Εργασία 3
Προγραμματισμός στο Διαδίκτυο και στον Παγκόσμιο Ιστό

Επιμέλεια: Αρμένη Αθηνά p20025

Αργύρα Ουρανία p20023

Αρχικά, γνωρίζουμε ότι θα πρέπει να έχουμε εγκατεστημένο το Eclipse(integrated development environment-ολοκληρωμένο περιβάλλον ανάπτυξης λογισμικού-εφαρμογή λογισμικού που παρέχει ολοκληρωμένες εγκαταστάσεις σε προγραμματιστές υπολογιστών για ανάπτυξη λογισμικού). Για να συνεχίσουμε με τη συγκεκριμένη εργασία, θα πρέπει να συμπεριλάβουμε και τις κλάσεις που δημιουργήσαμε στις δύο προηγούμενες εργασίες με κατάλληλες προσθήκες και τροποποιήσεις(ζητείται από την εργασία).

1ο βήμα : Επέκταση web project προηγούμενης άσκησης

·1.1

Μας ζητείται να επεκτείνουμε τη λειτουργικότητα άρα να προσθέσουμε περισσότερες λειτουργίες.

2ο βήμα : Ολοκλήρωση λειτουργιών όλων των κατηγοριών χρηστών (servlet)

·2.1

Σε αυτό το βήμα μας ζητείται να υλοποιήσουμε μηχανισμό login – password για κάθε κατηγορία χρηστών. Τη συγκεκριμένη λειτουργία την υλοποιούμε με δύο κλάσεις οι οποίες βρίσκονται στο Dao – System Dao. Η μία είναι υπολοιημένη στο bytesToHex όπου κάνει κρυπτογράφηση του κωδικού για να μην φαίνεται στη βάση μας ο πραγματικός κωδικός (να φαίνεται ο κρυπτογραφημένος δηλαδη όπως επίσης γίνεται και η αποκρυπτογράφησή του).Το ίδιο γίνεται και στην επόμενη κλάση getAlphaNumericString. Στο gunet υπήρχαν οι συγκεκριμένες κλάσεις ως βοηθητικές.

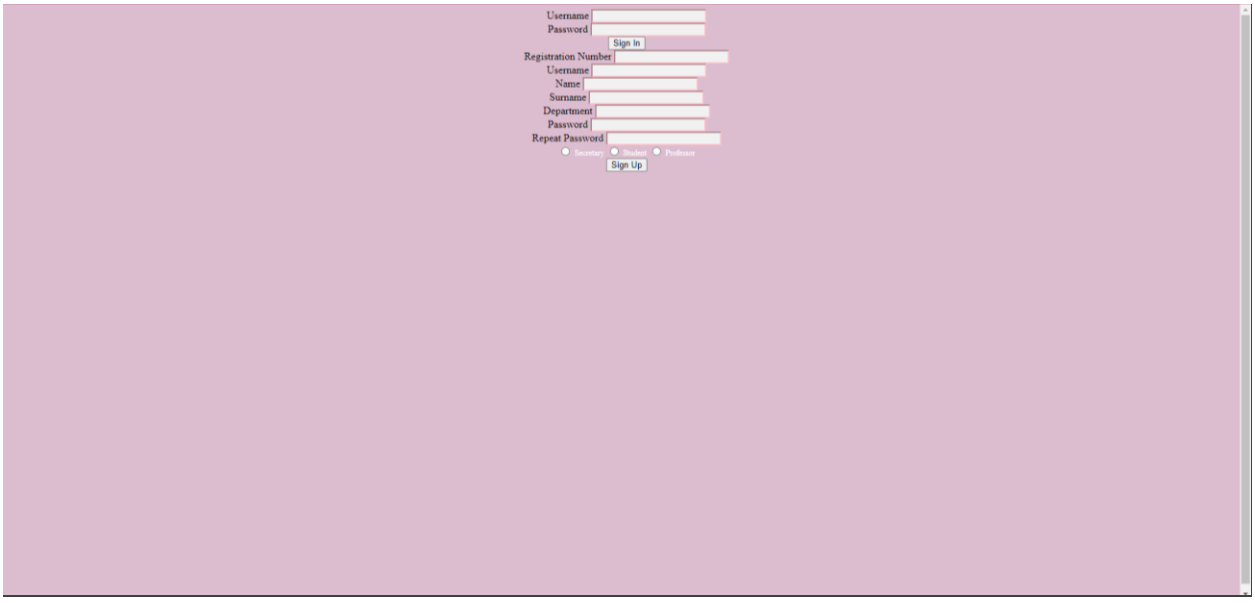
·2.2

Το βήμα 2.2 το υλοποιούμε με html για να δημιουργήσουμε το κεντρικό μενού, δηλαδή την αρχική σελίδα για όλους τους χρήστες. Ως τίτλο βάζουμε το Login Page. Στη συνέχεια, προσθέτουμε την ετικέτα link όπου ορίζει τη σχέση μεταξύ του τρέχοντος εγγράφου και ενός εξωτερικού πόρου. Με το

rel=”stylesheet” καθορίζεται η σχέση μεταξύ του τρέχοντος εγγράφου και του συνδεδεμένου εγγράφου. Με το href δηλώνουμε το url του λινκ και στη συνέχεια δηλώσαμε τον τύπο(css). Η ετικέτα script στη συνέχεια χρησιμοποιήθηκε για να οδηγήσει σε ένα εξωτερικό αρχείο σεναρίου μέσω του χαρακτηριστικού src. Προσθέτουμε μετά και τον τύπο(javascript). Με το request.getAttribute προσθέτουμε το αντικείμενο User. Προσθέτουμε το χαρακτηριστικό στο αίτημα και προωθούμε το αίτημα σε άλλο πόρο. Επομένως, όλος ο κώδικας που χειρίζεται αυτό είναι σε JSP. Αν το msg είναι κάτι από τις 3 επιλογές που έχουμε επιλέξει (δηλαδή είτε ο χρήστης δεν υπάρχει, είτε υπάρχει ήδη, είτε ο κωδικός είναι λανθασμένος) τότε βάζουμε το Swal fire έτσι ώστε να ανοίξει το αναδυόμενο παράθυρο και να μας εμφανίσει ότι έγινε λάθος και τι λάθος έγινε. Η μέθοδος Response.setHeader στη συνέχεια χρησιμοποιήθηκε ως μία ενσωματωμένη διεπαφή προγραμματισμού της ενότητας Http η ποία ορίζει μία τιμή κεφαλίδας σε κάποιες άλλες (implicit). Εάν αυτή η κεφαλίδα υπάρχει ήδη στις προς αποστολή κεφαλίδες, η τιμή της θα αντικατασταθεί. Άρα στη συνέχεια χρησιμοποιούμε μια σειρά από συμβολοσειρές (Cache-control). Οι πολλές κεφαλίδες θα σταλθούν με το ίδιο όνομα και οι τιμές που δεν είναι συμβολοσειρές θα αποθηκευτούν χωρις τροποποίηση. Για λόγους ασφαλείας, αν η χρονική περίοδος που είμαστε συνδεδεμένοι μέσα σε ένα λογαριασμό λήξει, τότε μας εμφανίζεται μήνυμα ότι έχουμε αποσυνδεθεί.

Στη συνέχεια, δημιουργούμε μία φόρμα και βάζουμε κάποια χαρακτηριστικά όπως όνομα, λειτουργία και μέθοδο(καθορίζει τον τρόπο αποστολής δεδομένων φόρμας τα οποία αποστέλλονται στη σελίδα που καθορίζεται το χαρακτηριστικό action). Μετά προσθέτουμε label(ετικέτα που χρησιμοποιείται για τον καθορισμό μιας ετικέτας για ένα στοιχείο input μίας φόρμας),input (εισαγωγή από το χρήστη) και αυτό το κάνουμε για το username του χρήστη και τον κωδικό του. Στη συνέχεια έχει προστεθέι ένα κουμπί για το sign in.

Τέλος, δημιουργούμε μία άλλη φόρμα για την εγγραφή ενός χρήστη, οπότε κάνουμε σχεδόν τα ίδια όπως και στην προηγούμενη φόρμα.



·2.3

·Τι είναι ο servlet και ποια η χρήση του;

Οι servlets Java είναι κλάσεις Java που έχουν σχεδιαστεί να ανταποκρίνονται σε αιτήματα HTTP στο πλαίσιο μίας εφαρμογής Ιστού. Μας βοηθάνε να έχουμε μία καλύτερη απόδοση γιατί δημιουργείται ένα νήμα(ένα μικρό σύνολο εντολών που έχουν σχεδιαστεί να εκτελούνται και να προγραμματίζονται από τη CPU, ανεξάρτητα από τη γονική διαδικασία,πχ ενώ το πρόγραμμα μπορεί να εκτελει άλλες εργασίες το νήμα μπορεί να εκτελεί ξεχωριστή εργασία ή να περιμένει να συμβεί ένα συγκεκριμένο συμβάν) για κάθε αίτημα και όχι για κάθε διαδικασία. Μας παρέχει φορητότητα εφόσον χρησιμοποιεί γλώσσα Java. Με τον όρο φορητότητα εννοούμε ότι ένα πρόγραμμα μπορεί να χρησιμοποιηθεί σε λειτουργικά συστήματα διαφορετικά από αυτό στο οποίο δημιουργήθηκε χωρίς να απαιτείται σημαντική επανεπεξεργασία. Η μεταφορά είναι απαραίτητη για την εκτέλεση οποιασδήποτε εργασίας που είναι απαραίτητη για την εκτέλεση του προγράμματος στο νέο περιβάλλον. Τέλος, η JVM (java virtual machine) διαχειρίζεται τα servlets και δε χρειάζεται να ανησυχούμε για διαρροή μνήμης, συλλογη σκουπιδιών κτλ. Με επιστημονική έννοια, servlet είναι μία κλάση γλώσσας προγραμματισμού της Java που χρησιμοποιείται για την επέκταση των δυνατοτήτων των διακομιστών που φιλοξενούν εφαρμογές στις οποιές έχει πρόσβαση μέσω ενός μοντέλου προγραμματισμού αιτήματος-απόκρισης. Αν και οι servlets μπορούν να ανταποκριθούν σε οποιοδήποτε τύπο αιτήματος, χρησιμοποιούνται συνήθως για την επέκταση των εφαρμογών που φιλοξενούνται από διακομιστές Ιστού. Για τέτοιες εφαρμογές, η Java Servlet ορίζει κλάσεις servlet ειδικών για το HTTP.

·LoginServlet

Στο LoginServlet(), έχουμε προσθέσει το super(); το οποίο είναι λέξη-κλειδί , μία μεταβλητή αναφοράς που χρησιμοποιείται για την αναφορά άμεσου αντικειμένου γονικής κλάσης. Δηλαδή κάθε φορά που αναφέρουμε μία υποκλάση, δημιουργείται η γονική κλάση. Στη συέχεια, με τον τελεστή new δημιουργούμε μία παρουσία ενός αντικειμένου που ορίζεται από το χρήστη. Με λίγα λόγια, δημιουργούμε νέα αντικείμενα.

Στο LoginServlet ζητάμε από το χρήστη να μας δώσει username και κωδικό και ελέγχει εάν το username είναι σωστό και υπάρχει στη βάση δεδομένων μας(String usernamevalidation = dao.loginusernameCheck(username)).Η κλάση βρίσκεται στο πακέτο dao στην κλάση SystemDao. Έχουμε δημιουργήσει μία μεταβλητή dao (μοτίβο αντικειμένου πρόσβασης δεδομένων) στην οποία έχουμε εκχωρήσει σύστημα με μοτίβο που παρέχει συγκεκριμένες λειτουργίες δεδομένων χωρίς να εκθέτει λεπτομέρειες της βάσης δεδομένων (δε θέλουμε ένας χρήστης να μπορεί να δει τον κωδικό ενός άλλου). Οπότε στη συνέχεια με το dao ελέγχουμε εάν υπάρχει το username στη βάση δεδομένων μας. Στο if(usernamevalidation!=username) ελέγχουμε εάν δεν υπάρχει το username. Για να μεταβιβαστεί η τιμή από το Servlet σε αρχείο html, η μέθοδος setAttribute() καλείται από το αντικείμενο αίτησης(request). Η μέθοδος setAttribute() λαμβάνει μία είσοδο ως αντικείμενο που

στέλνει τα δεδομένα από τον servlet στον αιτούντα ιστότοπο. Η συγκεκριμένη εντολή γράφεται ως εξής : setAttribute(Όνομα συμβολοσειράς, Object obj-αντικείμενο). Ορίζει το καθορισμένο αντικείμενο στο πεδίο εφαρμογής της εφαρμογής. Επίσης, έχουμε προσθέσει τη διεπαφή ReequestDispatcher η οποία παρέχει τη δυνατότητα αποστολής του αιτήματος σε έναν άλλο πόρο που μπορεί να είναι html,servlet ή jsp. Αυτή η διεπαφή μπορεί επίσης να χρησιμοποιηθεί για να συμπεριλάβει το περιεχόμενο ενός άλλου πόρου επίσης. Είναι ένας από τους τρόπου συνεργασίας servlet. Εάν το username υπάρχει στη βάση δεδομένων μας βάζουμε try catch Exception. Στο exception κατασκευάζουμε ένα NoSuchAlgorithm με μία συμβολοσειρά που περιγράφει τη συγκεκριμένη εξαίρεση, η οποία μπορεί να καθορίσει εάν ο αλγόριθμος είναι διαθέσιμος ή όχι. Βάζουμε το e.printStackTrace() για να διαχειριστούμε λάθη και εξαιρέσεις από τις λεπτομέρειες που λαμβάνουμε με τη συγκεκριμένη εντολή. Εάν καταφέρουμε όμως να μπούμε στο try, τότε ελέγχουμε εάν το κωδικός που δόθηκε είναι σωστός και εάν καταφέρουμε να συνδεθούμε βάζουμε το ρόλο του user δηλαδή εάν είναι μαθητής, καθηγητής ή γραμματέας. Πριν από αυτό έχουμε φροντίσει για την τρέχον HttpSession που σχετίζεται με αυτό το αίτημα και εάν δεν υπάρχει τρέχουσα περίοδο λειτουργίας επιστρέφει μία νέα περίοδο λειτουργίας(HttpSession session = request.getSession(true);). Εάν ο κωδικός δεν είναι σωστός, τότε δεν μας αφήνει η εφαρμογή να συνδεθούμε. Επίσης, το synchronized(session) χρησιμοποιείται όταν συγχρονίζονται 2 συνεδρίες και οι ενέργειες που εκτελούνται σε ένα παράθυροο συνεδρίας έχουν άμεση επίδραση στο άλλο παράθυρπ συνεδρίας(login page and after login page). Τέλος, βάζουμε το ρόλο που έχει ο user στη βάση δεδομένων μας και στην ιστοσελίδα μας.

·LogoutServlet

Στο LogoutServlet, στο HttpSession session = request.getSession(false); βάζουμε το false για να διασφαλίσουμε ότι ένα νέο αντικείμενο συνεδρίας δεν δημιουργείται από προεπιλογή,οπότε χρησιμοποιούμε τη συγκεκριμένη εντολή. Εάν η συνεδρία δεν υπάρχει ήδη μας επιστρέφεται Null. Κάνουμε session.removeAttribute(“username”); με σκοπό να καταργήσουμε το αντικείμενο που είναι δεσμευμένο με το καθορισμένο όνομα από αυτή την περίοδο λειτουργίας. Έπειτα καταργείται η σύνδεση οπότε κάνουμε αποσύνδεση. Έπειτα επιστρέφεται το αντικείμενο ServletContext που αντιστοιχεί στην καθορισμένη διεύθυνση URL του διακομιστή.

·RegisterServlet

Για το Registerservlet, συνδέουμε τη βάση με το πρόγραμμα και αρχίζουμε να λαμβάνουμε τις παραμέτρους που μας δίνει ο χρήστης (String username = request.getParameter(“newusername”);)).Η συγκεκριμένη εντολή διαβάζει μια παράμετρο αίτησης (φόρμας) και ως όρισμα παραμέτρου δίνουμε το username.Το όνομα της παραμέτρου πρέπει να είναι ακριβώς ίδιο με εκείνο του πηγαίου κώδικα HTML και το αποτέλεσμα που παίρνουμε είναι ακριβώς όπως το εισήγαγε ο τελικός χρήστης-αν είναι απαραίτητη η αποκωδικοποίηση γίνεται αυτόματα. Το πρόγραμμά μας εκχωρεί και τον αριθμό μητρώου του χρήστη (από την κλάση Users που είχαμε δημιουργήσει στην προηγούμενη εργασία),μετατρέπουμε τον κωδικό που έδωσε ο χρήστης από bytes στο δεκαεξαδικό σύστημα,βάζουμε τα στοιχεία στο dao.signup και μετά το synchronized(session) ππου έχουμε καταχωρήσει το ρόλο τον θέτουμε και ως Attribute(request.setAttribute(“role”,”student”)). Έχουμε αναλύσει παραπάνω τι κάνει η κάθεμία εντολή.

·SecretaryController

Στο SecretaryController, κάνουμε ότι κάναμε και στο LoginServlet, δηλαδή προσθέτουμε τη μεταβλητή super και ορίζουμε ένα νέο αντικείμενο (SecretaryDao).

Στο SecretaryController, εκτός του ότι του έχουμε βάλει τον ορισμό του ρόλου που αναφέραμε (γραμματέας) και προγουμένως, βάλαμε και τη δυνατότητα να μπορεί να δει ο γραμματέας τα στοιχεία

του ή να μπορεί να κάνει μετατροπή σε αυτά. Τέλος, στη συγκεκριμένη κλάση προσθέσαμε και τη μέθοδο doPost(req,resp); (doPost(HTTPServletRequest,HTTPServletResponse)) η οποία χρησιμοποιείται σε servlets για επεξεργασία αιτημάτων HTTP POST. Το όρισμα HTTPServletRequest μα επιτρέπει να παραλάβουμε όλα τα εισερχόμενα δεδομένα (η κλάση αυτή διαθέτει μεθόδους με τις οποίες μπορούμε να βρούμε πληροφορίες όπως τα δεδομένα της φόρμας, τις κεφαλίδες αίτησης HTTP και το όνομα του υπολογιστή υπηρεσίας του πελάτη) ενώ το όρισμα HTTPServletResponse μας επιτρέπει να καθορίσουμε εξερχόμενες πληροφορίες όπως κωδικούς κατάστασης HTTP και κεφαλίδες απάντησης. Χρησιμοποιείται για την υποβολή των δεδομένων από το πρόγραμμα περιήγησης στον διακομιστή για επεξεργασία. Τα δεδομένα που υποβάλλονται με τον τύπο της μεθόδου POST αποστέλλονται στο σώμα του μηνύματος, ώστε να είναι ασφαλή και να μην είναι ορατά στη διεύθυνση URL. Το @WebServlet("/secretary") και γενικά και στις υπόλοιπες κλάσεις χρησιμοποιείται για τον ορισμό ενός στοιχείου servlet σε μία εφαρμογή ιστού. Αυτός ο σχολιασμός καθορίζεται σε μία κλάση και περιέχει μεταδεδομένα σχετικά με το servlet που δηλώνεται.

·ProfessorController

Στο ProfessorController, επαναλαμβάνουμε την ίδια διαδικασία με την κλάση SecretaryController. Επίσης προσθέτουμε και τη μεταβλητή σουπερ και δημιουργούμε ένα νέο αντικείμενο.(professorDao)

·StudentController

Στην κλάση StudentController επαναλαμβάνουμε την ίδια διαδικασία με την κλάση SecretaryController με τη μόνη διαφορά την προβολή στοιχείων μαθητών. Επίσης προσθέτουμε και τη μεταβλητή σουπερ και δημιουργούμε ένα νέο αντικείμενο.(studentDao)

·Dao

Γενικά, το Dao χρησιμοποιήθηκε για να συνδέσει τη βάση δεδομένων με το πρόγραμμα. Είναι ένα αντικείμενο πρόσβασης δεδομένων στο λογισμικό και παρέχει ορισμένες συγκεκριμένες λειτουργίες δεδομένων χωρίς να εκθέτει λεπτομέρειες της βάσης δεδομένων. Αυτό μας βοηθάει στην απόκρυψη σημαντικών πληροφοριών όπως για παράδειγμα τα στοιχεία των μαθητών ή των καθηγητών. Η προσθήκη του στο πρόγραμμά μας, μας βοηθάει να προστατέψουμε κάποια μέρη του προγράμματος από εκτεταμένες τροποποιήσεις εάν αλλάξει η απόφαση του σχεδιασμού τους. Η προστασία περιλαμβάνει την παροχή μίας σταθερής διεπαφής που προστατεύει το υπόλοιπο πρόγραμμα από πιθανές τροποποιήσεις.

·SecretaryDao-StudentDao

Σε αυτές τις 2 κλάσεις καταχωρούμε τα στοιχεια των μαθητών στην StudentDao και των γραμματέων στη SecretaryDao στη βάση δεδομένων μας και έπειτα τα εμφανίζουμε σε περίπτωση που χρειαστεί.

·ProfessorDao

Σε αυτή την κλάση, παίρνουμε τα στοιχεία των καθηγητών και τα προσθέτουμε στη βάση. Επίσης στην πρώτη κλάση έχουμε χρησιμοποιήσει τη μέθοδο PreparedStatement της διεπαφής Connection η οποία χρησιμοποιείται για την επιστροφή του αντικειμένου της PreparedStatement. Στη συνέχεια,θέλουμε να μετρήσουμε τις σειρές στον πίνακα και το αποτέλεσμα να επιστρέφεται στην ανώνυμη στήλη COUNT(*).Για να έχουμε αυτό το αποτέλεσμα, πρέπει να το εξάγουμε από το αντικείμενο (αριθμό τηλεφώνου) ResultSet με τη μέθοδο getInt() , αφού το αποτέλεσμα είναι ακέραιος,

·SystemDao

Στην κλάση SystemDao αρχικά προσθέσαμε μία μετατροπή από bytes στο δεκαεξαδικό σύστημα κυρίως για τη μετατροπή του κωδικού. Στην κλάση getSalt προσθέτουμε το όνομα μέσω στη βάση μας ενώ

στην επόμενη κλάση ελέγχουμε αν το όνομα υπάρχει σε αυτή αλλιώς βγάζουμε μήνυμα σφάλματος. Επίσης, ελέγχουμε εάν ο κωδικός του συγκεκριμένου ονόματος είναι σωστός . Όταν ο χρήστης κάνει εγγραφή, ψάχνουμε στη βάση μας για το εάν υπάρχει το συγκεκριμένο όνομα με σκοπό να μην έχουμε χρήστες με ίδια ονόματα. Τέλος, βάζουμε τα δεδομένα που μας δίνονται από τους μαθητές και τους γραμματείς στη βάση μας με σκοπό να μπορούμε να τα χρησιμοποιήσουμε μέσα στο πρόγραμμα(όπως εκεί που πρέπει να εμφανίζουμε πληροφορίες για αυτούς).

·2.4

Το βήμα 2.4 υλοποιείται με τον κώδικα response.setHeader που εξηγήσαμε προηγουμένως στην περίπτωση όπου η χρονική περίοδος όπου ο χρήστης είναι συνδεδεμένος ληξει. Επίσης υλοποιείται με την προσθήκη του κώδικα package controller;

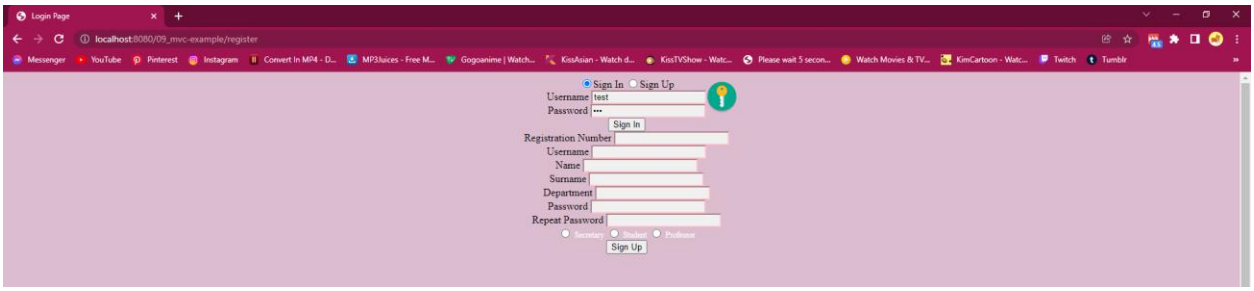
```
@WebServlet("/logout")
public class LogoutServlet extends HttpServlet {
	private static final long serialVersionUID = 1L;

  public LogoutServlet() {
    super();
    // TODO Auto-generated constructor stub
  }

	protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

		HttpSession session=request.getSession(false);
		session.removeAttribute("username");
		session.invalidate();
		RequestDispatcher requestDispatcher = getServletContext().getRequestDispatcher("/index.jsp");
		requestDispatcher.forward(request,response);
	}
```

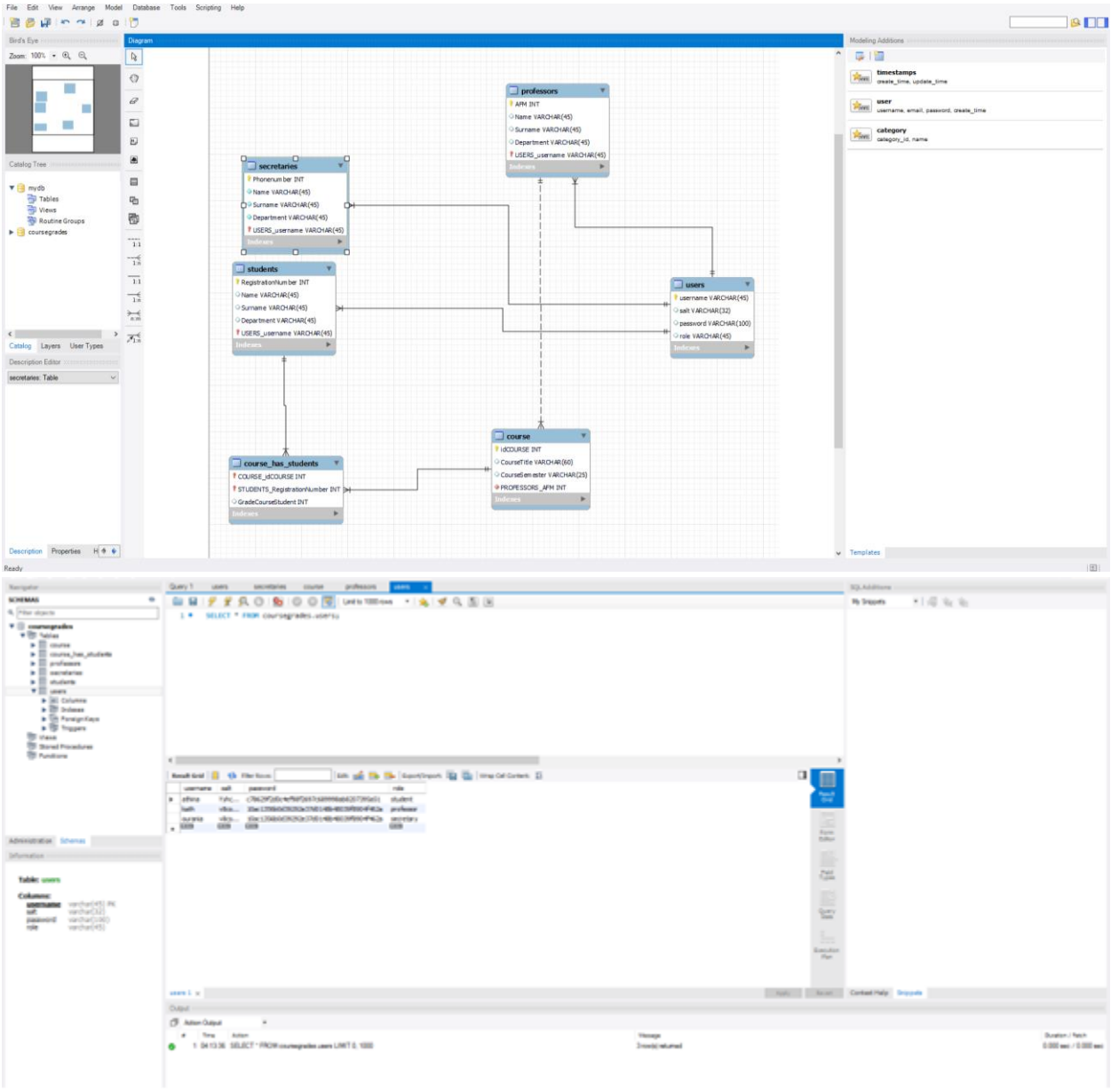
}

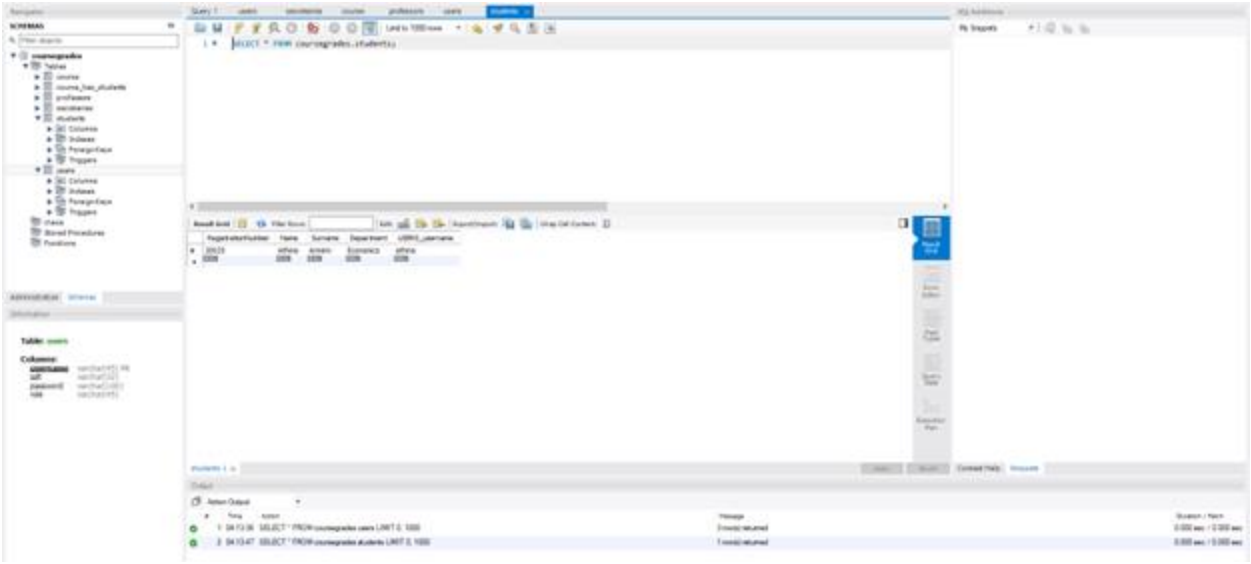


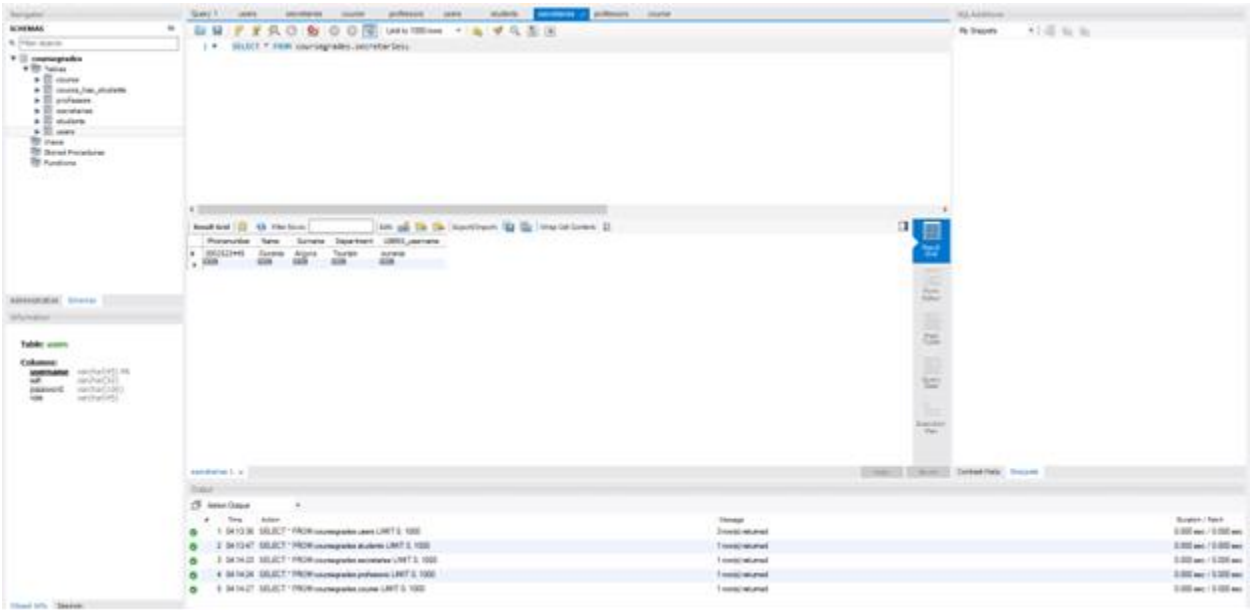
3ο βήμα : Ολοκλήρωση επιπέδου δεδομένων

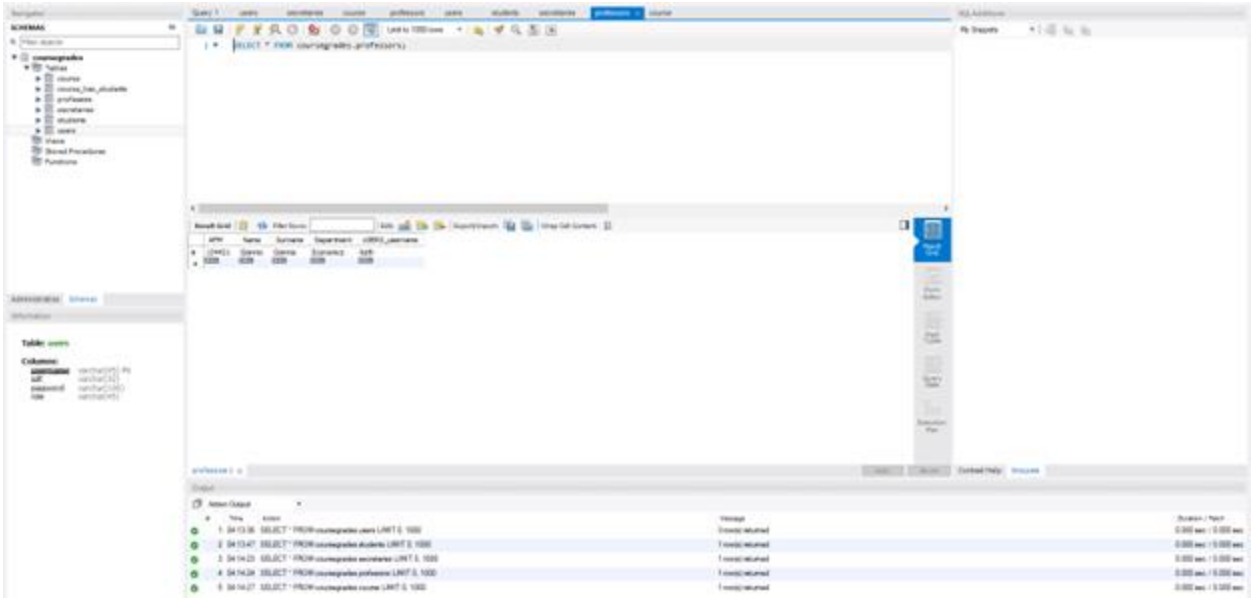
·3.1

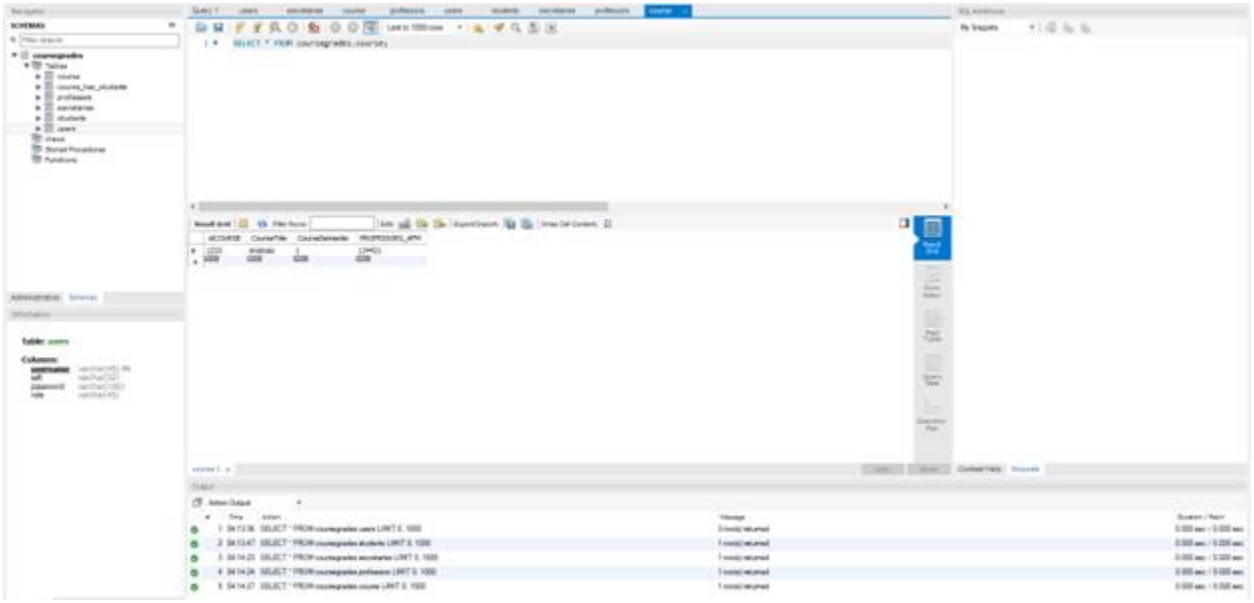
Δημιουργία βάσης δεδομένων όπου αποθηκεύονται όλα τα δεδομένα που έχουμε προσθέσει (mysql workbench).Η βάση δεδομένων μας είναι ανοιχτού κώδικα.Δημιουργήσαμε το αρχείο της βάσης δεδομένων και δημιουργήσαμε τη βάση με το CREATE DDATABASE.Ελέγχουμε τη βάση δεδομένων μας μέσω JDBC(επιτρέπει στα προγράμματα java να έχουνε πρόσβαση σε συστήματα διαχείρισης βάσεων δεδομένων). Μοντέλο οντοτήτων-σχέσεων. Για τη σύνδεση έχουμε χρησιμοποιήσει την κλάση dbutil.java


File Edit View Arrange Model Database Tools Scripting Help
Diagram
professors
APH INT
Name VARCHAR(45)
Surname VARCHAR(45)
Department VARCHAR(45)
USERS_username VARCHAR(45)
secretaries
Phonenumber INT
Name VARCHAR(45)
Surname VARCHAR(45)
Department VARCHAR(45)
USERS_username VARCHAR(45)
students
RegistrationNumber INT
Name VARCHAR(45)
Surname VARCHAR(45)
Department VARCHAR(45)
USERS_username VARCHAR(45)
users
username VARCHAR(45)
salt VARCHAR(32)
password VARCHAR(100)
role VARCHAR(45)
course
idCOURSE INT
CourseTitle VARCHAR(60)
CourseSemester VARCHAR(25)
PROFESSORS_APH INT
course_has_students
COURSE_idCOURSE INT
STUDENTS_RegistrationNumber INT
GradeCourseStudent INT
Modeling Additions
timestamps
create_time, update_time
user
username, email, password, create_time
category
category_id, name
Ready









4ο βήμα : Ολοκλήρωση επιπέδου προβολής (html,jsp)

·Jsp

Οι σελίδες JavaServer (JSP) είναι μία τυπική τεχνολογία Java. Στα αρχεία jsp που έχουμε δημιουργήσει έχουμε προσθέσει μέσα html (HyperText Markup Language) και περιέχει τις σελίδες του ιστότοπού μας. Δηλαδή μας δίνεται η δυνατότητα να γράφουμε δυναμικές σελίδες που βασίζονται σε δεδομένα για την εφαρμογή Web Java. Το JSP είναι χτισμένο πάνω από την προδιαγραφή Java Servlet. Όπως μπορούμε να δούμε και στο πρόγραμμα, έχουμε χρησιμοποιήσει και αναλύσει τα Servlets που προσθέσαμε (LoginServlet, LogoutServlet, RegisterServlet). Με λίγα λόγια, χρησιμοποιείται για τη δημιουργία δυναμικού περιεχομένου Ιστού. Επίσης, χρησιμοποιούνται ετικέτες για την εισαγωγή κώδικα Java στις σελίδες html. Για παράδειγμα, στο αρχείο professor.jsp έχουμε βάλει **<%} %>** `if (request.getAttribute("action")==("viewCourses2"))` **<% {%>** δηλαδή ενώ όλος ο κώδικας είναι γραμμένος σε html, με τις συγκεκριμένες ετικέτες μπορούμε να γράψουμε java.

Στο WebContent έχουμε δημιουργήσει 3 ιστοσελίδες οι οποίες εμφανίζουν τα αποτελέσματα.

Στο Professor.jsp εμφανίζουμε τις πληροφορίες των μαθητών και τους βαθμούς τους σε στήλες, στο Secretary.jsp εμφανίζουμε το όνομα του μαθήματος, το εξάμηνο στο οποίο βρίσκεται και ποιος καθηγητής το διδάσκει, όπως επίσης και το ID του καθηγητή και τέλος στο student.jsp εμφανίζουμε τις πληροφορίες του μαθητή, τους βαθμούς του, το εξάμηνο στο οποίο βρίσκεται και μετά από επιλογή μαθήματος του εμφανίζουμε και το βαθμό του σε αυτό. Να σημειωθεί ότι σε όλες αυτές τις ιστοσελίδες , έχουμε βάλει και τη χρονική περίοδο η οποία εάν λήξει, αποσυνδέει αυτόματα το χρήστη.