρεσίες. Υπάρχουν διάφορα προγράμματα - πελάτες για καθεμία από τις υπηρεσίες του Internet για διάφορα λειτουργικά συστήματα. Πολλά από αυτά διατίθενται ελεύθερα μέσω του Internet και μπορούμε να τα μεταφέρουμε στον υπολογιστή μας.

2.4 Σύστημα πηγών δεδομένων (ODBC)

Τα Open Database Connectivity (ODBC) μας επιτρέπουν σε πολλά προγράμματα να επικοινωνήσουν με πολλά προγράμματα και να ανταλλάξουν πληροφορίες με κάποιες εγκαταστημένες βάσεις δεδομένων.

Η επιλογή του ODBC επιτρέπει σε μας να χρησιμοποιήσουμε μια βάση δεδομένων να εισάγουμε η να αντλήσουμε πληροφορίες άμεσα μέσα από αυτή. Κατά την αλληλεπίδραση δημιουργούνται αρχεία με κατάληξη .CG, η οποία είναι μια ψευδο-εφαρμογή η οποία παράγεται στον κεντρικό υπολογιστή όπου είναι εγκαταστημένοι η εφαρμογή του ODBC.

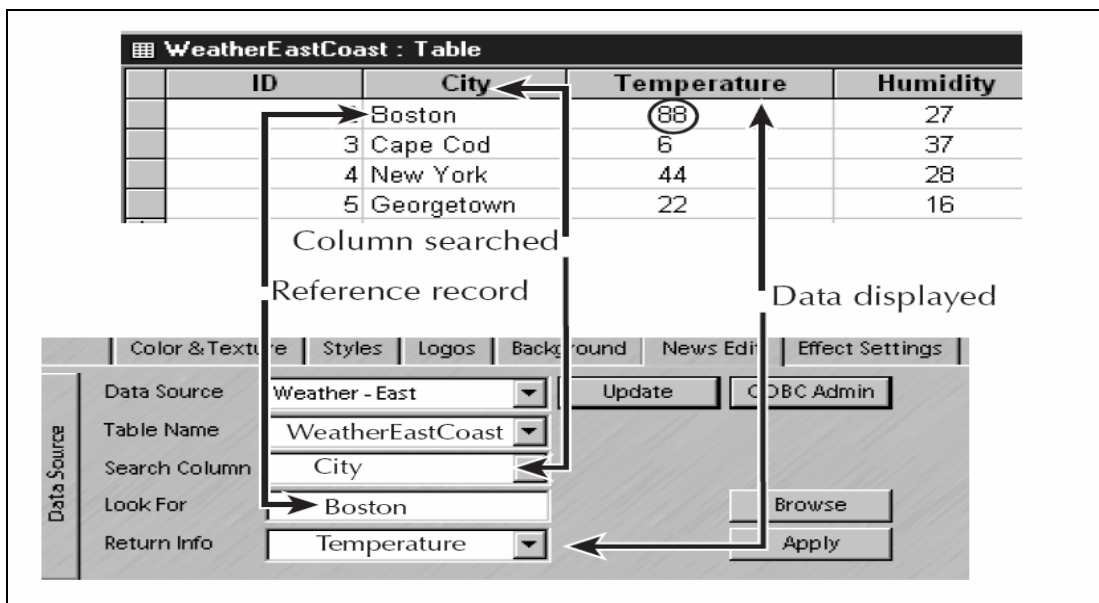
Μπορείτε να χρησιμοποιήσετε την επιλογή της ODBC χρησιμοποιώντας οποιαδήποτε εφαρμογή βάσης δεδομένων (Open Database Connectivity ή Σύστημα διαχείρισης βάσεων δεδομένων) αρκεί να υποστηρίξει το πρότυπο ODBC.

Η βάση δεδομένων μπορεί να είναι απλή ή τόσο σύνθετη. Η επιλογή ODBC πρέπει μόνο να κάνει μια μόνο σύνδεση την κεντρική βάση την οποία έχουμε ορίσει. Μπορείτε να τοποθετήσετε τα στοιχεία σε οποιοδήποτε πίνακα της βάσης μας. Για να εντάξουμε και να δημοσιεύσουμε μια βάση δεδομένων στον κυβερνοχώρο θα πρέπει να ακολουθήσουμε την παρακάτω διαδικασία:

- 1 Δημιουργούμε μια βάση δεδομένων με οποιοδήποτε πρόγραμμα εμείς επιθυμούμε.
- 2 Συνδέουμε το CG σας με τη βάση δεδομένων μας με τη ρύθμιση μιας πορείας.
- 3 Δημιουργούμε το σχεδιάγραμμά της βάσης μας στο CG και επιλέγουμε τα αρχεία που θέλουμε να δημοσιεύσουμε.
- 4 Εισάγουμε τα στοιχεία μας στην βάση δεδομένων μας.
- 5 Ρυθμίζουμε όλες τις παραμέτρους στην ODBC (καταχώριση βάσης, όνομα κ.α.).
- 6 Εκτός από τις ρυθμίσεις όπου πρέπει να εισάγουμε στον κεντρικό υπολογιστή, θα πρέπει να υπάρχουν οι κατάλληλες εφαρμογές στον υπολογιστή όπου θα ζητήσει αυτές τις πληροφορίες.
- 7 Τέλος ενημερώνουμε τη βάση δεδομένων μας σε οποιαδήποτε αλλαγή εμείς κάνουμε στα στοιχεία της βάσης δεδομένων μας, όταν απαιτείται να κάνουμε εμφάνιση πληροφοριών.

Η ανανέωση σε οποιαδήποτε δημοσιευμένα αρχεία γίνεται άμεσα, είτε με κάθε εισαγωγή μας στο σύστημα είτε κάθε 60 δευτερόλεπτα από την στιγμή όπου θα κάνουμε εισαγωγή. Για την διαχείριση της βάση δεδομένων διαμέσου της ODBC πρέπει η βάση μας να περιέχει όλες τις πληροφορίες που η εφαρμογή μας επιθυμεί. Κάθε βάση δεδομένων μπορεί να περιέχει τις διαφορετικές πληροφορίες σε μια ποικιλία από πίνακες. Η ODBC μπορεί να ανακτήσει τις πληροφορίες από οποιαδήποτε πίνακα ή βάση δεδομένων μόλις γίνει μια σύνδεση. Μπορούμε επίσης να έχουμε πρόσβαση σε πολλαπλές βάσεις δεδομένων η οποίες είναι θα είναι εγκατεστημένες τόσο στον υπολογιστή όπου έχει εγκατασταθεί η ODBC είτε σε κάποιον απομακρυσμένο υπολογιστή αρκεί να δηλωθεί σωστά η διαδρομή στην αρχική ρύθμιση της ODBC. Εάν η βάση δεδομένων μας βρίσκεται σε έναν απομακρυσμένο υπολογιστή, ρυθμίσεις πρέπει να γίνουν και σε εκείνον τον υπολογιστή ενώ παράλληλα ο απομακρυσμένος υπολογιστής θα πρέπει να είναι συνέχεια on-line για το λόγο ότι οι πληροφορίες θα πρέπει να είναι άμεσα διαθέσιμες στην εφαρμογή της ODBC και να παρέχει πληροφορίες

όποτε ζητηθούν. Εάν ο απομακρυσμένος υπολογιστής όπου περιέχει την βάση δεδομένων δεν έχει τη δυνατότητα να παρέχει τις πληροφορίες που ζητάει ο εκάστοτε χρήστης, θα λάβει ένα μήνυμα λάθους μετά από μια καθυστέρηση 60-90 δευτερολέπτων. Η ODBC όταν πραγματοποιεί μία αναζήτηση επιλέγει τα ονόματα των στηλών από την βάση δεδομένων, ενώ χρησιμοποιεί την πρώτη σειρά των ετικετών από τον πίνακα της βάσης μας. Για αυτό η ODBC θα πρέπει να ξέρει τα ακριβή ονόματα της βάσης δεδομένων, του πίνακα, καθώς και των στηλών από τις οποίες θα θέλαμε να αντλήσουμε τις πληροφορίες μας. Στο σχήμα 2.5.1 έχουμε ένα χαρακτηριστικό παράδειγμα σε μια σχέση μεταξύ του παραθύρου διαλόγου της ODBC και μιας πηγή δεδομένων.



Σχήμα 2.5.1 Χαρακτηριστική απεικόνιση σύνδεσης μεταξύ της ODBC και της πηγής των δεδομένων μας

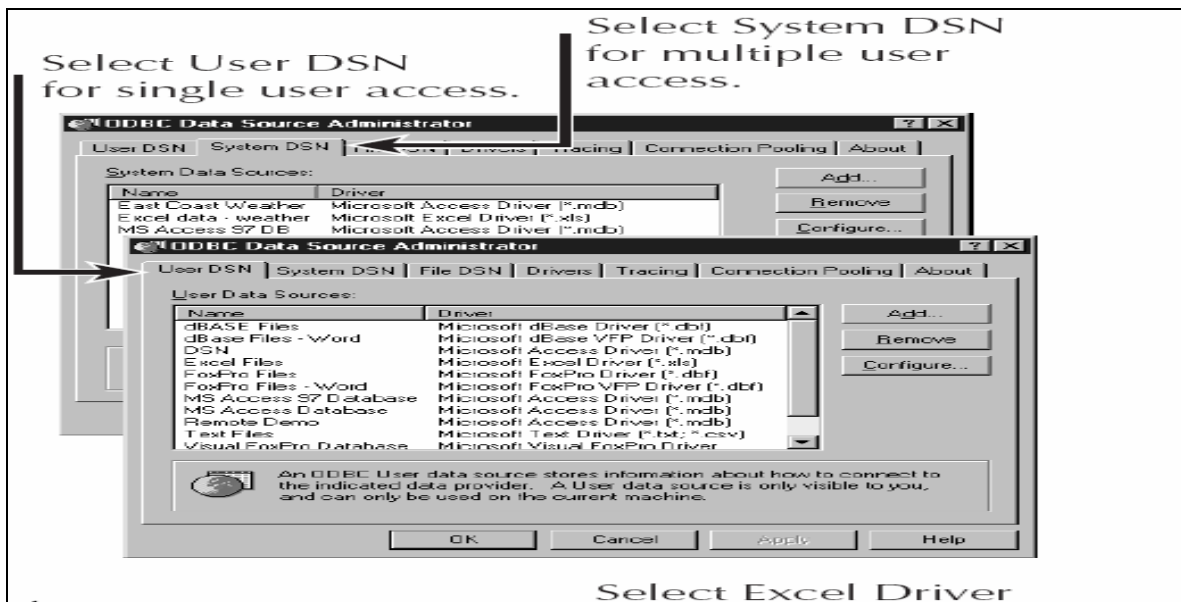
Μόλις θέσετε μια διαδρομή στη βάση δεδομένων μας, η εφαρμογή της ODBC μπορεί να διαβάσει τους τίτλους των στηλών μας. Ενώ εμείς με την σειρά μας επιλέγουμε ποιες από τις προαναφερθείσες στήλες περιέχουν τα στοιχεία που χρειαζόμαστε. Το πεδίο Look For ενεργεί ως αρχείο αναφοράς. Όπως φαίνεται στο παράδειγμά μας (σχήμα 2.5.1), θέλουμε να μελετήσουμε την θερμοκρασία για την πόλη της Βοστώνης και να μας την εμφανίσει σε ένα σχεδιάγραμμα. Πρέπει να προσδιορίσουμε σε ποια στήλη στο αρχείο αναφοράς μας θα εμφανίζεται, και σε ποιο ακριβές στοιχείο αναφερόμαστε, δηλαδή Βοστόνη. Μόλις προσδιορίσουμε στην επιλογή ODBC πού πρέπει να αναζητήσει, επιλέγουμε απλά τον τίτλο των στηλών των στοιχείων που θέλουμε και αυτό επιστρέφει τα δεδομένα που θέλουμε.

Εάν η βάση δεδομένων όπου χρησιμοποιούμε δεν είναι μία οργανωμένη βάση τότε η εφαρμογή της ODBC θα δημιουργήσει ένα αρχείο της μορφής MDAC (τμήματα πρόσβασης στοιχείων της Microsoft). Με αυτήν την εγκατεστημένη εφαρμογή, μπορούμε να δούμε, να εκδώσουμε, ή και να τρέξουμε τα αρχεία της βάσης δεδομένων μας στο διαδίκτυο χωρίς κανένα πρόβλημα. Για να επεξεργαστούμε τα δεδομένα της βάσης δεδομένων μας με την εφαρμογή της ODBC θα πρέπει να έχουμε εγκατεστημένη την

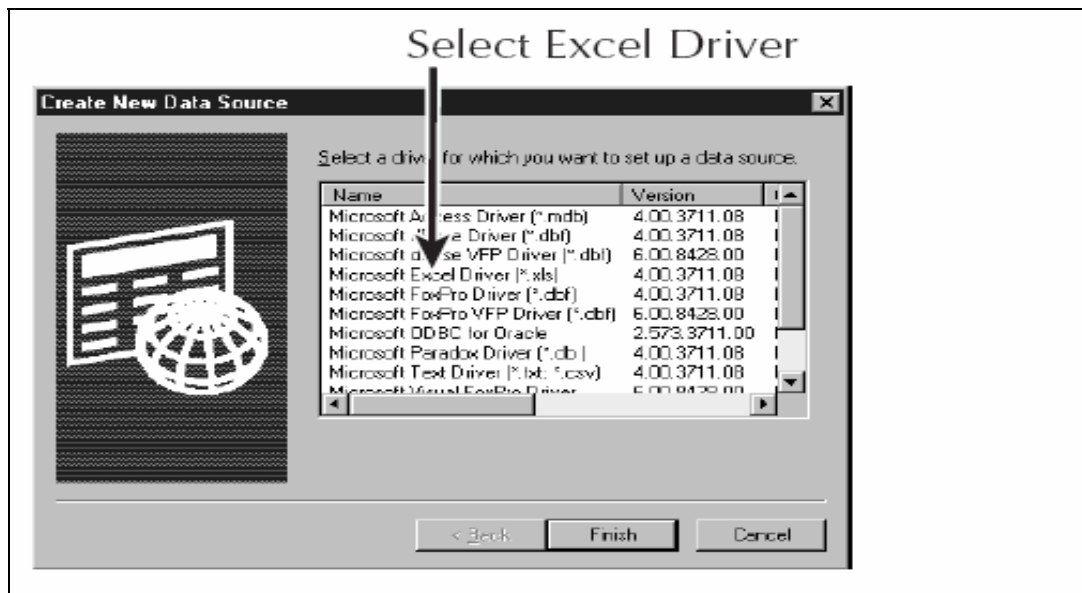
έκδοση της Microsoft Access 2000 η οποία είναι απόλυτα συμβατή με την εφαρμογή της ODBC. Τελειώνοντας την μικρή μας παρουσίαση, η οποία αναφέρεται στην ODBC θα πρέπει να σημειώσουμε ότι στην περίπτωση εκείνη όπου ζητηθεί από την ODBC να αντλήσει πληροφορίες από ένα μη υπαρκτό πίνακα ή από ένα πρόσφατα διαγραμμένο πίνακα θα αντλήσει στοιχεία από τον πιο πλησιέστερα σε αυτόν και θα δημιουργήσει ένα εικονικό πίνακα σε μορφή excel, και ο οποίος θα είναι άμεσα προσβάσιμος σε εμάς, αυτός ο πίνακας θα περιέχει όλα τα στοιχεία όπου θα θέλει ο χρήστης.

Επίσης, σημειώστε ότι τα στοιχεία που αναφέρονται σε ημερομηνία και σε χρόνο μπορούν να παρερμηνευθούν από τον οδηγό της ODBC εάν εμφανίζονται κενά μέσα στη βάση μας. Τέλος θα πρέπει να αναφέρουμε ότι η εφαρμογή της ODBC έχει την δυνατότητα εκτός από αρχεία DBC, να επεξεργαστεί και αρχεία FoxPro – ODBC. Με την διαφορά ότι πρέπει να γίνει σωστή εγκατάσταση του οδηγού χρήσης και ότι η επιλογή των χαρακτήρων μέσα στο FoxPro – ODBC πρέπει να αλλάξει σε υπόμνημα θέτοντας εάν θέλετε το μεταξύ τους διάστημα σαν ένα φυσικό αριθμό. Για να πραγματοποιήσουμε μία σύνδεση της βάσης δεδομένων μας με την ODBC θα πρέπει ο διαχειριστής της εφαρμογής να προβεί στις παρακάτω ενέργειες:

- 1 Επιλέγουμε <<Έναρξη>> → <<πίνακα ελέγχου>> → <<Εργαλεία διαχείρισης>> → <<Πηγές δεδομένων (ODBC)>>
- 2 Επιλέγουμε <<DNS συστήματος>> ή <<DNS Χρηστών>> (σχήμα2.2.2).
- 3 Επιλέγουμε <<Add>>.
- 4 Επιλέγουμε με διπλό κλικ την μορφή της βάσης μας (π.χ. MS Excel driver)
- 5 Επιλέξτε στο τέλος του οδηγού την δημιουργία ενός νέου ODBC driver, και αμέσως ανοίγει ένα νέο παράθυρο διαλόγου όπου πρέπει να εισάγουμε τα στοιχεία της βάσης που θέλουμε να δημοσιεύσουμε (Σχήμα 2.5.3) .
- 6 Εισάγουμε ένα νέο όνομα για την βάση μας με το οποίο θα μπορούμε να πραγματοποιήσουμε την σύνδεση .
- 7 Επιλέγουμε την κατάλληλη έκδοση του λογισμικού της βάσης δεδομένων που θα χρησιμοποιήσουμε (Σχήμα 2.5.4).

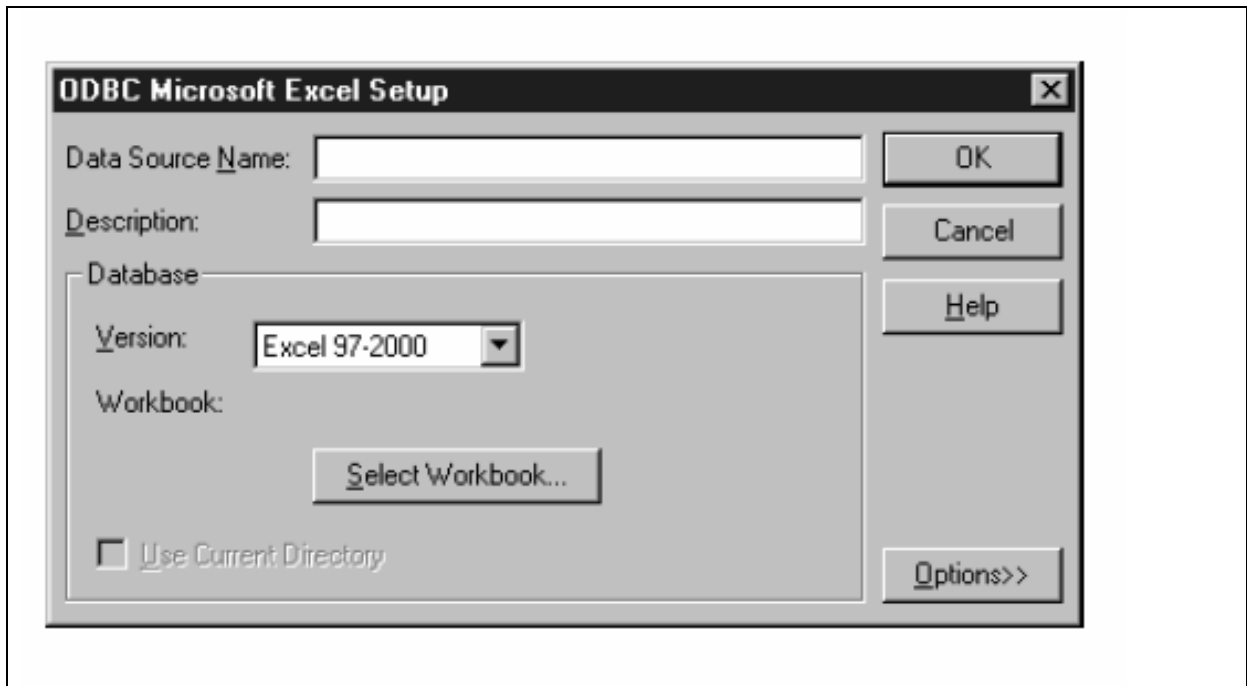


Σχήμα 2.5.2 Εικόνα ενεργοποίησης της ODBC

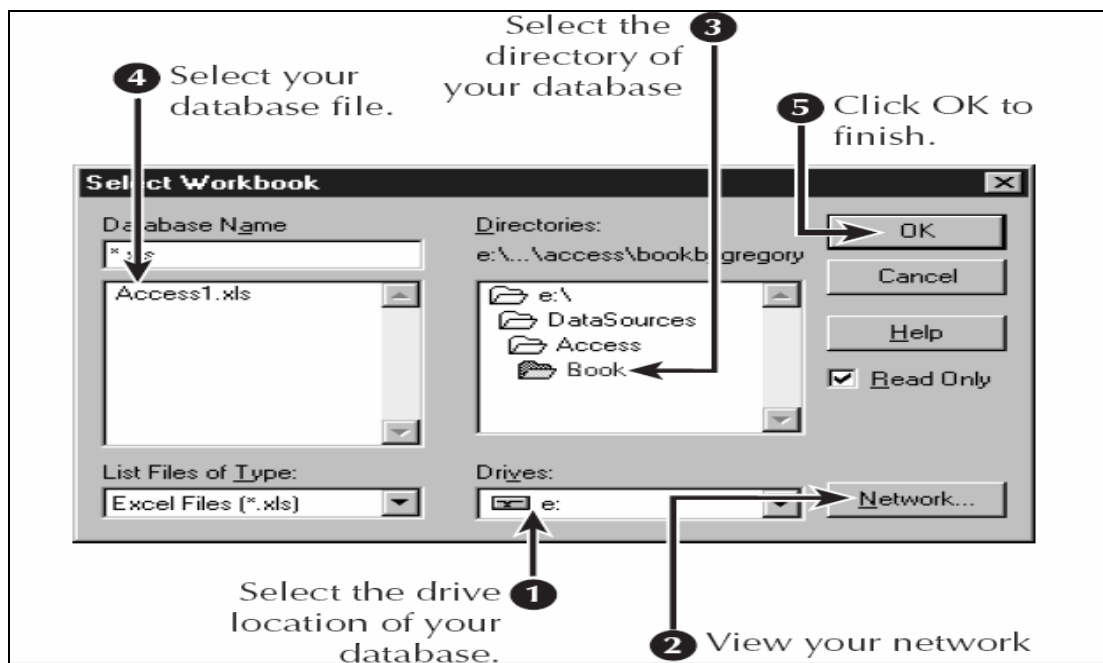


Σχήμα 2.5.3 Εικόνα επιλογής οδηγών για την ODBC

- 8 Επιλέγουμε το πλήκτρο με όνομα Workbook.
- 9 Αναζητούμε και βρίσκουμε την βάση δεδομένων όπου μας ενδιαφέρει (Σχήμα 2.5.5).
- 10 Επιλέγουμε <<OK>> για να βγούμε από τον οδηγό επιλογής της βάσης δεδομένων μας.



Σχήμα 2.5.4 Εικόνα Επιλογής ονόματος της βάσης



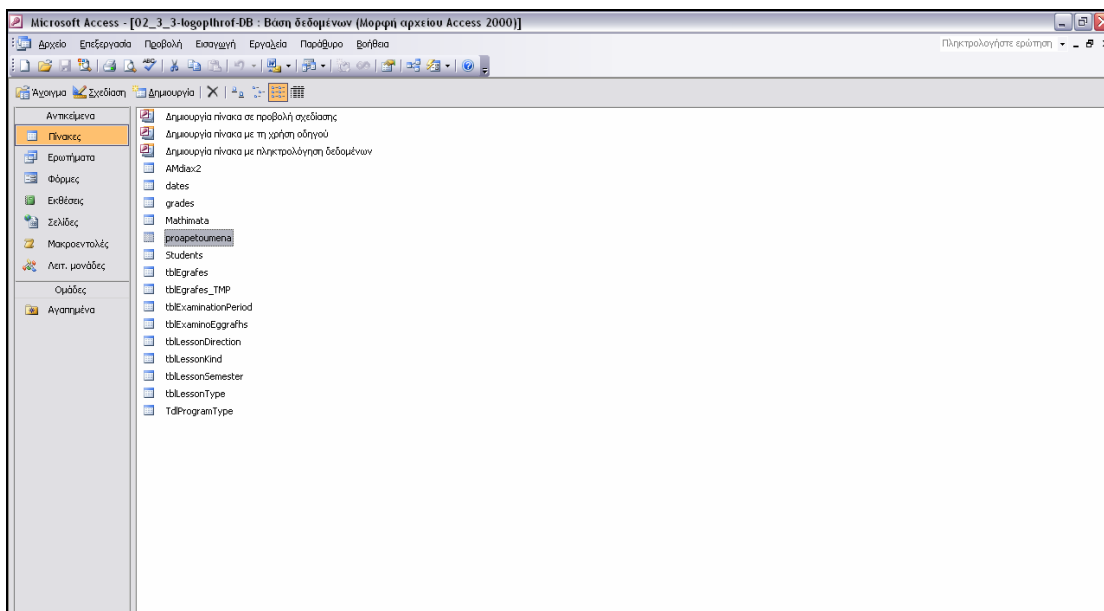
Σχήμα 2.5.5 Εικόνα επιλογής της Βάσης.

3. Σχεδίαση Εφαρμογής

3.1 Microsoft Access

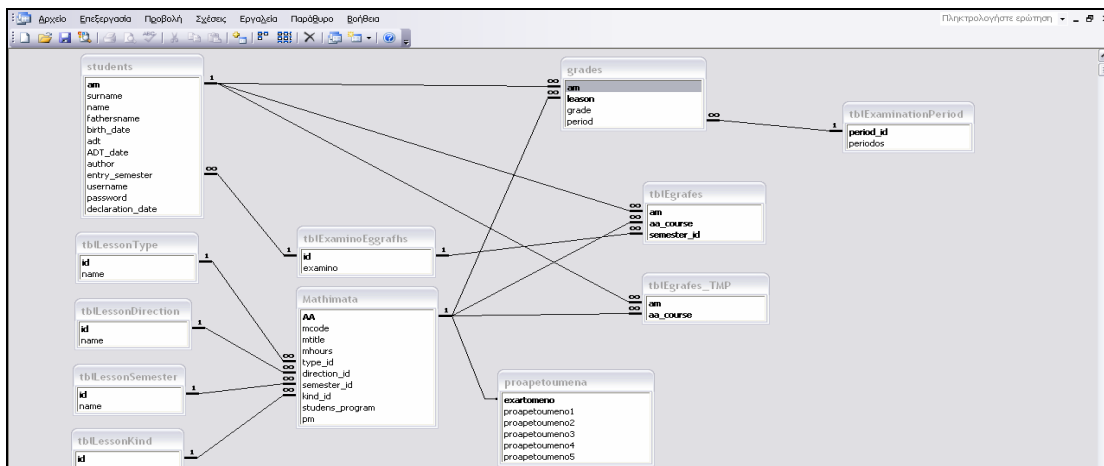
Κατά την διάρκεια αυτής της πτυχιακής εργασίας δημιουργήσαμε και εργαστήκαμε σε μία βάση δεδομένων. Αυτή η βάση είναι η Access της Microsoft μέσα από το γνώριμο περιβάλλον των Office. Πιο συγκεκριμένα χρησιμοποιήσαμε Microsoft Office Access 2000. Μέσω αυτού του περιβάλλοντος σχεδιάσαμε μια βάση δεδομένων με την οποία ανταλλάσσουμε πληροφορίες μέσω του διαδικτύου. Η όλη ιδέα στηρίζεται σε διάφορους πίνακες όπου μέσα περιέχουν τα δεδομένα και σε κάποιες σχέσεις που έχουμε δημιουργήσει.

Στο σχήμα 3.1.1 εμφανίζεται η κεντρική εικόνα της βάσης δεδομένων μας.



Σχήμα 3.1.1 Κεντρική εικόνα της βάσης δεδομένων μας

Ενώ οι σχέσεις που υπάρχουν μεταξύ των πινάκων εμφανίζονται στο σχήμα 3.1.2



Σχήμα 3.1.2 Εικόνα των σχέσεων της βάσης δεδομένων μας

Για το σωστό χειρισμό μιας βάσης δεδομένων αυτό που αρχικά πρέπει να επιλυθεί είναι το πως θα διαχωριστούν τα δεδομένα σε πίνακες ώστε να είναι γρήγορη και εύχρηστη η αναζήτηση τους. Οπότε μελετώντας τα δεδομένα που έπρεπε να επεξεργαστούμε, καταλήξαμε ότι έπρεπε να δημιουργηθούν τέσσερις (4) βασικοί πίνακες όπου έπρεπε να ταξινομήσουμε τα δεδομένα μας.

Οι πίνακες αυτοί είναι :

- Grades
- Mathimata
- Students
- tblEgrafes

am	lesson	grade	period
2542	103	5	7
2542	104	6	9
2542	105	6	9
2542	106	5	5
2542	107	5	7
2542	108	5	7
2542	109	6,8	5
2542	110	5	5
2542	111	5	5
2542	112	5	5
2542	113	9	7
2542	114	5,5	9
2542	115	6,5	9
2542	116	8	13
2542	117	6	9
2542	118	8,5	9
2542	119	5	11
2542	120	7	17
2542	121	6	7
2542	122	7	9
2542	123	5	7
2542	124	6	9
2542	125	8,5	9
2542	126	8	15
2542	127	7,1	13
2542	128	6	11
2542	129	7,5	11
2542	130	5	15
2542	131	8	9
2542	132	5	11
2542	133	7,5	22
2542	134	6,1	22
2542	135	6,5	11
2542	136	6,6	11

Σχήμα 3.1.3 Εμφάνιση του πίνακα grades

Μέσα στον πίνακα grades (σχήμα 3.1.3) έχουμε τέσσερις στήλες:

A. am → Εδώ έχουμε το Αριθμό Μητρώου κάθε σπουδαστή όπου είναι και αποκλειστικό για κάθε σπουδαστή.

B. leason → Εδώ έχουμε την κωδικοποίηση του μαθήματος. Η κωδικοποίηση αυτή έχει γίνει από το ΤΕΙ.

Γ. grade → Μας εμφανίζει το βαθμό που έχει ο σπουδαστής σε κάθε μάθημα.

Δ. period → Εδώ έχουμε την περίοδο που ο σπουδαστής πήρε αυτό τον βαθμό. Ο Κωδικός του period παίρνεται από τον υπό-πίνακα tblExaminationPeriod (σχήμα 3.1.4). Στον πίνακα tblExaminationPeriod έχουμε δυο στήλες:

A. period_id → Είναι ο κωδικοποιημένη αρίθμηση κάθε εξεταστικής περιόδου, και εμφανίζεται και στους κύριους πίνακες της βάσης. Στη συγκεκριμένη εφαρμογή υπάρχουν κωδικοί έως το 2020, αλλά αν εξαντληθούν πολύ εύκολα ο κύριος χρήστης του προγράμματος μπορεί να εισάγει νέες χρονικές περιόδους.

B. periodos → Είναι ονομαστικά οι χρονικές περιόδους που διεξάγονται εξετάσεις στο εκπαιδευτικό ίδρυμα.

period_id	periodos
1	2000-2001 A' Χειμερινή
2	2000-2001 B' Χειμερινή
3	2000-2001 A' Εαρινή
4	2000-2001 B' Εαρινή
5	2001-2002 A' Χειμερινή
6	2001-2002 B' Χειμερινή
7	2001-2002 A' Εαρινή
8	2001-2002 B' Εαρινή
9	2002-2003 A' Χειμερινή
10	2002-2003 B' Χειμερινή
11	2002-2003 A' Εαρινή
12	2002-2003 B' Εαρινή
13	2003-2004 A' Χειμερινή
14	2003-2004 B' Χειμερινή
15	2003-2004 A' Εαρινή
16	2003-2004 B' Εαρινή
17	2004-2005 A' Χειμερινή
18	2004-2005 B' Χειμερινή
19	2004-2005 A' Εαρινή
20	2004-2005 B' Εαρινή
22	2005-2006 A' Χειμερινή
23	2005-2006 B' Χειμερινή
24	2005-2006 A' Εαρινή
25	2005-2006 B' Εαρινή
26	2006-2007 A' Χειμερινή
27	2006-2007 B' Χειμερινή
28	2006-2007 A' Εαρινή
29	2006-2007 B' Εαρινή
30	2007-2008 A' Χειμερινή
31	2007-2008 B' Χειμερινή
32	2007-2008 A' Εαρινή
33	2007-2008 B' Εαρινή
35	2008-2009 A' Χειμερινή

Σχήμα 3.1.4 Εμφάνιση Πίνακα tblExaminationPeriod

AA	mcode	mtitle	mhours	type_id	direction_id	semester_id	kind_id	studens_progr	pm
1	TL1001	Ηλεκτρονικά Στοιχεία	4	Θεωρία	Γενικά	A	Υποχρεωτικό	2,0	
2	TL1002	Ηλεκτρικά Κυκλώματα I	2	Θεωρία	Γενικά	A	Υποχρεωτικό	2,0	
3	TL1004	Φυσική I	2	Θεωρία	Γενικά	A	Υποχρεωτικό	2,0	
4	TL1005	Μαθηματικά	4	Θεωρία	Γενικά	A	Υποχρεωτικό	2,0	
5	TL1006	Οικονομία/Επιχειρήσεων	2	Θεωρία	Γενικά	A	Υποχρεωτικό	2,0	
6	TL1007	Αγγλικά I	2	Θεωρία	Γενικά	A	Υποχρεωτικό	2,0	
7	TL1101	Ηλεκτρονικά Στοιχεία	2	Εργαστήριο	Γενικά	A	Υποχρεωτικό	2,0	
8	TL1102	Ηλεκτρικά Κυκλώματα I	2	Εργαστήριο	Γενικά	A	Υποχρεωτικό	2,0	
9	TL1103	Τεχνολογία Εξαρτημάτων κακοτασκειες	4	Εργαστήριο	Γενικά	A	Υποχρεωτικό	2,0	
10	TL1104	Φυσική I	2	Εργαστήριο	Γενικά	A	Υποχρεωτικό	2,0	
11	TL2001	Ηλεκτρονική I	3	Θεωρία	Γενικά	B	Υποχρεωτικό	2,0	
12	TL2002	Ψηφιακά Κυκλώματα I	2	Θεωρία	Γενικά	B	Υποχρεωτικό	2,0	
13	TL2003	Ηλεκτρικά Κυκλώματα II	2	Θεωρία	Γενικά	B	Υποχρεωτικό	2,0	
14	TL2006	Μαθηματικά με Η/Υ I	2	Θεωρία	Γενικά	B	Υποχρεωτικό	2,0	
15	TL2007	Προγραμματισμός Η/Υ	3	Θεωρία	Γενικά	B	Υποχρεωτικό	2,0	
16	TL2008	Αγγλικά II	2	Θεωρία	Γενικά	B	Υποχρεωτικό	2,0	
17	TL2009	Αρχές Δοσής/Επιχειρήσεων	2	Θεωρία	Γενικά	B	Υποχρεωτικό	2,0	
18	TL2101	Ηλεκτρονική I	2	Εργαστήριο	Γενικά	B	Υποχρεωτικό	2,1	
19	TL2102	Ψηφιακά Κυκλώματα I	2	Εργαστήριο	Γενικά	B	Υποχρεωτικό	2,0	
20	TL2103	Ηλεκτρικά Κυκλώματα II	2	Εργαστήριο	Γενικά	B	Υποχρεωτικό	2,0	
21	TL2106	Μαθηματικά με Η/Υ I	2	Εργαστήριο	Γενικά	B	Υποχρεωτικό	2,0	
22	TL2107	Προγραμματισμός Η/Υ	2	Εργαστήριο	Γενικά	B	Υποχρεωτικό	2,0	
23	TL3001	Ηλεκτρονική II	3	Θεωρία	Γενικά	Γ	Υποχρεωτικό	2,0	
24	TL3002	Ηλεκτρικές & Ηλεκτρονικές Μετρήσεις	2	Θεωρία	Γενικά	Γ	Υποχρεωτικό	2,0	
25	TL3004	CAD & Κατασκευή II	3	Θεωρία	Γενικά	Γ	Υποχρεωτικό	2,0	
26	TL3005	Εισαγ. Ηλεκτρομαγνητισμού	2	Θεωρία	Γενικά	Γ	Υποχρεωτικό	2,0	
27	TL3007	Μαθηματικά με Η/Υ II	2	Θεωρία	Γενικά	Γ	Υποχρεωτικό	2,0	
28	TL3008	Ιστορία Τεχνολογίας	2	Θεωρία	Γενικά	Γ	Υποχρεωτικό	2,0	
29	TL3009	Αγγλικά III	2	Θεωρία	Γενικά	Γ	Υποχρεωτικό	2,0	
30	TL3101	Ηλεκτρονική II	2	Εργαστήριο	Γενικά	Γ	Υποχρεωτικό	2,2	
31	TL3102	Ηλεκτρικές & Ηλεκτρονικές Μετρήσεις	2	Εργαστήριο	Γενικά	Γ	Υποχρεωτικό	2,0	
32	TL3104	CAD & Κατασκευή II	2	Εργαστήριο	Γενικά	Γ	Υποχρεωτικό	2,0	

Σχήμα 3.1.4 Εικόνα πίνακα Mathimata

Στο σχήμα 3.1.4 παρουσιάζουμε τον πίνακα που αναφέρεται στα Μαθήματα. Όπως βλέπουμε έχει δέκα διαφορετικά πεδία. Ας αρχίσουμε να αναλύουμε ένα-ένα πεδίο ξεχωριστά.

A. AA → Αναφέρεται στο κωδικό κάθε μαθήματος. Αυτός ο αριθμός είναι αποκλειστικός για κάθε μάθημα για αυτό και χρησιμοποιείτε σαν πρωτεύον κλειδί του πίνακα. Η κεντρική ιδέα του κωδικού του μαθήματος έχει δημιουργηθεί για να αντιμετωπίσουμε το πρόβλημα που προέκυπτε από τα διαφορετικά προγράμματα σπουδών που τρέχουν παράλληλα σε ένα ακαδημαϊκό έτος. Και ενώ με ίδιους κωδικούς μαθημάτων έχουμε περιγραφή διαφορετικών μαθημάτων

Πχ. Για το ΤΕΙ Κρήτης στο τμ. Ηλεκτρονικής έχουμε δύο προγράμματα σπουδών. Στο ένα πρόγραμμα έχουμε το μάθημα Ηλεκτρονικά Στοιχεία με κωδικό μαθήματος TL 1001, στο άλλο πρόγραμμα σπουδών έχουμε το μάθημα Ηλεκτρονική Φυσική με τον ίδιο κωδικό δηλαδή TL 1001.

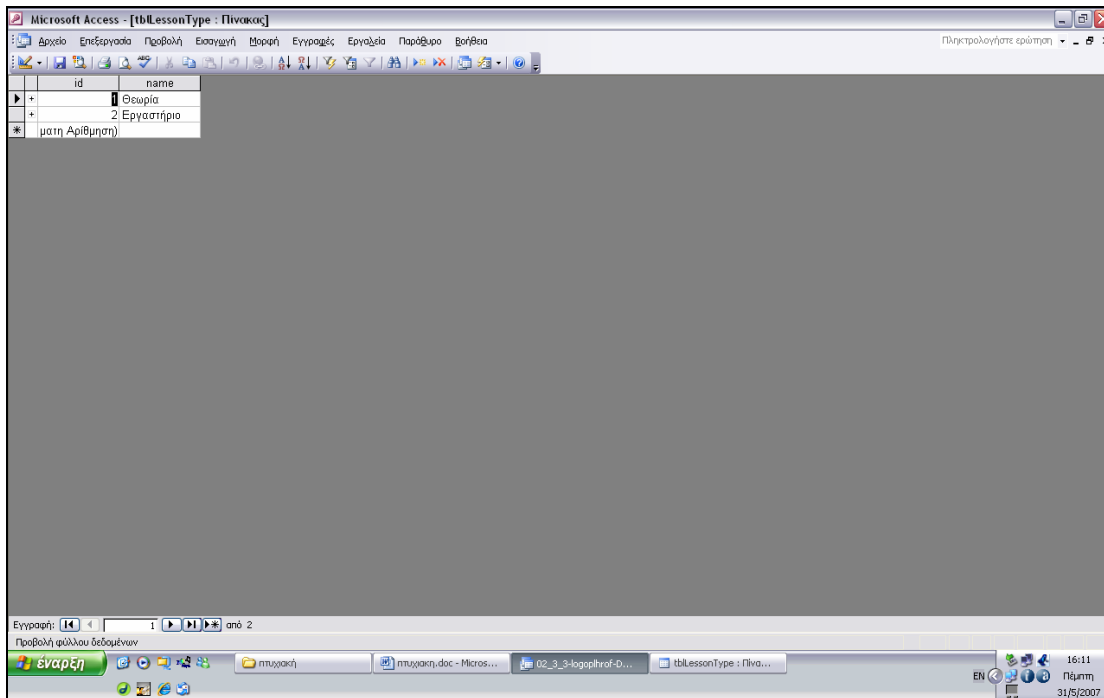
B. mcode → Είναι ο κωδικός του μαθήματος που δίνεται για κάθε μάθημα από την σχολή. Η διαμόρφωση (format) του κωδικού αυτού είναι ως εξής: TL1001 για θεωρία και TL1101 όπου με μια ματιά καταλαβαίνουμε το είδος, το εξάμηνο και αν είναι θεωρία ή εργαστήριο.

Γ. mhours → Μας παρέχει πληροφορίες για τις ώρες διδασκαλίας κάθε μαθήματος.

Δ. type_id → Τα μαθήματα είναι χωρισμένα σε δύο βασικές κατηγορίες αυτές οι κατηγορίες είναι οι εξής:

- 1) Θεωρία
- 2) Εργαστήριο

Σε αυτή τη στήλη του πίνακα τα στοιχεία αντλούνται από τον πίνακα tblLessonType (σχήμα 3.1.5).



Σχήμα 3.1.5 Εικόνα από τον πίνακα tblLessonType

Στο σχήμα 3.1.5 βλέπουμε δυο στήλες. Η πρώτη περιέχει ένα κωδικό και η δεύτερη το όνομα εκείνο που θα μας εμφανιστεί στην περιγραφή του μαθήματος μέσα στο κύριο πίνακα των μαθημάτων.

Ε. direction_id → Άλλη μια μεγάλη κατηγοριοποίηση των μαθημάτων είναι η εξής:

1. Γενικά μαθήματα, όπου τα παρακολουθούν όλοι οι σπουδαστές ανεξαιρέτως.
2. Μαθήματα που πρέπει να παρακολουθήσουν οι σπουδαστές ανάλογα με την κατεύθυνση που έχουν επιλέξει. π.χ. τομέας αυτοματισμού (B) ή τομέας τηλεπικοινωνιών (A)
3. Τελευταία κατηγορία είναι τα κατ' επιλογήν υποχρεωτικά όπου αναφερόμαστε σε παλιότερο πρόγραμμα σπουδών.

Και σε αυτό εδώ πέρα χρησιμοποιούμε το ίδιο σκεπτικό με τον πίνακα των μαθημάτων. Δηλαδή μέσα στο κύριο πίνακα έχουμε κωδικούς αριθμούς και τα στοιχεία αντλούνται από έναν άλλο υπό-πίνακα. Πιο συγκεκριμένα αυτός ο πίνακας που περιέχει τα συγκεκριμένα στοιχεία είναι ο tblLessonDirection (σχήμα 3.1.6):

id	name
1	Γενικά
2	Α
3	Β
4	Α και Β
5	Κατέπλησθην Υποχρεωτικά μαθη Αριθμηση)

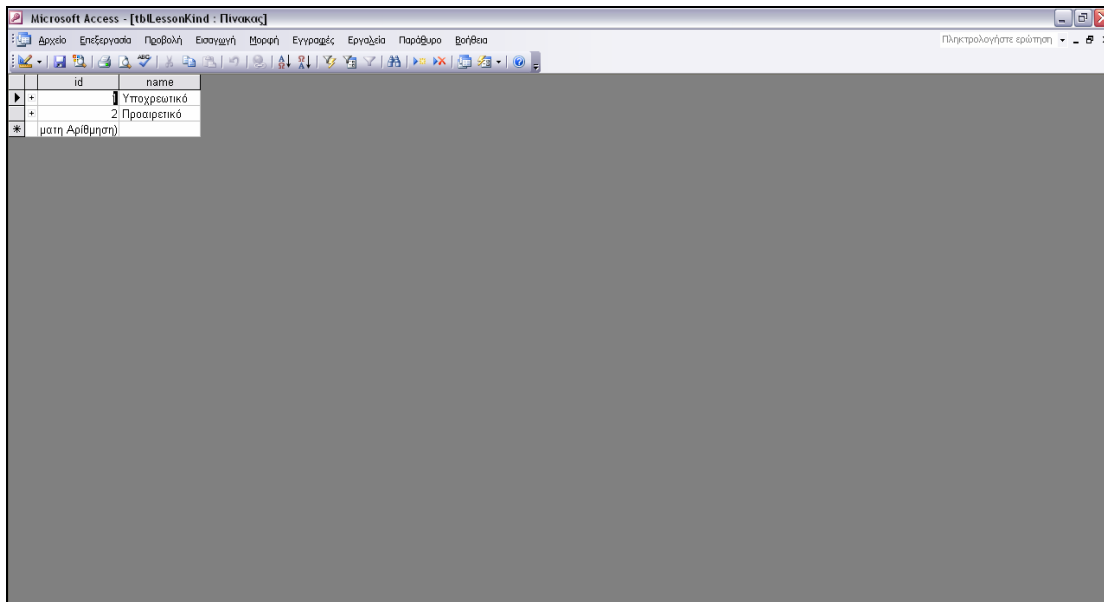
Σχήμα 3.1.6 Εικόνα του πίνακα tblLessonDirection

Στ. semester_id → Εδώ έχουμε το εξάμηνο που γίνεται το αντίστοιχο μάθημα. Και εδώ τα στοιχεία αντλούνται από κάποιο άλλο πίνακα και αυτός είναι ο tblLessonSemester (σχήμα 3.1.7) όπου έχουμε δύο στήλες. Στη μία στήλη έχουμε ένα κωδικό αριθμό του εξαμήνου και στην άλλη την κοινή ονομασία του.

id	name
1	Α
2	Β
3	Γ
4	Δ
5	Ε
6	ΣΤ
7	Ζ
8	Η
9	Θ
	μαθη Αριθμηση)

Σχήμα 3.1.7 Εικόνα πίνακα tblLessonSemester

Ζ. kind_id → Τώρα θα μιλήσουμε για μια άλλη κατηγοριοποίηση των μαθημάτων. Αυτή είναι σύμφωνα με το αν ένα μάθημα είναι υποχρεωτικό για ένα σπουδαστή ή είναι προαιρετικό. Στον κύριο πίνακα τα στοιχεία εισάγονται με κωδικούς αριθμούς. Η αποκωδικοποίηση αυτών των κωδικών πραγματοποιείτε στον tblLessonKind (σχήμα 3.1.8) και έτσι εμείς καταφέρνουμε να δούμε τις κανονικές τους ονομασίες. Η τεχνολογία της σχεδίασης αυτού του πίνακα είναι όμοια με τους υπό-πίνακες που περιγράψαμε παραπάνω, δηλαδή εισάγετε ένα id το οποίο μεταγλωττίζεται και εμφανίζεται σε εμάς με την κανονική ονομασία του.



Σχήμα 3.1.8 Εικόνα από πίνακα tblLessonKind

Η. *studens_program* → Κάθε μία σχολή έχει κάποια προγράμματα σπουδών που τρέχουν παράλληλα. Λέγοντας πρόγραμμα σπουδών εννοούμε ένα σύνολο μαθημάτων που πρέπει να παρακολουθήσει ένας σπουδαστής και να τα περάσει με επιτυχία προκειμένου να αποκτήσει το πτυχίο του. Στο ΤΕΙ Ηλεκτρονικής αυτή τη στιγμή τρέχουν παράλληλα δύο διαφορετικά προγράμματα σπουδών. Η κατάταξη των σπουδαστών στο αντίστοιχο πρόγραμμα γίνεται βάση του Αριθμού μητρώο τους, δηλαδή όσο πιο παλιό είναι το ΑΜ τόσο πιο παλιό πρόγραμμα πρέπει να παρακολουθήσει. Έτσι λοιπόν και εγώ με αυτό το σκεπτικό κατηγοριοποίησα τα μαθήματα σύμφωνα με το πρόγραμμα όπου ανήκουν.

Θ. *pm* → Σε αυτή τη στήλη έχουμε τα μαθήματα εκείνα που πρέπει να περαστούν προκειμένου να παρακολουθήσουμε την συνέχεια αυτών των μαθημάτων. Αυτά τα μαθήματα είναι τα λεγόμενα μαθήματα αλυσίδας. Στη παρούσα στήλη δεν βλέπουμε τα ονόματα των μαθημάτων αλλά τα συνολικά μαθήματα που πρέπει να περάσουμε. Π.χ. Για να παρακολουθήσουμε το μάθημα Ψηφιακά κυκλώματα 2 πρέπει να έχουμε περάσει τα Ψηφιακά κυκλώματα 1. Έτσι λοιπόν στο μάθημα Ψηφιακά κυκλώματα 2 στην στήλη *pm* βλέπουμε τον αριθμό 1 όπου εννοεί ότι πρέπει να περάσουμε ένα μάθημα (δηλαδή τα ψηφιακά 1) για να το παρακολουθήσουμε το μάθημα που επιθυμούμε (ψηφιακά 2).

Τον επόμενο πίνακα που θα μελετήσουμε είναι ο πίνακας *Students* (σχήμα 3.1.9)

am	surname	name	fathersname	birth_date	adt	ADT_date	author	entry_semester	username	password	declaration_date
2672	Γουγούλη	Μαρία	Κωνσταντίνος	13/8 /1981	P666719	15/5 /1996	AT Ν. Σάντας	12	gougou	240782	1/1/1990
2673	Ατσάλη	Μαρίνα	Παναγιώτης	15/10/1982	T329604	31/3 /1999	YA Ηρακλείου	20	atsali	240782	1/1/1990
2693	Φριμουλάκη	Σοφία	Νικόλαος	21/5 /1982	Z668193	1 /9 /1999	AT Αγ. Μύρωνας	10	free	123456	1/1/1990
2699	Ηλίας	Μιχάλης χαράλαμπος	Γεώργιος	24/7 /1982	A9935743	25/8 /2006	AT Περιφέρειας	22	mihalisxh	240782	1/1/1990
2713	Χασάπη	Αικατία	Εμμανουήλ	1 /11/1981	Σ149659	14/5 /1996	AT Πύκκης	12	xas	222222	1/1/1990
2787	Ρενίσης	Εμμανουήλ	Ιωάννης	14/1 /1983	Σ450745	8 /5 /1997	AT Θήρας	12	ren	123456	1/1/1990
								0			1/1/1990

Σχήμα 3.1.9 Εικόνα από τον πίνακα Students

Ο πίνακας αυτός έχει δέκα επιμέρους στήλες τις οποίες θα μελετήσουμε αναλυτικότερα παρακάτω. Σε γενικές γραμμές μέσα σε αυτό τον πίνακα θα βρούμε τα στοιχεία όλων των σπουδαστών που έχουν γραφτεί στο ΤΕΙ. Λέγοντας ενεργοί σπουδαστές εννοούμε ότι παρακολουθούν ή έχουν δηλωμένο ένα τουλάχιστον μάθημα.

A. → am: Εδώ έχουμε τον αριθμό μητρώου του σπουδαστή. Κατά την εγγραφή του στο τμήμα ο σπουδαστή δέχεται ένα αποκλειστικό αριθμό μητρώου από την γραμματεία. Ο αριθμός αυτός είναι μοναδικός για κάθε σπουδαστή και δεν μπορεί να αλλάξει, δηλαδή από την στιγμή που ένας φοιτητής εγγράφεται στο τμήμα μέχρι και την στιγμή που θα πάρει το πτυχίο του αυτός ο αριθμός μητρώου τον ακολουθεί.

B. → surname: Όπως καταλαβαίνουμε και από την ονομασία της στήλης εδώ έχουμε το επίθετο του σπουδαστή.

Γ. → name : Εισάγουμε το όνομα το σπουδαστή.

Δ. → fathersname: Το πατρώνυμο.

E. → birth_date: Ημερομηνία γέννησης σπουδαστή.

Στ. → adt: Αριθμός δελτίου αστυνομικής ταυτότητας σπουδαστή.

Z. → ADT_date: Ημερομηνία εκτύπωσης ταυτότητας.

H. → author: Εκδούσα αρχή.

Θ. → entry_semester: Ημερομηνία εγγραφής στη σχολή.

I. → username: Κατά την εισαγωγή μας στην εφαρμογή έχουμε μια login φόρμα. Σ αυτή τη φόρμα πρέπει να εισάγουμε το username μας.

Οι λόγοι που επιλέξαμε να χρησιμοποιήσουμε αυτή την βάση δεδομένων δεν είναι τυχαίοι.

Ένας από τους κυριότερους λόγους είναι η απόλυτη συνεργασία του εξυπηρετητή μας με αυτή τη μορφή της βάσης, δεύτερον, ότι το γνώριμο γραφικό περιβάλλον της Microsoft διευκόλυνε αρκετά την περάτωση αυτής της εργασίας και τρίτον, ότι μέσω της Access έχουμε πρόσβαση και στο γραφικό περιβάλλον αλλά και στο ερωτήματα που δημιουργήσαμε.

3.2 Σχεδίαση Σελίδων JSP

Όλο το πρόγραμμά μας στηρίζεται σε δυναμικές σελίδες σχεδιασμένες με την τεχνολογία των jsp. Καταρχήν θα απαριθμήσουμε τις σελίδες αυτές και μετά θα δούμε αναλυτικά το ρόλο καθεμίας σελίδας ξεχωριστά. Οι σελίδες επιγραμματικά είναι:

- A. → index.htm
- B. → central.jsp
- Γ. → analitikh.jsp
- Δ. → dhlwshrun.jsp
- E. → password.jsp
- ΣΤ. → confirmation.jsp
- Z. → update_pass.jsp
- H. → dhlwsh.jsp
- Θ. → telikieggraphh.jsp
- I. → confirmationrun.jsp
- Ia. → printleasson.jsp

Η πρώτη σελίδα που συναντάμε είναι η index.htm. Σχεδιαστικά αυτή η σελίδα στηρίζεται σε HTML κώδικα και στην ουσία πραγματοποιείται μια επιβεβαίωση στοιχείων του εκάστοτε χρήστη της εφαρμογής.

Αφού γίνει η επιβεβαίωση των στοιχείων θα μεταβούμε στην επόμενη σελίδα όπου είναι central.jsp. Σχεδιαστικά η σελίδα αυτή κρύβει ένα sql ερώτημα που δέχεται τα δεδομένα μας, κάνει την επιβεβαίωση και μας παρουσιάζει τα στοιχεία του σπουδαστή.

```
rs = stmt.executeQuery("SELECT am, surname, name, fathersname, adt, username, password FROM Students WHERE username= '"+T1+"' AND password= '"+T2+"' ");
```

Επίσης σε αυτή την σελίδα ο κάθε σπουδαστής βλέπει και τις επιλογές που έχει σε αυτή την εφαρμογή.

- Οι επιλογές είναι :
1. Ανανέωση εγγραφής
 2. Αναλυτική Βαθμολογία
 3. Εκτύπωση της δήλωσης εγγραφής μας
 4. Αλλαγή του κωδικού πρόσβασης και
 5. Την έξοδο από το πρόγραμμα μας.

Εάν ο σπουδαστής επιλέξει την ανανέωση εγγραφής το πρόγραμμα μας εμφανίζει την σελίδα `dhlwsh.jsp`. Η πρώτη εικόνα που θα συναντήσουμε είναι όλα τα μαθήματα που αντιστοιχούν στο πρόγραμμα σπουδών μας και μπροστά από κάθε μάθημα ένα `checkbox` για να το επιλέξουμε.

Για την δημιουργία των `checkbox` χρησιμοποιήσαμε ένα συνδυασμό από κώδικα `htm` και `jsp`.

```
<form name="myform" TYPE=GET ACTION=confirmation.jsp
class="checkbox.checktest">
</tr><div align=left>
<br><input name=aa type=checkbox name=aa
value=<%out.println(""+aa1+"");%>
<%out.println(""+code1+", "+title1+", "+ma8+", "+sem2+"");;%>
```

Όσο για την επιλογή του εκάστοτε προγράμματος σπουδών έχει δημιουργηθεί ένας πίνακας στη βάση δεδομένων με όνομα `AMdiax2` όπου θέτει τα όρια για τα τρέχοντα προγράμματα σπουδών και ανάλογα με τον αριθμό μητρώου κάθε σπουδαστή εμφανίζει και τα κατάλληλα μαθήματα. Σε μία μεταβλητή τύπου `integer` εισάγεται ο αριθμός μητρώου (`AM`), και γίνεται η σύγκριση που χρειάζεται προκειμένου να εμφανιστεί το σωστό πρόγραμμα σπουδών.

```
rs2 = stmt2.executeQuery("SELECT * FROM AMdiax2 WHERE AM <=
"+am+" ");
while (rs2.next())
{String AM1= rs2.getString ("AM");
String AA1= rs2.getString("A/A");
ps= Integer.parseInt(AA1);
ts=(ps+1);
}if ( ts == 1 )
```

Και σε αυτή την σελίδα θα παρατηρήσουμε ότι κύρια μεταβλητή παραμένει ο αριθμός μητρώου του σπουδαστή γιατί είναι μοναδικός.

Η επόμενη σελίδα είναι η `confirmationrun.jsp`. Σε αυτή την σελίδα θα δούμε εκτός από `sql` ερωτήματα και άλλες εντολές όπως είναι η `<< for >>` ή η μεταβλητή `<< BigDecimal bdRowCount >>`, όπου είναι υπεύθυνη για την άθροιση των συνολικών ωρών. Η παραπάνω εντολή λειτουργεί εφόσον ενεργοποιηθεί και μία συγκεκριμένη βιβλιοθήκη με το όνομα `java.math`.

Η διαδικασία που μας εισάγει την εντολή `if` έχει να κάνει με την δημιουργία ενός εικονικού πίνακα όπου περιέχει τις ώρες.

```
String [ ] leasson ;
//
//
//
if (leasson != null)
{
for (int i=0; i<leasson.length; i++)
```

```

    {
        rs = stmt.executeQuery("SELECT AA, mcode,
mtitle,mhours FROM Mathimata WHERE AA="+leasson[i]+" ");

```

Ένας δείκτης μετράει τις θέσεις μνήμης του πίνακα και μας παρουσιάζει τα αποτελέσματά μας. Στην κάτω πλευρά της σελίδας θα βρούμε ένα πλήκτρο όπου αναγράφει τον αριθμό μητρώου και μας προωθεί στην επόμενη σελίδα. Σε γενικές γραμμές η σελίδα αυτή μας εμφανίζει τα μαθήματα όπου έχουμε δηλώσει και τον συνολικό αριθμό των ωρών.

Η τελευταία σελίδα όπου αναφέρεται στην ανανέωση εγγραφής είναι η `telikieggraphh.jsp`. Σε αυτή την σελίδα έχουμε την τελική εγγραφή του σπουδαστή και παράλληλα την διαγραφή του από τον πρόχειρο πίνακα εγγραφών.

```

    {
        rs = stmt.executeQuery("SELECT * FROM trexonexamino");
        while (rs.next()) {
            String T3 = rs.getString("examino");
            stmt2.executeUpdate("INSERT INTO
tblEgrafes(am,aa_course,semester_id) VALUES
('"+T+"', '"+codes[i]+"', '"+T3+"'");
        }
        stmt3.executeUpdate("DELETE * FROM tblEgrafes_TMP WHERE am =
"+T4+"");
        out.println("<td> Η εγγραφή σας στα μαθήματα που επιλέξατε
έγινε επιτυχώς </td>");

```

Σε όλες της προηγούμενες σελίδες τα στοιχεία μας αποθηκευόταν στον `tblEgrafes_TMP` και αυτό για το λόγο ότι ένας σπουδαστής μπορεί να αλλάξει αρκετές φορές γνώμη για τα μαθήματα όπου έχει δηλώσει. Η τελική εγγραφή και παράλληλα η ανανέωση της εγγραφής του γίνεται εφόσον τα μαθήματα που έχει δηλώσει εμφανιστούν στον πίνακα `tblEgrafes`.

Επιστρέφοντας στην κεντρική σελίδα της εφαρμογής μας η επόμενη επιλογή που έχουμε είναι η αναλυτική βαθμολογία του σπουδαστή. Για να δούμε την βαθμολογία του σπουδαστή θα μεταβούμε στην σελίδα `analitikh.jsp`. Εδώ πέρα θα συναρτήσουμε ένα sql ερώτημα της παρακάτω μορφής.

```

rs3 = stmt3.executeQuery("SELECT AA,mcode,mtitle,type_id FROM
Mathimata WHERE AA="+title);

```

Στην ουσία ένας δείκτης ελέγχει όλα τα δεδομένα της βάσης δεδομένων μας, τα φιλτράρει και ανάλογα με τον Αριθμό μητρώου που του έχουμε εισαγάγει εμφανίζει και τα επιθυμητά αποτελέσματα.

Η επόμενη επιλογή που έχουμε στην κεντρική σελίδα είναι η αλλαγή του προσωπικού κωδικού εισαγωγής στην εφαρμογή. Η σελίδα όπου ανοίγει είναι η `update_pass.jsp`. Σε αυτή τη σελίδα θα πρέπει να εισάγουμε τον παλιό κωδικό κα αμέσως μας ζητάει να εισάγουμε ένα νέο κωδικό δύο φορές. Στην εφαρμογή μας θα γίνει έλεγχος των νέων κωδικών και εάν είναι ίδιοι η αλλαγή θα γίνει επιτυχώς μέσα στα προσωπικά στοιχεία μας στην βάση δεδομένων, εάν οι νέοι κωδικοί είναι διαφορετικοί το πρόγραμμα θα μας προειδοποιήσει και η αλλαγή δεν θα πραγματοποιηθεί. Η επικείμενη αλλαγή γίνεται για λόγους ασφαλείας για τους χρήστες του προγράμματος και φυσικά αυτή η αλλαγή θα πρέπει να γίνεται σε σύντομα χρονικά διαστήματα. Η σύγκριση των δύο κωδικών πραγματοποιείται στο αρχείο `password.jsp` και συγκεκριμένα στο παρακάτω κομμάτι κώδικα:

```
String T2 = request.getParameter("pass");
        String T3 = request.getParameter("pass1");
        String T4 = request.getParameter("pass2");
        String ps3= "asdfghjkl1g2";
        String ps4= "zxcvbnmlqwer";

        ps3= (""+T3+"");
        ps4= (""+T4+"");

                                                //
                                                //
                                                //
                                                //

if( T3.equals(T4) && ("am") != null)
    {
out.println("<td>Το Password σας αλλαξε με επιτυχία. Το νέοο
σας Password είναι:</td>");
out.println("<td>" +ps3+ " </td></tr>");
    rs2 = stmt2.executeQuery("UPDATE Students SET password =
'"+ps3+"' WHERE am="+am+"");
    }
```

Συγκεκριμένα το πάνω κομμάτι του κώδικα αναφέρεται στην αποστολή και στη λήψη των passwords από την σελίδα `update_pass.jsp` στην `password.jsp` την σύγκρισή του καθώς και την αλλαγή τους. Η σύγκριση γίνεται με την εντολή `<<if>>` που συγκρίνει την μεταβλητή T3 με την T4, αυτές οι μεταβλητές περιέχουν τον κωδικό καθώς επίσης και την επιβεβαίωσή του ενώ όλη διαδικασία του ελέγχου και της σύγκρισης γίνεται με την εντολή `equals` όπου παράλληλα ελέγχει και τον αριθμό μητρώου του σπουδαστή. Αφού γίνει η σύγκριση και η επιβεβαίωση με τον αριθμό μητρώου πραγματοποιείτε η αλλαγή στο πεδίο `password` στον πίνακα `Students` στην βάση δεδομένων μας.

Η επόμενη σελίδα της εφαρμογής μας που θα παρουσιάσουμε είναι η επιλογή της εκτύπωσης των δηλωμένων μαθημάτων μας. Όπως είδαμε

παραπάνω τα δηλωμένα μαθήματα μας εμφανίζονται στον πίνακα tblEgrafes και συγκεκριμένα μέσα στον πίνακα θα βρούμε τα εξής πεδία:

- A) am
- B) aa_course
- Γ) semester_id

Έτσι λοιπόν όταν επιλέξουμε την εκτύπωση της δήλωσής μας, θα μας εμφανιστεί ένας πίνακας με τα μαθήματα που έχουμε δηλώσει, και το ακαδημαϊκό εξάμηνο όπου το δηλώσαμε.

```
rs = stmt.executeQuery("SELECT * FROM tblEgrafes WHERE am="+T1+"  
ORDER BY semester_id");
```

Για λόγους χρηστικότητας σε αυτή την εφαρμογή δεν χρειάζεται να εισάγουμε ξανά τα στοιχεία του σπουδαστή αλλά μας παρουσιάζει κατευθείαν τα μαθήματα όπου έχουμε δηλώσει. Ενώ στην κάτω πλευρά της σελίδας όπου θα μας ανοίξει εμφανίζεται ένα ενεργό κουμπί όπου μας οδηγεί στην έξοδο της εφαρμογής μας.

Η τελευταία επιλογή όπου μας παρέχει η κεντρική σελίδα της εφαρμογής μας είναι η ασφαλής αποσύνδεση από την εφαρμογή μας. Για λόγους ασφαλείας μετά από κάθε έξοδο της εφαρμογής μας πρέπει να εισάγουμε από την αρχή τα δεδομένα μας στην φόρμα εισαγωγής του προγράμματός μας.

Όπως είδαμε παραπάνω στην ανάλυση της βάσης δεδομένων μας, παρατηρήσαμε ότι όλα τα ονόματα που εμφανίζονται είναι καταχωρημένα με κάποιους κωδικούς για λόγους ασφαλείας. Ένα εύλογο ερώτημα που θα δημιουργηθεί είναι πώς είναι δυνατόν από κωδικούς αριθμός εμείς να έχουμε στην οθόνη του υπολογιστή μας τίτλους. Αυτό θα το επιτύχουμε με την βοήθεια κάποιων sql ερωτημάτων όπου εισάγοντας τους κωδικούς και απευθυνόμενοι στον κατάλληλο πίνακα εμφανίζει τα αποτελέσματα που εμείς επιθυμούμε. Οι πίνακες που χρησιμοποιούμε έχουν την χαρακτηριστική ονομασία με την κατάληξη <<ID>>. Ένα χαρακτηριστικό παράδειγμα μπορούμε να το δούμε στο παρακάτω κομμάτι κώδικα που ακολουθεί:

```
if ("title"!=null);  
{  
rs3 = stmt3.executeQuery("SELECT AA,mcode,mtitle,type_id FROM  
Mathimata WHERE AA="+title);  
while (rs3.next()) {  
String mt = rs3.getString("mtitle");  
out.println("<tr><td>" +mt+ "</td>");  
}
```

Όπου ελέγχει εάν η μεταβλητή <<title>> περιέχει κάποια αριθμητική τιμή, τρέχει το ερώτημα <<rs3>>, ανοίγει ο πίνακας <<Mathimata>>, βρίσκει τον συγκεκριμένο κωδικό σε ποιο μάθημα αναφέρεται, και τέλος εμφανίζει σε εμάς τον συγκεκριμένο τίτλο του μαθήματος όπως ακριβώς εμφανίζεται μέσα στη βάση δεδομένων.

4. Οδηγίες Χρήσεως της Εφαρμογής

Όλη την εφαρμογή μας μπορούμε να την δούμε από δύο οπτικές γωνίες. Η μία είναι από την πλευρά του χρήστη και η δεύτερη από την πλευρά του διαχειριστή.

4.1 Οδηγίες Χρήσεως για τον χρήστη της εφαρμογής

Η εφαρμογή μας ξεκινάει ξεκινώντας τον browser του ηλεκτρονικού υπολογιστή μας. Η κεντρική σελίδα της εφαρμογής μας ανοίγει εάν πληκτρολογήσουμε την παρακάτω διεύθυνση <http://127.0.0.1:8080>, Εννοείται ότι αυτή η διεύθυνση είναι εικονική και θα αλλάξει εφόσον η εφαρμογή τρέξει σε ένα ρεαλιστικό περιβάλλον. Η εικόνα της κεντρικής σελίδας εμφανίζεται στο σχήμα 4.1.1

ΤΕΙ ΚΡΗΤΗΣ	
ΠΑΡΑΡΤΗΜΑ ΧΑΝΙΩΝ	
ΕΦΑΡΜΟΓΗ ΔΗΛΩΣΗΣ ΜΑΘΗΜΑΤΩΝ ΕΞ'ΑΠΟΣΤΑΣΕΩΣ	
Επιλέξτε Τμήμα :	Ηλεκτρονικής ▾
Δώστε username :	<input type="text"/>
Δώστε password :	<input type="password"/>
<input type="button" value="Είσοδος"/> <input type="button" value="Καθαρισμός"/>	
Σπουδαστής: Μιχάλης-Χαράλαμπος Ηλίας	

Σχήμα 4.1.1 Εικόνα κεντρικής σελίδας της εφαρμογής

Ο χρήστης θα πρέπει να εισάγει τον προσωπικό του όνομα (username) κωδικό χρήστη (password). Ο σπουδαστής θα μπορεί να παίρνει αυτά τα στοιχεία από την γραμματεία της σχολής μετά από κάποια χειρόγραφη δήλωση που θα συμπληρώνει.

Μετά την εισαγωγή αυτών των στοιχείων και την επιβεβαίωση του ο χρήστης θα εισέρχεται στην ατομική του καρτέλα και θα βλέπει τα προσωπικά του στοιχεία καθώς και τις δυνατότητες που έχει χρησιμοποιώντας αυτήν την εφαρμογή (σχήμα 4.1.2). Οι δυνατότητες που έχουμε μέσω της εφαρμογής μας είναι:

- 1 → Ανανέωση Εγγραφής
- 2 → Αναλυτική Βαθμολογία
- 3 → Αλλαγή κωδικού πρόσβασης
- 4 → Εμφάνιση δηλωμένων μαθημάτων

5 → Έξοδος

Αυτές οι πέντε εφαρμογές είναι οι κυριότερες και οι πιο χρηστικές για ένα σπουδαστή χρήστη αυτής της εφαρμογής, επιπλέον όλες αυτές οι δυνατότητες, εκτός από την ανανέωση εγγραφής η οποία θα είναι σε ισχύ μόνο κατά την διάρκεια των εγγραφών, θα είναι προσβάσιμες καθ' όλη την διάρκεια του έτους.

ΠΛΗΡΗ ΣΤΟΙΧΕΙΑ ΣΠΟΥΔΑΣΤΗ:	
ΑΜ:	2700
ΟΝΟΜΑ:	Νίκος
ΕΠΙΘΕΤΟ:	Παπαδόπουλος
ΟΝΟΜΑ ΠΑΤΡΟΣ:	Πέτρου
ΑΡΙΘΜΟΣ ΔΕΛΤΙΟΥ ΤΑΥΤΟΤΗΤΑΣ: AB342134	
Δηλώσεις Μαθημάτων :	<input type="text" value="2700"/>
Αναλυτική Βαθμολογία :	<input type="text" value="2700"/>
Αλλαγή Κωδικού Πρόσβασης :	<input type="text" value="2700"/>
Εμφάνιση Δηλωμένων Μαθημάτων	<input type="text" value="2700"/>
Έξοδος:	<input type="button" value="έξοδος"/>

Σχήμα 4.1.2: Κεντρική σελίδα της εφαρμογής με τα στοιχεία και της επιλογές ενός σπουδαστή

Ολόκληρη η ιδέα αυτής της πτυχιακής εργασίας ξεκίνησε από την ανάγκη για μια ηλεκτρονική ανανέωση της εγγραφής ενός σπουδαστή. Έτσι λοιπόν αυτή την δυνατότητα θα εξετάσουμε πρώτα. Καθώς θα επιλέξουμε την επιλογή η οποία αναφέρεται σαν δήλωση μαθημάτων, έχουμε μια εικόνα όλων των μαθημάτων όπου αντιστοιχούν στο πρόγραμμα σπουδών μας (σχήμα 4.1.3). Όλα τα μαθήματα εμφανίζονται με ένα checkbox, έτσι λοιπόν επιλέγουμε τα μαθήματα όπου επιθυμούμε να δηλώσουμε για το τρέχον εξάμηνο. Στη τέλος της σελίδας υπάρχει ένα button όπου αναγράφει τον αριθμό του μητρώου μας και μας προωθεί στην επόμενη σελίδα που γίνεται ο έλεγχος των μαθημάτων.

ΤΑ ΣΤΟΙΧΕΙΑ ΣΑΣ ΕΙΝΑΙ:	
ΑΜ:	2700
ΟΝΟΜΑ:	Νίκος
ΕΠΙΘΕΤΟ:	Παπαδόπουλος
ΟΝΟΜΑ ΠΑΤΡΟΣ:	Πέτρου
ΑΡΙΘΜΟΣ ΔΕΛΤΙΟΥ ΤΑΥΤΟΤΗΤΑΣ: AB342134	
Τα μαθήματα που έχετε δικαίωμα να δηλώσετε είναι τα παρακάτω:	
<input type="checkbox"/>	ΤΑΠ024,Γερμανικά Ι,Θεωρία,
<input type="checkbox"/>	ΤΑΠ003,Ηλεκτροακουστική & Συστήματα Ήχου,Θεωρία,
<input type="checkbox"/>	ΤΑΠ013,Γαλλικά Ι,Θεωρία,
<input type="checkbox"/>	ΤΑΠ023,Γαλλικά ΙΙ,Θεωρία,
<input type="checkbox"/>	ΤΑΠ025,Γερμανικά ΙΙ,Θεωρία,
<input type="checkbox"/>	ΤΑΠ026,Γερμανικά ΙΙΙ,Θεωρία,
<input type="checkbox"/>	ΤΑΠ027,Γαλλικά ΙΙΙ,Θεωρία,
<input type="checkbox"/>	ΤΑΠ057,Θεωρία πληροφορίας και κωδικών,Θεωρία,
<input type="checkbox"/>	ΤΑΠ111,Ασφάλεια και υγιεινή,Θεωρία,
<input type="checkbox"/>	ΤΑΠ115,Μηχανουργείο,Εργαστήριο,

Σχήμα 4.1.3 Σελίδα όπου εμφανίζει όλα τα μαθήματα με το checkbox

Ο έλεγχος των μαθημάτων γίνεται τόσο στις ώρες όπου έχει δηλώσει ο σπουδαστής όσο και στα μαθήματα, δηλαδή εάν ένα μάθημα είναι δηλωμένο ή ο σπουδαστής έχει περάσει αυτό το μάθημα τότε εμφανίζεται ένα μήνυμα όπου προειδοποιεί τον σπουδαστή ότι το συγκεκριμένο μάθημα το έχει ήδη δηλωμένο η καταχωρημένο σε παλιότερη εξεταστική περίοδο (σχήμα 4.1.4 και σχήμα 4.1.5).

ΤΛ1007,Αγγλικά Ι,2

exception : java.sql.SQLException: General errorGeneral error

Σχήμα 4.1.4 Μάθημα το οποίο το έχουμε ήδη δηλώσει και προσπαθήσαμε να το δηλώσουμε ξανά

ΤΛ1001,Ηλεκτρονικά Στοιχεία,4 :Αυτο το μάθημα το έχεις καταχωρημένο με 5.00

ΤΛ1104,Φυσική Ι,2 :Αυτο το μάθημα το έχεις καταχωρημένο με 9.00

ΤΛ1103,Τεχνολογια Εξαρτηματων καικατασκευες,4 :Αυτο το μάθημα το έχεις καταχωρημένο με 8.80

Οι συνολικές ώρες που έχειται δηλώσει είναι: 20.0

2699

Σχήμα 4.1.5 Εικόνα με μαθήματα όπου έχουν καταχωρηθεί σε προηγούμενο εξάμηνο

Εάν πάλι ο σπουδαστής δηλώσει το μάθημα που επιθυμεί και δεν υπάρχει πρόβλημα, τότε το σύστημα εμφανίζει ένα μήνυμα επιβεβαίωσης όπου λέει στον χρήστη ότι το μάθημα που έχει επιλέξει καταχωρήθηκε με επιτυχία (σχήμα 4.1.6), Ένας ακόμη έλεγχος που πραγματοποιείτε είναι και ο έλεγχος των συνολικών δηλωμένων ωρών. Κάθε σπουδαστής έχει δικαίωμα να δηλώσει ένα συνολικό αριθμό ωρών ο οποίος προκαθορίζεται από τη γραμματεία, εάν ο σπουδαστής ξεπεράσει τις επιτρεπτές του ώρες το σύστημα προειδοποιεί τον χρήστη ότι έχει ξεπεράσει τις επιτρεπτές γι' αυτόν ώρες και φυσικά δεν γίνεται η ανανέωση της εγγραφής του, ενώ η εφαρμογή πρέπει να ξανατρέξει από την αρχή. Εφόσον γίνει η ανανέωση της εγγραφής, θα πρέπει να κάνουμε μια ασφαλή έξοδο από το σύστημα και να εισάγουμε ξανά τα στοιχεία επιβεβαίωσης μας για να γίνει εκ νέου η εισαγωγή μας στο σύστημα.

Η επόμενη δυνατότητα που μας δίνει το η εφαρμογή μας είναι η εμφάνιση της αναλυτικής βαθμολογίας του σπουδαστή. Επιλέγονται το αντίστοιχο σύνδεσμο μας εμφανίζει την καρτέλα του σπουδαστή με τα προσωπικά του στοιχεία, και ένα πίνακα με όλα τα μαθήματα όπου πρέπει να καταχωρίσει. Δίπλα στην στήλη με τα μαθήματα υπάρχει μία δεύτερη στήλη με τον

βαθμό και δίπλα ακριβώς την χρονική περίοδο όπου καταχωρήθηκε η βαθμολογία του εκάστοτε σπουδαστή (σχήμα 4.1.7).

<p>ΤΛ5104,Συστήματα αυτοματου ελεγχου,2 Το μάθημα που επιλέξατε καταχωρήθηκα</p> <p>ΤΛ5105,Ανάλυση και συνθεσηκυκλωμάτων με Η/Υ,3 Το μάθημα που επιλέξατε καταχωρήθηκα</p> <p>ΤΛ5001,Μικροηλεκτρονικη - VLSI,2 Το μάθημα που επιλέξατε καταχωρήθηκα</p> <p>ΤΛ7001,Δίκτυα Η/Υ,3 Το μάθημα που επιλέξατε καταχωρήθηκα</p> <p>ΤΛ7005,Ποιότητα & Δεοντολογία,2 Το μάθημα που επιλέξατε καταχωρήθηκα</p> <p>Οι συνολικές ώρες που έχετε δηλώσει είναι: 12.0</p> <p>2700</p>
--

Σχήμα 4.1.6 Εικόνα κατοχύρωσης μαθημάτων

Η επόμενη επιλογή της εφαρμογής μας αναφέρεται στην ασφάλεια του όλου συστήματος. Συγκεκριμένα αναφερόμαστε στην διαδικασίας αλλαγής κωδικού εισαγωγής του σπουδαστή. Όπως αναφέραμε παραπάνω, κάθε σπουδαστής κατά την διάρκεια της εγγραφής του θα παραλαμβάνει και ένα κωδικό όπου θα μπορεί να κάνει την ασφαλή εισαγωγή του στο σύστημα. Μετά την πρώτη εισαγωγή του θα πρέπει να αλλάξει τον προσωπικό του κωδικό.

Τίτλος Μαθήματος	Τύπος Μαθήματος	Βαθμός	Περίοδος Εξέτασης
Ηλεκτρονικά Στοιχεία	Εργαστήριο	5.00	2000-2001 Α' Εαρινή
Ηλεκτρικά Κυκλώματα Ι	Θεωρία	5.00	2000-2001 Α' Εαρινή
Ηλεκτρικά Κυκλώματα Ι	Εργαστήριο	5.00	2000-2001 Α' Εαρινή
Τεχνολογία Εξαρτημάτων κακατασκευες	Εργαστήριο	8.80	2000-2001 Α' Εαρινή
Φυσική Ι	Θεωρία	5.00	2000-2001 Α' Εαρινή
Φυσική Ι	Εργαστήριο	9.00	2000-2001 Α' Εαρινή
Μαθηματικά με Η/Υ Ι	Θεωρία	5.00	2001-2002 Α' Χειμερινή
Αρχές Δοήκησης Επιχειρήσεων	Θεωρία	6.00	2001-2002 Α' Χειμερινή
Προγραμματισμός Η/Υ	Εργαστήριο	7.50	2001-2002 Α' Χειμερινή
Μαθηματικά με Η/Υ Ι	Εργαστήριο	5.00	2001-2002 Α' Χειμερινή
Ηλεκτρικά Κυκλώματα ΙΙ	Εργαστήριο	5.50	2001-2002 Α' Χειμερινή
Ψηφιακά Κυκλωματα Ι	Εργαστήριο	6.50	2001-2002 Α' Χειμερινή
ΟικονομίαΕπιχειρήσεων	Θεωρία	7.00	2001-2002 Α' Χειμερινή
Προγραμματισμός Η/Υ	Θεωρία	5.50	2001-2002 Α' Χειμερινή
Ψηφιακά Κυκλωματα Ι	Θεωρία	5.20	2001-2002 Α' Χειμερινή
Ηλεκτρικές & Ηλεκτρονικες Μετρήσεις	Εργαστήριο	5.00	2001-2002 Α' Εαρινή
Ηλεκτρονικά Στοιχεία	Θεωρία	5.00	2001-2002 Α' Εαρινή

Σχήμα 4.1.7 Εικόνα της αναλυτικής βαθμολογίας του σπουδαστή

Μέσα στην αντίστοιχη φόρμα πρέπει να εισάγουμε τον παλιό μας κωδικό και δύο φορές τον καινούργιο μας κωδικό. Θα πρέπει η δύο κωδικοί να είναι όμοιοι γιατί εάν δεν είναι οι ίδιοι το σύστημα δεν θα κάνει την αλλαγή. Για να πραγματοποιηθεί η αλλαγή θα πρέπει να είναι όμοιοι και οι δύο κωδικοί ασφαλείας (σχήμα 4.1.8). Εφόσον οι κωδικοί είναι όμοιοι, γίνεται η αλλαγή που επιθυμούμε. Το επόμενο βήμα που θα κάνουμε είναι να γίνει η έξοδος μας από την εφαρμογή και να ξανακάνουμε εισαγωγή με τα νέα μας πλέον στοιχεία μας. Η εισαγωγή μας στο σύστημα γίνεται με τα νέα

στοιχεία για το λόγο ότι η ανανέωση του κωδικού ασφαλείας γίνεται αμέσως μετά το τρέξιμο της συγκεκριμένης εφαρμογής. Τελειώνοντας την παρουσίαση αυτής της συγκεκριμένης εφαρμογής θα πρέπει να αναφέρουμε ότι ο καινούργιος κωδικός μπορεί να αποτελείτε από 1 έως 8 σύμβολα αποτελούμενο από γράμματα ή αριθμούς ή και τα δύο μαζί.

Δώστε το ΑΜ σας:	<input type="text"/>
Δώστε το Παλιό Password :	<input type="text"/>
Δώστε το νέο Password :	<input type="text"/>
Επιβεβαίωση νέου Password :	<input type="text"/>
<input type="button" value="Είσοδος"/> <input type="button" value="Καθαρισμός"/>	
<p>**Παρακαλώ το συνθηματικό (Password) να αποτελείται από 1 έως 8 χαρακτήρες (αριθμούς και γράμματα)</p>	

Σχήμα 4.1.8 φόρμα αλλαγής κωδικού πρόσβασης

Η επόμενη επιλογή του προγράμματος στην κεντρική σελίδα μας είναι η εκτύπωση της δήλωσης των μαθημάτων όπου πραγματοποιήσαμε στην πρώτη επιλογή της εφαρμογής. Επιλέγοντας αυτήν την επιλογή εμφανίζετε ένας πίνακας με δύο στήλες (σχήμα 4.1.9). Αυτές οι στήλες ονομαστικά είναι:

- 1 → Τίτλος Μαθήματος
- 2 → Δηλωθέν εξάμηνο

Όπως μπορούμε εύκολα να καταλάβουμε ο τίτλος του μαθήματος εμφανίζει το όνομα του μαθήματος όπου έχουμε κάνει την εγγραφή μας και η δεύτερη στήλη αναφέρεται στο εξάμηνο όπου έγινε η εγγραφή. Γενικότερα η κεντρική σκέψη ήταν να δημιουργηθεί μια εύκολη σε χρήση, για τον εκάστοτε χρήστη, εφαρμογή, κάτι το οποίο έχουμε πετύχει. Για την χρήση αυτής της εφαρμογής θα πρέπει να γνωρίζουμε μόνο το όνομα και τον κωδικό επιβεβαίωσης έτσι ώστε να γίνει η σωστή εισαγωγή μας στο σύστημα. Για την περαιτέρω πλοήγηση μας στο πρόγραμμα χρειάζονται μερικές στοιχειώδεις γνώσεις για τον τρόπο χρήσης του internet.

Τίτλος Μαθήματος	Δηλωθέν Εξάμηνο
Ηλεκτρονικά Στοιχεία	2007-2008 Εαρινό
Ηλεκτρονικά Στοιχεία	2007-2008 Εαρινό
Ηλεκτρικά Κυκλώματα Ι	2007-2008 Εαρινό
Έξοδος	<input type="button" value="Έξοδος"/>

Σχήμα 4.1.9 Εικόνα από την αναλυτική βαθμολογία του εκάστοτε σπουδαστή

4.2 Οδηγίες Χρήσεως για τον διαχειριστή της εφαρμογής

Καλό θα ήτανε ο διαχειριστής του συστήματος να έχει την δυνατότητα να πραγματοποιεί όλες τις απαιτούμενες εργασίες διαχείρισης και συντήρησης της εφαρμογής μέσω του Διαδικτύου αλλά στην παρούσα φάση αυτό δεν είναι ακόμα εφικτό. Ούτως η άλλως το κύριο μέλημα για την εκπόνηση αυτής της πτυχιακής είναι ένας εύκολος τρόπος για την ανανέωση της εγγραφής των σπουδαστών από οποιαδήποτε σημείο και αν βρίσκονται χωρίς να είναι απαραίτητη η παρουσία τους στο τμήμα της γραμματείας της εκάστοτε σχολής εκτός βέβαια από την πρώτη εγγραφή όπου η παρουσία είναι απαραίτητη και υποχρεωτική. Άρα με όσα αναφέραμε παραπάνω όλη η διαδικασία της διαχείρισης της εφαρμογής γίνεται από τον κεντρικό ηλεκτρονικό υπολογιστή όπου έχει γίνει εγκατάσταση της εφαρμογής μας (server). Επίσης ολόκληρη η διαχείριση της εφαρμογής γίνεται μέσα από την βάση δεδομένων και δεν είναι απαραίτητη η παρεμβολή οποιουδήποτε στις σελίδες όπου είναι σχεδιασμένες με την τεχνολογία jsp. Η βάση δεδομένων είναι εκείνη που περιέχει όλες τις απαραίτητες πληροφορίες για τους σπουδαστές (προσωπικά στοιχεία, στοιχεία βαθμολογίας, δηλωμένα μαθήματα κ.α.) άρα οι απαιτούμενες γνώσεις που πρέπει να έχει κάποιος όπου θέλει να κάνει κάποιες αναπροσαρμογές στο πρόγραμμα θα πρέπει να είναι τέτοιες έτσι ώστε μπορεί να διαχειριστεί μια βάση δεδομένων χωρίς προβλήματα.

4.2.1 Εισαγωγή ενός νέου σπουδαστή

Ένας σπουδαστής για να μπορέσει να κάνει χρήση αυτού του προγράμματος θα πρέπει να έχει εγγραφεί στο σύστημα. Η εγγραφή του θα γίνεται με μία χειρόγραφη αίτηση όπου θα πρέπει να συμπληρώσει στην γραμματεία και μετά θα παραλαμβάνει το όνομα χρήστη και κωδικό επιβεβαίωσης. Η γραμματεία με την σειρά της θα πρέπει να καταχωρεί τα στοιχεία του σπουδαστή στην βάση δεδομένων. Μετά το πέρας αυτής της εργασίας ο χρήστης του προγράμματος θα μπορεί να κάνει οποιαδήποτε δυνατή εργασία αυτός επιθυμεί. Η εισαγωγή του σπουδαστή γίνεται μέσα στη βάση δεδομένων με όνομα 02_3_3-logoplhrof-DB.mdb, όπου θα την βρούμε μέσα στον πίνακα root του tomcat. Το επόμενο βήμα που κάνουμε είναι να ανοίξουμε τον πίνακα με τον τίτλο Students. Μέσα σε αυτόν το πίνακα θα βρούμε τα προσωπικά στοιχεία των σπουδαστών όπου είναι χρήστες αυτής της εφαρμογής (σχήμα 4.2.1).

am	surname	name	fathersname	birth_date	adt	ADT_date	author	entry_semester	username	password	declaration_date
2672	Γουγαύλη	Μαρία	Κωνσταντίνος	13/8 /1981	P866719	15/6 /1996	AT Ν. Σάντας	2002-2003 Β' Εαρινή	gougou	240782	1/1/1990
2673	Ατσάλη	Μαρίνα	Παναγιώτης	15/10/1982	T329604	31/3 /1999	ΥΑ Ηρακλείου	2004-2005 Β' Εαρινή	atsali	240782	1/1/1990
2693	Φρυσουλάκη	Σοφία	Νικόλαος	21/5 /1982	Σ868193	1 /9 /1999	AT Αγ. Μύρωνας	2002-2003 Β' Χειμερινή	free	123456	1/1/1990
2699	Ηλίας	Μιχάλης χαράλαμπος	Γεώργιος	24/7 /1982	AE935743	25/8 /2006	AT Περιφέρειας	2005-2006 Α' Χειμερινή	mikalixh	240782	1/1/1990
2700	Παπαδόπουλος	Νίκος	Πέτρος	2 /5 /1985	AE342134	3 /4 /2004	AT Μυτιλήνης	2003-2004 Β' Χειμερινή	pap	123456	1/1/1990
2713	Χασάπη	Αντωνία	Εμμανουήλ	1 /1/1981	Σ149669	14/6 /1996	AT Πεύκης	2002-2003 Β' Εαρινή	xas	222222	1/1/1990
2787	Ρονίρης	Εμμανουήλ	Ιωάννης	14/1 /1983	Σ450745	8 /5 /1997	AT Θήρας	2002-2003 Β' Εαρινή	ren	123456	1/1/1990

Σχήμα 4.2.1 Πλήρη εμφάνιση του πίνακα Students

Όταν εισάγουμε ένα σπουδαστή θα πρέπει να συμπληρώνουμε όλα τα πεδία υποχρεωτικά γιατί εάν δεν εισάγουμε όλα τα στοιχεία και ζητήσουμε κάποιες πληροφορίες τότε θα δεχόμαστε λάθος ή ελλιπή στοιχεία.

Σε αυτό τον πίνακα θα εισάγουμε ονοματεπώνυμο, όνομα πατρός, αριθμό δελτίου ταυτότητας, αρχή έκδοσης, εξάμηνο εγγραφής, username, password και πρόγραμμα σπουδών. Αφού γίνει αυτή η διαδικασία θα πρέπει να κατευθυνθούμε στον πίνακα με τίτλο grades. Μέσα στον παραπάνω πίνακα έχουμε τις συγκεντρωτικές βαθμολογίες σε όλα τα μαθήματα για όλους τους σπουδαστές (σχήμα 4.2.2). Μετά την εισαγωγή των στοιχείων μας στον πίνακα όπου αναφέρεται στα προσωπικά δεδομένα θα πρέπει να γίνει και η εγγραφή μας και σε αυτό τον πίνακα. Η εγγραφή σε όλους τους πίνακες γίνεται μία φορά για κάθε σπουδαστή, δηλαδή μία φορά θα γίνει η εισαγωγή των στοιχείων μας καθώς μια φορά πρέπει να γίνει και η εγγραφή μας στον πίνακα όπου αναφέρεται στις βαθμολογίες. Μέσα στον πίνακα των βαθμολογιών θα δούμε ότι υπάρχουν τέσσερις στήλες. Αυτές οι τέσσερις στήλες ονομαστικά είναι:

- 1 → am
- 2 → leason
- 3 → grade
- 4 → period

Έτσι μέσα σε αυτό τον πίνακα θα πάρουμε πληροφορίες όπου αναφέρονται στον αριθμό μητρώου του σπουδαστή, στα μαθήματα, στην βαθμολογία και σε ποια περίοδο ο σπουδαστής πήρε αυτή την βαθμολογία. Μέσα στον πίνακα θα δούμε κάποιες καταχωρίσεις με κωδικούς, και σε αυτό τον πίνακα χρησιμοποιήσαμε την μέθοδο με τους κωδικούς και τους εξαρτημένους πίνακες που περιγράψαμε παραπάνω. Έτσι λοιπόν εάν ο διαχειριστής θέλει να περάσει κάποιο νέο σπουδαστή θα πρέπει πρώτον να περάσει τα πλήρη στοιχεία του στον πίνακα Students και μετά να περάσει τα απαιτούμενα στοιχεία στον παραπάνω πίνακα. Ένα βασικό μειονέκτημα

στον συγκεκριμένο πίνακα είναι ότι όλα τα στοιχεία θα πρέπει να καταχωρηθούν με την κωδική τους μορφή. Τους κωδικούς και τις απαιτούμενες αντιστοιχίες θα τις πάρουμε από τους άλλους πίνακες της βάσης δεδομένων μας.

am	leason	grade	period
2542	103	5	7
2542	104	6	9
2542	105	6	9
2542	106	5	5
2542	107	5	7
2542	108	5	7
2542	109	6,8	5
2542	110	5	5
2542	111	5	5
2542	112	5	5
2542	113	9	7
2542	114	5,5	9
2542	115	6,5	9
2542	116	8	13
2542	117	6	9
2542	118	9,5	9
2542	119	5	11
2542	120	7	17
2542	121	5	7
2542	122	7	9
2542	123	5	7
2542	124	6	9
2542	125	8,5	9
2542	126	8	15
2542	127	7,1	13
2542	128	6	11
2542	129	7,5	11
2542	130	5	15
2542	131	8	9
2542	132	5	11
2542	133	7,5	22
2542	134	6,1	22
2542	135	6,5	11
2542	136	8,5	11

Σχήμα 4.2.2 Πλήρη εμφάνιση του πίνακα grades

Επίσης κατά την πρώτη εγγραφή θα πρέπει να περαστούν για κάθε σπουδαστή όλα τα μαθήματα με αρχική βαθμολογία μηδέν και σαν εξάμηνο καταχώρισης βαθμολογίας το εξάμηνο εγγραφής. Για να γίνει πλήρης κατανοητό αυτό που αναφέραμε παραπάνω ένας νέος χρήστης του προγράμματος καθώς κάνει την εισαγωγή του στην εφαρμογή θα πρέπει εάν επιλέξει την εφαρμογή αναλυτική βαθμολογία θα πρέπει να βλέπει ένα πίνακα με όλα τα μαθήματα με βαθμό μηδέν (0,0) και σαν εξάμηνο όπου καταχωρήθηκε η βαθμολογία να είναι όμοιο με το εξάμηνο όπου έκανε την πρώτη εγγραφή του στη σχολή του (σχήμα 4.2.3). Εάν μετά την πρώτη εισαγωγή του στο σύστημα δεν παίρνει αυτή την αναμενομένη εικόνα θα πρέπει να επικοινωνήσει άμεσα με τον εκάστοτε διαχειριστή για να γίνει διόρθωση των στοιχείων του. Αυτό το κομμάτι είναι το πιο δύσκολο και το πιο χρονοβόρο για την εισαγωγή ενός νέου σπουδαστή.

Τίτλος Μαθήματος	Τύπος Μαθήματος	Βαθμός	Περίοδος Εξέτασης
Ηλεκτρονική II	Εργαστήριο	.00	2005-2006 Α' Εαρινή
Μαθηματικά με Η/Υ I	Εργαστήριο	.00	2005-2006 Α' Εαρινή
Προγραμματισμός Η/Υ	Εργαστήριο	.00	2005-2006 Α' Εαρινή
Ηλεκτρονική II	Θεωρία	.00	2005-2006 Α' Εαρινή
Ηλεκτρικές & Ηλεκτρονικές Μετρήσεις	Θεωρία	.00	2005-2006 Α' Εαρινή
CAD & Κατασκευή II	Θεωρία	.00	2005-2006 Α' Εαρινή
Εφαρμ. Ηλεκτρομαγνητισμού	Θεωρία	.00	2005-2006 Α' Εαρινή
Ηλεκτρονικά Στοιχεία	Θεωρία	.00	2005-2006 Α' Εαρινή
Αγγλικά III	Θεωρία	.00	2005-2006 Α' Εαρινή
Ηλεκτρονική I	Εργαστήριο	.00	2005-2006 Α' Εαρινή
Ηλεκτρικές & Ηλεκτρονικές Μετρήσεις	Εργαστήριο	.00	2005-2006 Α' Εαρινή
CAD & Κατασκευή II	Εργαστήριο	.00	2005-2006 Α' Εαρινή
Εφαρμ. Ηλεκτρομαγνητισμού	Εργαστήριο	.00	2005-2006 Α' Εαρινή
Μαθηματικά με Η/Υ II	Εργαστήριο	.00	2005-2006 Α' Εαρινή
Ηλεκτρονική III	Θεωρία	.00	2005-2006 Α' Εαρινή
Ψηφιακά Κυκλώματα II	Θεωρία	.00	2005-2006 Α' Εαρινή
Μικροπολογιστές	Θεωρία	.00	2005-2006 Α' Εαρινή
Μαθηματικά με Η/Υ II	Θεωρία	.00	2005-2006 Α' Εαρινή
Ηλεκτρονική I	Θεωρία	.00	2005-2006 Α' Εαρινή
Ηλεκτρικά Κυκλώματα I	Θεωρία	.00	2005-2006 Α' Εαρινή
Φυσική I	Θεωρία	.00	2005-2006 Α' Εαρινή
Μαθηματικά	Θεωρία	.00	2005-2006 Α' Εαρινή
Οικονομία/Επιχειρήσεων	Θεωρία	.00	2005-2006 Α' Εαρινή

Σχήμα 4.2.3 Εικόνα μετά την πρώτη εισαγωγή μας στην εφαρμογή.

Με αυτές τις δύο διαδικασίες για κάθε σπουδαστή θα μπορούσαμε να πούμε ότι γίνεται η εγγραφή ενός νέου χρήστη για την συγκεκριμένη εφαρμογή. Σε αυτό το σημείο μπορούμε να τονίσουμε την δυνατότητα της ολικής διαχείρισης των δεδομένων μέσω διαδικτύου και είναι ένα σημείο όπου αναφέρεται στις δυνατότητες επεκτασιμότητας της όλης εφαρμογής.

4.2.2 Ενεργοποίηση και απενεργοποίηση της εφαρμογής όπου αναφέρεται στην διαδικασία της ανανέωσης της εγγραφής

Κατά την έναρξη κάθε εξαμήνου θα πρέπει να ενεργοποιείται η επιλογή της ανανέωσης της εγγραφής έτσι ώστε κάθε σπουδαστής να μπορεί να κάνει την συγκεκριμένη διαδικασία κάποια συγκεκριμένη χρονική στιγμή και όχι καθ' όλη την διάρκεια του ακαδημαϊκού εξαμήνου. Η συγκεκριμένη εφαρμογή είναι η μοναδική σε όλη την εφαρμογή όπου πρέπει ο χρήστης να επέμβει σε ένα συγκεκριμένο κομμάτι κώδικα και να τον διαφοροποιήσει. Συγκεκριμένα θα πρέπει να γίνει σχόλιο το ενεργό button όπου αναφέρεται στην ανανέωση της εγγραφής. Η σελίδα όπου πρέπει να διαφοροποιηθεί θα την βρούμε στον υπο-φάκελο με όνομα work. Αυτός ο υπο-φάκελος βρίσκεται μέσα στον κεντρικό φάκελο root του προγράμματος tomcat, όπου τον χρησιμοποιούμε για κεντρικό εξυπηρετητή (server). Αφού μεταβούμε σε αυτόν τον υπο-φάκελο θα επιλέξουμε και θα ανοίξουμε το αρχείο με όνομα central.jsp. Το άνοιγμα και η επεξεργασία αυτού του αρχείου μπορεί να γίνει με οποιοδήποτε πρόγραμμα το οποίο μπορεί να κάνει αποκωδικοποίηση jsp σελίδων πχ textpad. Η εισαγωγή σχολίων σε σελίδες οι οποίες αναφέρονται σε java γίνεται με την τοποθέτηση δύο πλάγιων γραμμών στο μπροστινό κομμάτι όπου θέλουμε να μην εκτελείται. Στα σχήματα 4.2.4 και 4.2.5 που υπάρχουν παρακάτω θα δούμε τον κώδικά μας με ενεργή και ανενεργή την συγκεκριμένη εφαρμογή.

```
// ενεργό κουμπί που αναφέρεται στην ανανέωση της εγγραφής

    out.println("<table width=330 height=30 border=1 >");
        out.println("<tr>");
            out.println("<td                                height=43><div
align=left><strong>Δήλωσεις Μαθημάτων :</strong></div></td>");
            out.println("<form method=get action=dhlwsh.jsp>");
            out.println("<td>");
            out.println("<div align=center>");
            out.println("<input type=submit name=am1 value="+am1+"
>");
            out.println("</div></td>");
            out.println("</form>");
            out.println("</tr>");
            out.println("</table>");
// τέλος επιλογής αννανέωσης
```

Σχήμα 4.2.4 Εμφάνιση κωδικού με ενεργή την εφαρμογή ανανέωσης εγγραφής

```
// ενεργό κουμπί που αναφέρεται στην ανανέωση της εγγραφής
//out.println("<table width=330 height=30 border=1 >");
//out.println("<tr>");
//out.println("<td height=43><div align=left><strong>Δήλωσεις
Μαθημάτων :</strong></div></td>");
//out.println("<form method=get action=dhlwsh.jsp>");
//out.println("<td>");
//out.println("<div align=center>");
//out.println("<input type=submit name=am1 value="+am1+" >");
//out.println("</div></td>");
//out.println("</form>");
//out.println("</tr>");
//out.println("</table>");
// τέλος επιλογής ανανέωσης
```

Σχήμα 4.2.5 Εμφάνιση κωδικού με ανενεργή την εφαρμογή ανανέωσης εγγραφής

Όπως παρατηρούμε και από τις παραπάνω εικόνες η μοναδική διαφορά και παράλληλα η μοναδική προσθήκη όπου κάνουμε στον κύριο κορμό του προγράμματος είναι να τοποθετήσουμε δύο κάθετες γραμμές << // >> από το σημείο που λέει << // ενεργό κουμπί που αναφέρεται στην ανανέωση της εγγραφής >> έως και το σημείο που λέει << // τέλος επιλογής ανανέωσης >>.

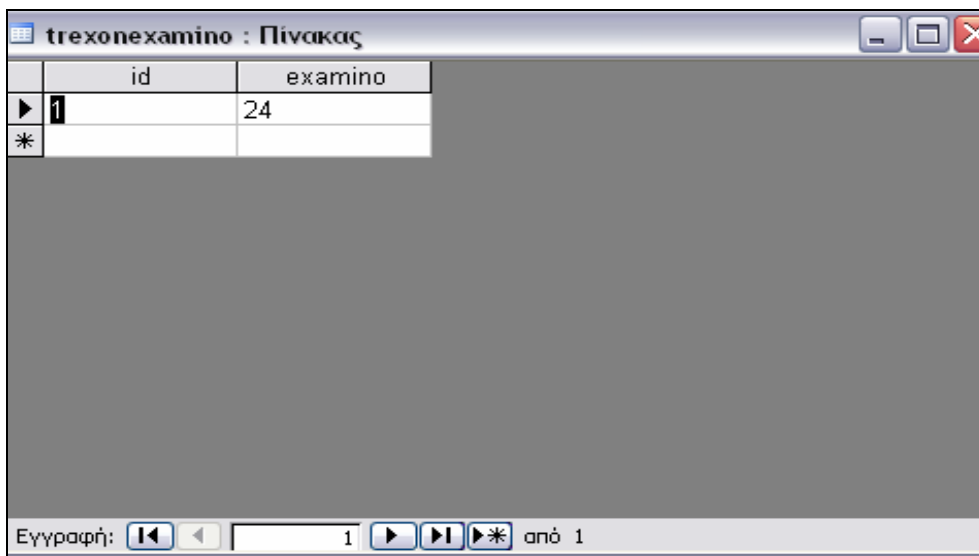
4.2.3 Εισαγωγή εξαμήνου εγγραφής

Όπως αναφέραμε παραπάνω κατά την διαδικασία της ανανέωσης εγγραφής ένας σπουδαστή δηλώνει τα μαθήματα όπου επιθυμεί να παρακολουθήσει το τρέχον ακαδημαϊκό έτος και μέσα στη βάση δεδομένων καταχωρούνται εκτός από τα δηλωθέντα μαθήματα και ο αριθμός μητρώου του σπουδαστή

αλλά και το τρέχον εξάμηνο εγγραφής. Για το εξάμηνο εγγραφής έχουμε δύο γενικές επιλογές. Αυτές οι δύο είναι:

- α → Χειμερινό εξάμηνο και
- β → Εαρινό εξάμηνο.

Για να μπορέσουμε λοιπόν να περάσουμε αυτά τα στοιχεία θα πρέπει ο διαχειριστής της εφαρμογής να ορίσει το τρέχον εξάμηνο. Η διαδικασία ορισμού είναι πολύ απλή και γίνεται μέσα από την βάση δεδομένων. Συγκεκριμένα ανοίγουμε τον πίνακα με όνομα `trexonexamino` (σχήμα 4.2.4) και καταχωρούμε τον κωδικό όπου αντιστοιχεί στο τρέχον εξάμηνο. Τους κωδικούς αυτούς θα τους δούμε από τον πίνακα με όνομα `tblExaminoEggraphhs`.



	id	examino
▶	1	24
*		

Εγγραφή: 1 από 1

Σχήμα 4.2.4 Εμφάνιση πίνακα `trexonexamino`

Η διαδικασία της αλλαγής πρέπει να γίνεται κάθε φορά όπου ενεργοποιείται η επιλογή της ανανέωσης εγγραφής δηλαδή μια φορά το εξάμηνο.

5 Συμπεράσματα

Βασικός σκοπός αυτής της πτυχιακής ήταν να κατανοήσουμε πλήρως τους όρους της δυναμικής σελίδας, της ανταλλαγής δεδομένων και το σκεπτικό μια πλήρους βάσης δεδομένων. Χρησιμοποιήσαμε και μελετήσαμε διάφορες πλατφόρμες όπου σχετίζονται με αυτούς τους όρους. Αυτές οι εφαρμογές είναι:

- A → Jakarta Tomcat
- B → Dreamweaver
- Γ → ODBC
- Δ → Microsoft Access.

Οι εφαρμογές που χρησιμοποιήθηκαν ήταν όλες επαρκείς για την ολοκλήρωση αυτής της πτυχιακής εργασίας. Η επαφή μας με αυτά τα προγράμματα έγινε πρώτη φορά κατά την έναρξη της μελέτης μας για την περάτωση αυτής της εργασίας. Σίγουρα η γνώση μας πάνω σε αυτήν την μορφή της τεχνολογίας έχει αναπτυχθεί με την βοήθεια της εφαρμογής που δημιουργήθηκε. Κατανοήσαμε πλήρως τον τρόπο όπου τα δεδομένα «ταξιδεύουν» μέσα στο διαδίκτυο και περνάν από σελίδα σε σελίδα μέχρι να φτάσουν στον τελικό προορισμό τους.

Μία εφαρμογή ακόμη όπου μελετήσαμε είναι η διαδικασία ενεργοποίησης ενός εξυπηρετητή και συγκεκριμένα του Jakarta Tomcat. Ο συγκεκριμένος εξυπηρετητής είναι ένα ελεύθερο εξυπηρετητής όπου έχει δημιουργηθεί από την εταιρία Java Sun. Παρά τα αρκετά μειονεκτήματα που έχει εμφανίσει, σε γενικές γραμμές θεωρείται ένα αξιόλογο πρόγραμμα που έχει την δυνατότητα να δημοσιεύει τα δεδομένα που επιθυμούμε στο Διαδίκτυο.

Η τρίτη εφαρμογή που μελετήσαμε είναι η διαδικασία ανταλλαγής δεδομένων μέσω της εφαρμογής της ODBC. Η ODBC είναι μια εφαρμογή που υποστηρίζεται από την Microsoft και παρέχεται δωρεάν με την εγκατάσταση των Windows. Με αυτή την εφαρμογή δημοσιεύουμε την βάση δεδομένων που επιθυμούμε στο Διαδίκτυο για να μπορούμε να αντλούμε ή να καταχωρούμε δεδομένα από απόσταση. Η εγκατάσταση της βάσης και η δημοσίευση της γίνεται με εύκολο τρόπο.

Στην παρούσα εργασία κύριο μέλημά μας ήταν πρώτον να κατανοήσουμε και να πειραματιστούμε πάνω σε αυτές τις τεχνολογίες. Η πολυπλοκότητα της κατανόησης της δυναμικής σελίδας είναι κάτι απολύτως κατανοητό, μιας και δεν είναι ευρέως γνωστή τεχνολογία δημοσίευσης ιστοσελίδων. Στις μέρες μας χρησιμοποιείται κατά κόρον το μοντέλο του πελάτη και του εξυπηρετητή, όπου ένας περιφερειακός ηλεκτρονικός υπολογιστής ζητάει κάποιες πληροφορίες από ένα κεντρικό σύστημα και αυτό με την σειρά του, του της παρέχει. Ο χρήστης δέχεται παθητικά τα δεδομένα, χωρίς να μπορεί να τα μεταβάλει. Στις δυναμικές σελίδες δεν ισχύει αυτό. Ο χρήστης μπορεί να τροποποιήσει τα δεδομένα με την έγκριση φυσικά και του διαχειριστή π.χ. για την παραπάνω εφαρμογή που μόλις αναλύσαμε ο

εκάστοτε χρήστης μπορεί να μεταβάλει τα δεδομένα του, να τα επεξεργαστεί και μετά να τα ελέγξει.

Όπως είναι κατανοητό για την δημιουργία ενός τέτοιου πηγαίου κώδικα χρειάστηκε αρκετή μελέτη η οποία αναφερόταν σε αυτού του είδους την τεχνολογία. Μετά από ένα χρόνο μελέτης και πειραματισμών σε αυτού του είδους την τεχνολογία πιστεύω ότι είμαι σε θέση να εκφέρω μια προσωπική άποψη η οποία θα αναφέρεται στην τεχνολογία της ανταλλαγής δεδομένων μέσω διαδικτύου και την δημιουργία δυναμικών σελίδων. Πιστεύω πως αυτή η τεχνολογία μπορεί να αναπτυχθεί και να προσφέρει πολλά περισσότερα στον κόσμο της τεχνολογίας και στην ανταλλαγή των δεδομένων. Πολλές μεγάλες εταιρίες αρχίζουν και εμπιστεύονται αυτού του είδους τις τεχνολογίες χωρίς να έχουν προβλήματα, επιπλέον μπορούν και παρέχουν στους πελάτες τους υπηρεσίες που πραγματικά τους «λύνουν τα χέρια». Η εφαρμογή που είδαμε παραπάνω είναι μια εφαρμογή που θα μας βοηθήσει να κατανοήσουμε πλήρως αυτού του είδους την τεχνολογία. Βέβαια, επειδή είναι η πρώτη μου προσπάθεια και ο πειραματισμός μου έχει πολλές ελλείψεις και πολλά μειονεκτήματα.

Η πιο μεγάλη έλλειψη πιστεύω ότι είναι η απουσία της διαδικτυακής διαχείρισης της εφαρμογής μας. Μία μελλοντική επανέκδοση του προγράμματος σίγουρα θα περιέχει και αυτή την δυνατότητα που τώρα στερείται αυτή η εφαρμογή. Άλλο ένα μειονέκτημα είναι και ο παράγοντας της ασφάλειας, όπου δεν έχει δημιουργηθεί ένας αλγόριθμος όπου θα προστάτευε τα δεδομένα, καθώς θα μεταβαίνουν από σελίδα σε σελίδα και θα καταλήγουν στη βάση δεδομένων μας. Παρόλα αυτά ο συνολικός κώδικας περιέχει πολλές τεχνοτροπίες οι οποίες είναι αρκετά καλές και άξια μπορούν να συναγωνιστούν με μεγάλες και αναγνωρισμένες εφαρμογές οι οποίες στηρίζονται στην ίδια τεχνολογία και στο ίδιο σκεπτικό.

Οι σελίδες jsp, γενικότερα από μόνες τους παρέχουν ένα είδος ασφάλειας αφού ο κάθε χρήστης δεν μπορεί να δει τον κώδικα από όπου προέρχονται. Αυτό δεν συμβαίνει μόνο στην συγκεκριμένη εφαρμογή αλλά όλο το σκεπτικό των jsp, είναι να δημιουργείται στον κεντρικό υπολογιστή και ο εκάστοτε χρήστης λαμβάνει και βλέπει μόνο τον κώδικα ο οποίος έχει την μορφή HTML σελίδας.

Τελειώνοντας και παρόλα τα μειονεκτήματα και τα πλεονεκτήματα όπου παρέχει η συγκεκριμένη εφαρμογή θα πρέπει να τονιστεί για μία ακόμη φορά ότι το συγκεκριμένο πρόγραμμα δημιουργήθηκε όχι για να αντικαταστήσει την γραμματειακή υποστήριξη του ΤΕΙ ούτε να την υποβιβάσει. Σκοπός μας είναι να την βοηθήσει και να γίνει ένα καλό εργαλείο στο προσωπικό του τμήματος, έτσι ώστε να ξεφύγει από τυπικές διαδικασίες γραφειοκρατίας και να μπορεί να ασχοληθεί αμέριστα στην ανάπτυξη της ήδη άριστης γραμματειακής υποστήριξης του ΤΕΙ.

Βιβλιογραφία

- 1: Teach yourself ODBC in 21 days / Bill Whiting, Bryan Morgan
Indianapolis, IN :Sams Pub., c1996
- 2: Teach yourself SQL in 14 days / Bryan Morgan and Jeff Perkins
Indianapolis, Ind. :Sams Pub., c1995
- 3: Professional JSP / Simon Brown
Birmingham, UK :, Chicago :Wrox Press, c2001
- 4: Beginning JSP Web development / Jayson Falkner
Birmingham, UK :Wrox Press, c2001
- 5: JSP : the complete reference / Phil Hanna
New York:Osborne/McGraw-Hill, c2001
- 6: Teach yourself Web publishing with HTML in a week / Laura Lemay
Indianapolis, Ind. :Sams Publishing, c1995
- 7: HTML / Άγγελος Γαλλής
Θεσσαλονίκη :Anubis, 1995
- 8: Προγραμματίστε σε HTML / Άγγελος Γαλλής
Αθήνα :Anubis, 1996
- 9: Special edition using HTML / written by Tom Savola with
AlanWestenbroek, Joseph Heck
Indianapolis, IN :Que, c1995
- 10: Ο οδηγός της Microsoft για τη Microsoft Access 2000 / John Viescas
Αθήνα :Κλειδάριθμος Microsoft Press, 1999
- 11: Microsoft Office XP step by step / Perspection, Inc., Online Training
Solutions, Inc., Curtis Frye and Kristen Crupi
Redmond, Wash. :Microsoft Press, c2001
- 12: Tomcat : kick start / Martin Bond, Debbie Law
Indianapolis, Ind. :Sams, c2003
- 13: Definitive VoiceXML/Adam Hocek, David Cuddihy
USA, Pearson Education Inc, c2003
- 14: Java Servlet Programming/ Jason Hunter
USA, O' Reilly & Associates Inc, c1998
- 15: Professional JSP, 2nd Edition
Wrox Press LTD, UK Birmingham, c2002

Ιστοσελίδες

- 1: <http://www.roseindia.net/jsp/Accessingdatabase-fromJSP.shtml>
- 2: <http://www.oracle.com>
- 3: <http://www.javaworld.com/javaworld/jw-11-2001/jw-1130-jsp.html>
- 4: <http://webgrp.ceid.upatras.gr/html.htm>
- 5: <http://www.eeei.gr/odhgos/htmllex.htm>
- 6: <http://java.sun.com/>
- 7: <http://www.java2s.com/Code/Java/JSP/JspFormCheckBoxDemo.htm>
- 8: <http://www.freestuff.gr/forums/viewtopic.php?t=26919>
- 9: <http://www.javalobby.org/>

- 10: http://www.stardeveloper.com/articles/index.html?category_id=211
- 11: <http://www.echoecho.com/htmlforms08.htm>
- 12: <http://www.java2s.com/Code/Java/JSP/CatalogJSP.htm>
- 13: <http://tomcat.apache.org/tomcat-5.0-doc/jasper-howto.html>
- 14: <http://dide.flo.sch.gr/Plinet/Tutorials/Tutorials-JSP-3-Commands.html>

Παράρτημα

Παρακάτω θα δούμε μερικά χαρακτηριστικά κομμάτια από τον κώδικά μας. Μαζί με των κωδικό θα υπάρχει και μία επεξήγηση για την ακριβή χρήση αυτού του κομματιού στην εφαρμογή μας.

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<title>Κεντρική Σελίδα</title>
<meta http-equiv="Content-Type" content="text/html;
charset=iso-8859-7">
</head>
<body>
<div align="center">
  <p><strong>ΤΕΙ ΚΡΗΤΗΣ</strong></p>
  <p><strong>ΠΑΡΑΡΤΗΜΑ ΧΑΝΙΩΝ</strong></p>
  <p><strong>ΕΦΑΡΜΟΓΗ ΔΗΛΩΣΗΣ ΜΑΘΗΜΑΤΩΝ
ΕΞ'ΑΠΟΣΤΑΣΕΩΣ</strong></p>
  <p>&nbsp;</p>
  <div align="center">
    <table width="370" border="1">
      <tr>
        <td width="137" height="33">Επιλέξτε Τμήμα :</td>
        <td width="217"><span class="tabletext">
          <SELECT name="tmhma" class="searchformarea" >
            <OPTION value="1"> Ηλεκτρονικής </OPTION>
            <OPTION value="2"> Φυσικών Πόρων </OPTION>
          </SELECT>
        </span></td>
      </tr>
      <tr>
        <td colspan="2"><form name="form1" method="post"
action="work/central.jsp" encoding="iso-8859-7">
          <td>Δώστε username :</td>
          <td><input type="text" name="user" ></td>
        </tr>
        <tr>
          <td colspan="2"><form name="form1" method="post"
action="work/central.jsp" encoding="iso-8859-7" >
            <td>Δώστε password :</td>
            <td><input type="password" name="pass"></td>
          </tr>
          <tr>
            <td colspan="2"><div align="center">
              <input type="submit" name="Submit" value="Είσοδος">
              <input type="reset" name="Submit2" value="Καθαρισμός">
            </div></td>
          </tr>
        </tr>
      </table>
    </div>
  </div>

```

```

</table>
<p>&nbsp;</p>
<p>&nbsp;</p>
<p>&nbsp;</p>
<p>&nbsp;</p>
<p>&nbsp;</p>
<p>&nbsp;</p>
<p>&nbsp;</p>
<div align="left">
<p><strong>Σπουδαστής: Μιχάλης-Χαράλαμπος Ηλίας </strong></p>
</body>
</html>

```

Η παραπάνω σελίδα είναι η αρχική σελίδα και η μοναδική σελίδα όπου είναι σχεδιασμένη κώδικα HTML.

```

<%
    {
request.setCharacterEncoding ("iso-8859-7");
    try {
        Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
    } catch (ClassNotFoundException e)
    {
        out.println("<h1>Driver not found:" + e.getMessage() +
"</h1>" );
    }
    try {
        Connection conn=DriverManager.getConnection
("jdbc:odbc:database");
        Statement stmt = conn.createStatement();
        Statement stmt2 = conn.createStatement();
        ResultSet rs;
        ResultSet rs2;
        if (request.getParameter("user") != null)
            if (request.getParameter("pass") != null)
                {
                    String T1 = request.getParameter("user");
                    String T2 = request.getParameter("pass");

rs = stmt.executeQuery("SELECT am, surname, name, fathersname,
adt, username, password FROM Students WHERE username= '"+T1+"'
AND password= '"+T2+"' ");
out.println("<table>");
        int aml=00000;
while (rs.next())    {
                String am = rs.getString("am");
                String surname = rs.getString("surname");
                String name = rs.getString("name");

```

```

        String fathersname =
rs.getString("fathersname");
        String adt = rs.getString("adt");
        out.println("<b>ΠΛΗΡΗ ΣΤΟΙΧΕΙΑ ΣΠΟΥΔΑΣΤΗ:</b>");
        out.println("<hr>");
        out.println("<tr><td><b>AM: </b></td>");
        out.println("<td>" +am+ " </td></tr>");
        out.println("<tr><td><b>ONOMA: </b></td>");
        out.println("<td>" +name+ " </td></tr>");
        out.println("<tr><td><b>ΕΠΙΘΕΤΟ:</b></td>");
        out.println("<td>" +surname+ " </td></tr>");
        out.println("<tr><td><b>ONOMA ΠΑΤΡΟΣ: </b></td>");
        out.println("<td>" +fathersname+ " </td></tr>");
        out.println("<tr><td><b>ΑΡΙΘΜΟΣ ΔΕΛΤΙΟΥ ΤΑΥΤΟΤΗΤΑΣ:
</b></td>");
        out.println("<td>" +adt+ " </td></tr>");
        out.println("</table>");
        am1= Integer.parseInt(am);
// ενεργό κουμπί που αναφέρεται στην ανανέωση της εγγραφής
        out.println("<table width=330 height=30 border=1 >");
        out.println("<tr>");
        out.println("<td height=43><div
align=left><strong>Δήλωσεις Μαθημάτων :</strong></div></td>");
        out.println("<form method=get action=dhlwsh.jsp>");
        out.println("<td>");
            out.println("<div align=center>");
            out.println("<input type=submit name=am1
value="+am1+ " >"); out.println("</div></td>");
        out.println("</form>");
        out.println("</tr>");
        out.println("</table>");
// τέλος επιλογής αννανέωσης

```

Παραπάνω βλέπουμε το πρόγραμμα με τα ενεργά πλήκτρα στην πρώτη σελίδα.

```

if (request.getParameter("am1") != null)
{
String T1 = request.getParameter("am1");
rs = stmt.executeQuery("SELECT am, surname, name, fathersname,
adt, username, password FROM Students WHERE am= "+T1+" ");
out.println("<b><h2> ΤΑ ΣΤΟΙΧΕΙΑ ΣΑΣ ΕΙΝΑΙ:</h2></b>");
out.println("<table>");
while (rs.next())
{
String am = rs.getString("am");
String surname = rs.getString("surname");
String name = rs.getString("name");
String fathersname = rs.getString("fathersname");
String adt = rs.getString("adt");
out.println("<tr><td><b>AM: </b></td>");
out.println("<td>" +am+ " </td></tr>");

```

```

out.println("<tr><td><b>ONOMA: </b></td>");
out.println("<td>" +name+ " </td></tr>");
out.println("<tr><td><b>ΕΠΙΘΕΤΟ:</b></td>");
out.println("<td>" +surname+ " </td></tr>");
out.println("<tr><td><b>ONOMA ΠΑΤΡΟΣ: </b></td>");
out.println("<td>" +fathersname+ " </td></tr>");
out.println("<tr><td><b>ΑΡΙΘΜΟΣ ΔΕΛΤΙΟΥ ΤΑΥΤΟΤΗΤΑΣ:
</b></td>");
out.println("<td>" +adt+ " </td></tr>");
out.println("<table width=500 height=30 border=0 >");
%>

```

Τα μαθήματα που έχειται δικαίωμα να δηλώσει είναι τα παρακάτω:

```

<% int ps =8888;
    int ts=0000;
    int pma=0000;
    int ar=0000;
    int gr=0000;
    int i=0;
    String ma8="asdfghjklkgj";
    String sem2="";
    String aal="";
    String code1="";
    String title1="";
    if ("am" != null)
    { rs2 = stmt2.executeQuery("SELECT * FROM AMdiax2 WHERE
AM <= "+am+ " ");
    while (rs2.next())
    {String AM1= rs2.getString ("AM");
    String AA1= rs2.getString("A/A");
    ps= Integer.parseInt(AA1);
    ts=(ps+1);
    }
}

```

Παραπάνω βλέπουμε τον τρόπο επιλογής του εξαμήνου εγγραφής καθώς και τον τρόπο όπου κάνουμε την εμφάνιση των μαθημάτων.

```

{
try {
Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
} catch (ClassNotFoundException e)
{
out.println("<h1>Driver not found:"+e +e.getMessage()
+ "</h1>" );
}
try {Connection conn=DriverManager.getConnection(
"jdbc:odbc:database");
Statement stmt = conn.createStatement();
Statement stmt2 = conn.createStatement();
ResultSet rs;
ResultSet rs2;
if (request.getParameter("am") != null)

```



```

        if (request.getParameter("pass") != null)
        if (request.getParameter("pass1") != null)
            if (request.getParameter("pass2") != null)
            {
                String T1 = request.getParameter("am");
                String T2 = request.getParameter("pass");
                String T3 = request.getParameter("pass1");
                String T4 = request.getParameter("pass2");
                String ps3= "asdfghjklj";
                String ps4= "zxcvbnmlqwer";
                ps3= (""+T3+"");
                ps4= (""+T4+"");
                rs = stmt.executeQuery("SELECT am, surname, name,
                fathersname, adt, username, password FROM Students WHERE
                am="+T1+" AND password= '"+T2+" '");
                out.println("<table>");
                while (rs.next())    {
                    String am = rs.getString("am");
                    String surname = rs.getString("surname");
                    out.println("<td>Κύριε/α: " +surname+ "
                </td></tr>");
                    String name = rs.getString("name");
                    String fathersname =
                rs.getString("fathersname");
                    String adt = rs.getString("adt");
                    if( T3.equals(T4) && ("am") != null)
                    { out.println("<td>Το Password σας άλλαξε με
                επιτυχία. Το νέο σας Password είναι:</td>");
                    out.println("<td>" +ps3+ " </td></tr>");
                    rs2 = stmt2.executeQuery("UPDATE Students SET password =
                '"+ps3+"' WHERE am="+am+"");
                    }
                    else
                    {
                        out.println("<td>Δώσατε διαφορετικά
                Passwords</td>");
                    }
                }
            }

```

Παραπάνω έχουμε την σύγκριση των κωδικών επιβεβαίωσης έτσι ώστε να κάνουμε αλλαγή του κωδικού μας.

```

if (request.getParameter("am1") != null)
{String t1 = request.getParameter("am1");
rs = stmt.executeQuery("SELECT * FROM grades WHERE am="+t1+"
ORDER BY period");
out.println("<left><table border=1>");
out.println("<tr><th>Τίτλος Μαθήματος </th><th>Τύπος
Μαθήματος</th><th>Βαθμός</th><th>Περίοδος
Εξέτασης</th></tr>");

```

Με αυτό το κομμάτι του κώδικα έχουμε την εμφάνιση της αναλυτικής βαθμολογίας του εκάστοτε σπουδαστή.

```
while (rs.next())    {String title = rs.getString("aa_course");
                    String semester = rs.getString("semester_id");
                    if ("title"!=null);
                    {rs3 = stmt3.executeQuery("SELECT
AA,mcode,mtitle,type_id FROM Mathimata WHERE AA="+title);
                    while (rs3.next())
                        {String mt = rs3.getString("mtitle");
                        out.println("<tr><td>" +mt+
"</td>");}
```

Ενώ στο παραπάνω κομμάτι βλέπουμε τα μαθήματα όπου έχει δηλώσει ο σπουδαστής.

Πίνακας Περιεχομένων

1. Εισαγωγή.....	1
1.1 Βασική Ιδέα για την δημιουργία του προγράμματος.....	1
1.2 Απαιτήσεις - Προδιαγραφές του συστήματος.....	2
1.3 Μια γρήγορη ματιά στην παρούσα εργασία.....	3
2. Εξοικείωση με τα αναπτυξιακά περιβάλλοντα.....	4
2.1 Γενικά.....	4
2.2 Macromedia Dreamweaver MX 2004	4
2.3 Εισαγωγή στον Jakarta Tomcat.....	7
2.4 Μοντέλο Πελάτη – Εξυπηρετητή.....	9
2.4 Σύστημα πηγών δεδομένων (ODBC).....	10
3. Σχεδίαση Εφαρμογής.....	16
3.1 Microsoft Access	16
3.2 Σχεδίαση Σελίδων JSP	24
4. Οδηγίες Χρήσεως της Εφαρμογής.....	30
4.1 Οδηγίες Χρήσεως για τον χρήστη της εφαρμογής.....	30
4.2 Οδηγίες Χρήσεως για τον διαχειριστή της εφαρμογής.....	35
4.2.1 Εισαγωγή ενός νέου σπουδαστή.....	35
4.2.2 Ενεργοποίηση και απενεργοποίηση της εφαρμογής όπου αναφέρεται στην διαδικασία της ανανέωσης της εγγραφής.....	38
4.2.3 Εισαγωγή εξαμήνου εγγραφής.....	39
5 Συμπεράσματα.....	41
Βιβλιογραφία.....	43
Ιστοσελίδες.....	43
Παράρτημα.....	45