

**Πανεπιστήμιο Πειραιώς
Τμήμα Πληροφορικής
Έτος: 2021 - 2022**



Μάθημα:
«Προγραμματισμός Στο Διαδίκτυο Και Στον Παγκόσμιο Ιστό»
Εργασία: 2η Άσκηση Μαθήματος
Εξάμηνο: 4ο

Ομάδα εργασίας:

Αιμιλιανός Κουρπάς Δανάς Π20100,
Απόστολος Σιαμπάνης Π20173,
Αγγελική Καλδίρη Π20069,
Θεόδωρος Κοξάνογλου Π20094

Περιεχόμενα

1. Εκφώνηση της άσκησης.....	4
1.1 Αναλυτικά Βήματα:.....	4
1.2 Οδηγίες:.....	5
2. Γενική Περιγραφή Προγράμματος.....	6
2.1 Βιβλιοθήκες:	7
2.2 JSP σελίδες:.....	7
2.3 Βάση Δεδομένων:	8
3. Κώδικας Προγράμματος	8
3.1 Περιγραφή των Servlets του Package com.controller.....	8
3.2 Περιγραφή των Κλάσεων ανά Package	9
3.3 Περιγραφή των σελίδων JSP	10
3.4 Περιγραφή των Servlets του Package com.controller.....	12
3.5 Βάση Δεδομένων Project:	13
4. Εκτέλεση Προγράμματος.....	14
4.1 Προϋποθέσεις εκτέλεσης προγράμματος	14
4.2 Φωτογραφίες από το UI:	18

5. Βιβλιογραφικές Πηγές.....	27
------------------------------	----

1. Εκφώνηση της άσκησης

2η Άσκηση Μαθήματος

Στόχοι άσκησης: web project, δημιουργία Βάσης Δεδομένων της εφαρμογής, υλοποίηση ορισμένων λειτουργιών. Σε αυτή την άσκηση θα γίνουν τα εξής:

- η εγκατάσταση του application server (π.χ. Tomcat) και του database server (π.χ. mysql ή postgres) και η μεταξύ τους διασύνδεση,
- η δημιουργία του dynamic web project το οποίο θα αποτελέσει τον κορμό για την τελική εργασία και η υλοποίηση ορισμένων λειτουργιών.

Στο web project, θα χρησιμοποιήσετε τις κλάσεις που δημιουργήσατε στην προηγούμενη εργασία, με τις κατάλληλες προσθήκες και τροποποιήσεις.

1.1 Αναλυτικά Βήματα:

1. Εγκατάσταση και παραμετροποίηση application server και database server.

- 1.1. Εγκαταστήστε και παραμετροποιήστε τον Tomcat application server (εάν επιθυμείτε μπορείτε να χρησιμοποιήσετε άλλο αντίστοιχο) και το Σύστημα Διαχείρισης Βάσης Δεδομένων (mysql ή postgres). Η εγκατάσταση του application server να συνδεθεί με το περιβάλλον IDE που χρησιμοποιείτε (π.χ. Eclipse).
- 1.2. Δημιουργήστε την σύνδεση του application server με τον database server, χρησιμοποιώντας τον αντίστοιχο jdbc database connector για το σύστημα βάσης της επιλογής σας. Χρησιμοποιήστε τη σύνδεση του μοντέλου 3-tier.

2. Δημιουργία Βάσης Δεδομένων

- 2.1. Δημιουργήστε το Μοντέλο Οντοτήτων-Σχέσεων, το οποίο περιγράφει τη Βάση Δεδομένων που θα χρησιμοποιήσετε για την εφαρμογή σας. Ενδεικτικά θα περιλαμβάνει πίνακες όπως, Φοιτητές, Καθηγητές, Μαθήματα. Να παραλάβετε στο μοντέλο σας τις σχέσεις μεταξύ των πινάκων.
- 2.2. Με τη βοήθεια του Μοντέλου Οντοτήτων-Σχέσεων, να δημιουργήσετε και να εκκινήσετε τη βάση στον database server. Μπορείτε να χρησιμοποιήσετε οποιοδήποτε βοηθητικό εργαλείο για την εξαγωγή της βάσης από το μοντέλο (π.χ. mysql Workbench για mysql).
- 2.3. Εισάγετε εικονικά δεδομένα σε όλους τους πίνακες, λαμβάνοντας υπόψη τα εξωτερικά κλειδιά που πιθανώς έχουν οι πίνακες.

3. Δημιουργία web project και υλοποίηση λειτουργιών

- 3.1. Δημιουργήστε ένα Dynamic Web Project.
- 3.2. Δημιουργήστε ένα ή περισσότερα πακέτα κλάσεων, τα οποία θα περιλαμβάνουν τις βασικές κλάσεις που έχετε υλοποιήσει στην προηγούμενη άσκηση.

- 3.3. Δημιουργήστε ένα νέο πακέτο κλάσεων το οποίο θα περιλαμβάνει όλα τα servlet που θα χρησιμοποιήσετε στην εργασία (ενδεικτικά και όχι περιοριστικά, μπορείτε να ορίσετε ένα servlet για όλες τις λειτουργίες κάθε κατηγορίας χρήστη: StudentServlet, ProfessorServlet, SecretaryServlet, ή όποια άλλη δομή θέλετε). Στην συγκεκριμένη άσκηση θα υλοποιήσετε μόνο ένα μέρος από ένα από αυτά όπως αναφέρεται στο επόμενο βήμα.
- 3.4. Για το/τα servlet το οποίο θα υλοποιεί τις λειτουργίες της Γραμματείας να υλοποιήσετε τις παρακάτω λειτουργίες:
 - 3.4.1. Λειτουργία σύνδεσης (login) .
 - 3.4.2. Προβολή συνόλου μαθημάτων.
 - 3.4.3. Προβολή μαθημάτων και υπεύθυνου Καθηγητή ανά μάθημα.
 - 3.4.4. Ανάθεση μαθήματος σε Καθηγητή.
- 3.5. Για την προβολή του αποτελέσματος κάθε μίας από τις παραπάνω ενέργειες, θα δημιουργείται μία δυναμική html σελίδα μέσω του servlet (ή συνδυασμό servlet και JSP). Δημιουργήστε επίσης τις απαραίτητες στατικές html σελίδες που απαιτούνται.

1.2 Οδηγίες:

- Ισχύουν οι ίδιες ομάδες και οι ίδιες οδηγίες με την προηγούμενη εργασία.
- Το συνολικό παραδοτέο θα περιλαμβάνει σε ένα συμπιεσμένο αρχείο: (α) το project, (β) τη βάση δεδομένων (.sql ή .mwb αρχείο εάν χρησιμοποιείτε το Workbench) και (γ) την τεκμηρίωση αντίστοιχα

2. Γενική Περιγραφή Προγράμματος

Ακολουθώντας τις γενικές οδηγίες της εργασίας δημιουργήσαμε ένα Dynamic Web Project στο Eclipse IDE for Enterprise and Web Developers:

Τίτλος project: Ergasia_2_Java_Tomcat9

Όνομα αρχείου βάσης δεδομένων: dbfor2.mwb

Όνομα βάσης δεδομένων: gradedb

Project packages:

- com.controller
- com.dbUtil
- com.sqlFunctions
- Com.university

Εσωτερικοί φάκελοι του webapp:

- adminPages
- css
- images
- js
- professorPages
- secretaryPages
- studentPages

Στο **package com.controller** δημιουργήσαμε **10 servlets** (LoginServlet, LogoutServlet, SessionServlet, RegisterProfessorServlet, RegisterSecretaryServlet, RegisterStudentServlet, ProfessorController, SecretaryController, StudentController, CoursesController).

Στο **package com.dbUtil** δημιουργήσαμε **2 class** (DBConnection, DBAccount).

Στο **package com.sqlFunctions** δημιουργήσαμε **4 classes** (ProfessorDBFunction, SecretaryDBFunction, StudentDBFunction, CoursesDBFunction).

Στο **package com.university** δημιουργήσαμε **8 classes** (AssignCourse, Course, Department, Grade, Professor, Secretary, Student, User).

Στο πρόγραμμα είναι υλοποιημένες οι βασικές λειτουργίες της τελικής εφαρμογής, όσον αφορά την γραμματεία.

Σε αυτή τη φάση της εργασίας θεωρούμε ότι η γραμματεία μπορεί να γράψει μαθήματα, φοιτητές και καθηγητές που ανήκουν μόνο σε ένα τμήμα της σχολής στη βάση δεδομένων. Για παράδειγμα: **Department of Informatics**.

2.1 Βιβλιοθήκες:

Το project περιλαμβάνει τέσσερις (4) βασικές βιβλιοθήκες:

- 1) **com.controller:** περιέχει controllers και servlets που αναλαμβάνουν την επικοινωνία του φυλλομετρητή με τον server και αντίστροφα.
- 2) **com.dbUtil:** περιλαμβάνει κλάσεις που περιέχουν μεθόδους σύνδεσης του project με την βάση και επαλήθευση στοιχείων των χρηστών της πλατφόρμας όταν προσπαθούν να συνδεθούν.
- 3) **com.sqlFunctions:** έχει όλες τις κατάλληλες μεθόδους και τα αντίστοιχα sql query που θα χρειαστεί το project για να είναι λειτουργικές οι λειτουργίες που έχει αναλάβει ο κάθε χρήστης
- 4) **com.university:** όλες τις κλάσεις που υπήρχαν και στο πρώτο μέρος της εργασίας. Στη βιβλιοθήκη αυτή περιλαμβάνονται όλες οι κλάσεις και οι κληρονομικότητες που έχουν οι οντότητες της εφαρμογής (για παράδειγμα class student, professor και courses)

2.2 JSP σελίδες:

Επιπλέον περιέχει jsp σελίδες χωρισμένες σε φακέλους. Κάθε φάκελος ορίζεται είτε ως σελίδες χρήστη ή λειτουργίες που υλοποιούν όλες ή ένα μέρος των jsp σελίδων.

Φάκελοι χρήστη:

- 1) **studentPages:** περιλαμβάνει τις σελίδες του χρήστη student (courses, grades, profile)
- 2) **professorPages:** περιλαμβάνει τις σελίδες του χρήστη professor (courses, grades, profile)
- 3) **secretaryPages:** περιλαμβάνει τις σελίδες του χρήστη secretary (addCourse, addNewCTop, assignCToP, courses, editCourse, editProfessor, editStudent, professors, profile, register, registerProfessor, registerStudent, students, viewCourses, viewCToP)

(επιπλέον υπάρχουν και οι σελίδες jsp που είναι κοινές για όλους τους χρήστες, όπως είναι η index, login και sessionPage.)

(**Σημείωση:** Οι περετέρο σελίδες στους φακέλους **studentPages** και **professorPages** θα υλοποιηθούν στην επόμενη εργασία.)

Φάκελοι μορφοποίησης:

- 1) **css**: περιλαμβάνει όλα τα αρχεία μορφοποίησης των σελίδων `jsp`. Ανάλογα με τα στοιχεία της κάθε σελίδας, φορτώνονται και τα αντίστοιχες σελίδες.
- 2) **js**: περιέχει αρχεία που κάνουν όλες της σελίδες `html` δυναμικές.

2.3 Βάση Δεδομένων:

Τέλος, το project περιλαμβάνει μία **βάση δεδομένων**, στην οποία αποθηκεύονται οι χρήστες και τα μαθήματα, όπως είναι τα προσωπικά στοιχεία του χρήστη και η ανάθεση μαθημάτων στους καθηγητές και τους μαθητές.

Στην επόμενη ενότητα, αναλύουμε τα παραπάνω αρχεία του project ως προς τις κατηγορίες τους, τα χαρακτηριστικά τους και τις λειτουργίες τους στο πρόγραμμα.

3. Κώδικας Προγράμματος

3.1 Περιγραφή των Servlets του Package `com.controller`

- **LoginServlet**: είναι το servlet που χρησιμοποιείται για τη λειτουργία login και τους ανάλογους ελέγχους για την υλοποίηση της.
- **LogoutServlet**: είναι το servlet που χρησιμοποιείται για τη λειτουργία logout της εφαρμογής.
- **SessionServlet**: είναι το servlet που χρησιμοποιείται για την λειτουργία του session management.
- **RegisterProfessorServlet**: είναι το servlet που χρησιμοποιείται για την υλοποίηση της εγγραφής και δημιουργίας ενός χρήστη, ιδιότητας καθηγητή.
- **RegisterSecretaryServlet**: είναι το servlet που χρησιμοποιείται για την υλοποίηση της εγγραφής και δημιουργίας ενός χρήστη, ιδιότητας γραμματείας.
- **RegisterStudentServlet**: είναι το servlet που χρησιμοποιείται για την υλοποίηση της εγγραφής και δημιουργίας ενός χρήστη, ιδιότητας φοιτητή.
- **ProfessorController**: είναι το servlet που χρησιμοποιείται για την υλοποίηση όλων των λειτουργιών μεταξύ του προγράμματος και της βάσης δεδομένων, των αντικειμένων τύπου καθηγητή.
- **SecretaryController**: είναι το servlet που χρησιμοποιείται για την υλοποίηση της ανάθεσης μαθημάτων στους καθηγητές, καθώς και την προβολή των μαθημάτων μαζί με τον υπεύθυνο καθηγητή ανά μάθημα.

- **StudentController:** είναι το servlet που χρησιμοποιείται για την υλοποίηση όλων των λειτουργιών μεταξύ του προγράμματος και της βάσης δεδομένων, των αντικειμένων τύπου φοιτητή.
- **CoursesController:** είναι το servlet που χρησιμοποιείται για την υλοποίηση όλων των λειτουργιών μεταξύ του προγράμματος και της βάσης δεδομένων, των αντικειμένων τύπου μάθημα.

3.2 Περιγραφή των Κλάσεων ανά Package

Όπως αναφέραμε υπάρχουν **13 κλάσεις** όπου η γενική τους λειτουργία είναι:

- **com.dbUtil:**
 - **DBConnection:** είναι η κλάση που αναλαμβάνει την σύνδεση της εφαρμογής με την βάση δεδομένων.
 - **DBAccount:** είναι η κλάση που με τις μεθόδους που περιέχει βοηθάει στην κρυπτογράφηση του κωδικού που δημιουργείται και χρησιμοποιείτε από τους χρήστες. Επίσης, υποστηρίζει την λειτουργία για την αναζήτηση του username του χρήστη στον κατάλληλο πίνακα κάθε φορά για την ολοκλήρωση του ελέγχου της σύνδεσης στην εφαρμογή.
- **com.sqlFunctions:**
 - **coursesDBFunction:** είναι η κλάση που με τις μεθόδους της υλοποιεί και τις ανάλογες λειτουργίες για τα αντικείμενα των μαθημάτων, όπου συνδέονται και με τη βάση δεδομένων.
 - **professorDBFunction:** είναι η κλάση που με τις μεθόδους της ολοκληρώνει την σύνδεση και την εγγραφή ενός χρήστη καθηγητή, καθώς επίσης υλοποιεί και τις ανάλογες λειτουργίες για τα αντικείμενα των καθηγητών, όπου συνδέονται με τη βάση δεδομένων.
 - **secretaryDBFunction:** είναι η κλάση που με τις μεθόδους της ολοκληρώνει την σύνδεση και την εγγραφή ενός χρήστη γραμματεία.
 - **studentDBFunction:** είναι η κλάση που με τις μεθόδους της ολοκληρώνει την σύνδεση και την εγγραφή ενός χρήστη φοιτητή, καθώς επίσης υλοποιεί και τις ανάλογες λειτουργίες για τα αντικείμενα των φοιτητών, όπου συνδέονται με τη βάση δεδομένων.
- **com.university:**
 - **AssignCourse:** είναι η κλάση που η κύρια λειτουργία της είναι να δείχνει τους καθηγητές που αντιστοιχούν ανά μάθημα.

- **Course:** είναι η κλάση των μαθημάτων. Τα αντικείμενά της έχουν όλα τα γνωρίσματα των μαθημάτων της σχολής. Μερικά γνωρίσματα μαθημάτων είναι: η περιγραφή του μαθήματος, οι απαιτήσεις του και το τμήμα στο οποίο ανήκουν.
- **Department:** καθορίζει το τμήμα που ανήκουν οι φοιτητές, οι καθηγητές, η γραμματεία και τα μαθήματα.
- **Grade:** είναι η κλάση που η κύρια λειτουργία της είναι να δείχνει τον βαθμό των φοιτητών ανά μάθημα.
- **Professor:** είναι η κλάση των εκπαιδευτικών.
- **Secretary:** είναι η κλάση της γραμματείας.
- **Student:** είναι η κλάση των φοιτητών.
- **User:** είναι abstract class από την οποία υιοθετούν γνωρίσματά της οι φοιτητές, οι καθηγητές και η γραμματεία. Για παράδειγμα και οι τρεις κατηγορίες χρηστών έχουν όνομα, username και password, γνωρίσματα τα οποία υιοθετούν από την κλάση User.

3.3 Περιγραφή των σελίδων JSP ανά φάκελο και σελίδες που δεν βρίσκονται σε φάκελο

- **adminPages**

- **registerSecretary.jsp:** είναι η σελίδα που θα γίνεται η δημιουργία των χρηστών της γραμματείας.

- **professorPages**

- **courses.jsp:** είναι η σελίδα που θα εμφανίζονται τα μαθήματα του καθηγητή. (η λειτουργία αυτή θα υλοποιηθεί στην επόμενη εργασία)
- **grades.jsp:** είναι η σελίδα που θα δίνεται η επιλογή στον καθηγητή να προσθέσει τους βαθμούς στους φοιτητές που είναι εγγεγραμμένοι στο μάθημα. (η λειτουργία αυτή θα υλοποιηθεί στην επόμενη εργασία)
- **profile.jsp:** είναι η σελίδα που θα εμφανίζονται τα στοιχεία του καθηγητή. (η λειτουργία αυτή θα υλοποιηθεί στην επόμενη εργασία)

- **secretaryPages**

profile.jsp: είναι η σελίδα που γίνεται η προβολή των στοιχείων του χρήστη της γραμματείας που έχει ολοκληρώσει την σύνδεση.

courses.jsp: είναι η σελίδα που δίνει την δυνατότητα στον χρήστη της γραμματείας να επιλέξει την προβολή των διαθέσιμων μαθημάτων ή την προσθήκη ενός μαθήματος. Το menu περιέχει:

- **addCourse.jsp:** είναι η σελίδα που γίνεται η δημιουργία των μαθημάτων.
- **editCourse.jsp:** είναι η σελίδα που δίνεται η δυνατότητα στον χρήστη να αλλάξει τις πληροφορίες ενός μαθήματος.
- **viewCourses.jsp:** είναι η σελίδα που γίνεται η προβολή όλων των διαθέσιμων μαθημάτων.

professors.jsp: είναι η σελίδα που γίνεται η προβολή όλων των εγγεγραμμένων καθηγητών. Μέσω αυτού του πίνακα, η γραμματεία μπορεί να επιλέξει να κάνει edit ή delete τον professor.

- **editProfessor.jsp:** είναι η σελίδα που δίνεται η δυνατότητα στον χρήστη να αλλάξει τις πληροφορίες ενός καθηγητή.

students.jsp: είναι η σελίδα που γίνεται η προβολή όλων των εγγεγραμμένων φοιτητών. Μέσω αυτού του πίνακα, η γραμματεία μπορεί να επιλέξει να κάνει edit ή delete τον professor.

- **editStudent.jsp:** είναι η σελίδα που δίνεται η δυνατότητα στον χρήστη να αλλάξει τις πληροφορίες ενός φοιτητή.

register.jsp: είναι η σελίδα που δίνει την δυνατότητα στον χρήστη της γραμματείας να επιλέξει τι είδους χρήστη θέλει να δημιουργήσει.

- **registerStudent.jsp:** είναι η σελίδα στην οποία γίνεται η δημιουργία των φοιτητών, από την γραμματεία.
- **registerProfessor.jsp:** είναι η σελίδα στην οποία γίνεται η δημιουργία των καθηγητών, από την γραμματεία.

assignCToP.jsp: είναι η σελίδα που δίνει την δυνατότητα στο χρήστη της γραμματείας να επιλέξει αν θέλει είτε να κάνει ανάθεση κάποιου μαθήματος σε κάποιο καθηγητή είτε να προβάλει τις ήδη υπάρχουσες αναθέσεις μαθημάτων σε καθηγητές.

- **addNewCToP.jsp:** είναι η σελίδα που γίνεται η επιλογή μαθήματος και καθηγητή, για την υλοποίηση της ανάθεσης μαθήματος σε καθηγητή.
- **viewCToP.jsp:** είναι η σελίδα που γίνεται η προβολή όλων των μαθημάτων όπου έχει γίνει η ανάθεση τους σε τουλάχιστον ένα καθηγητή.

● studentPages

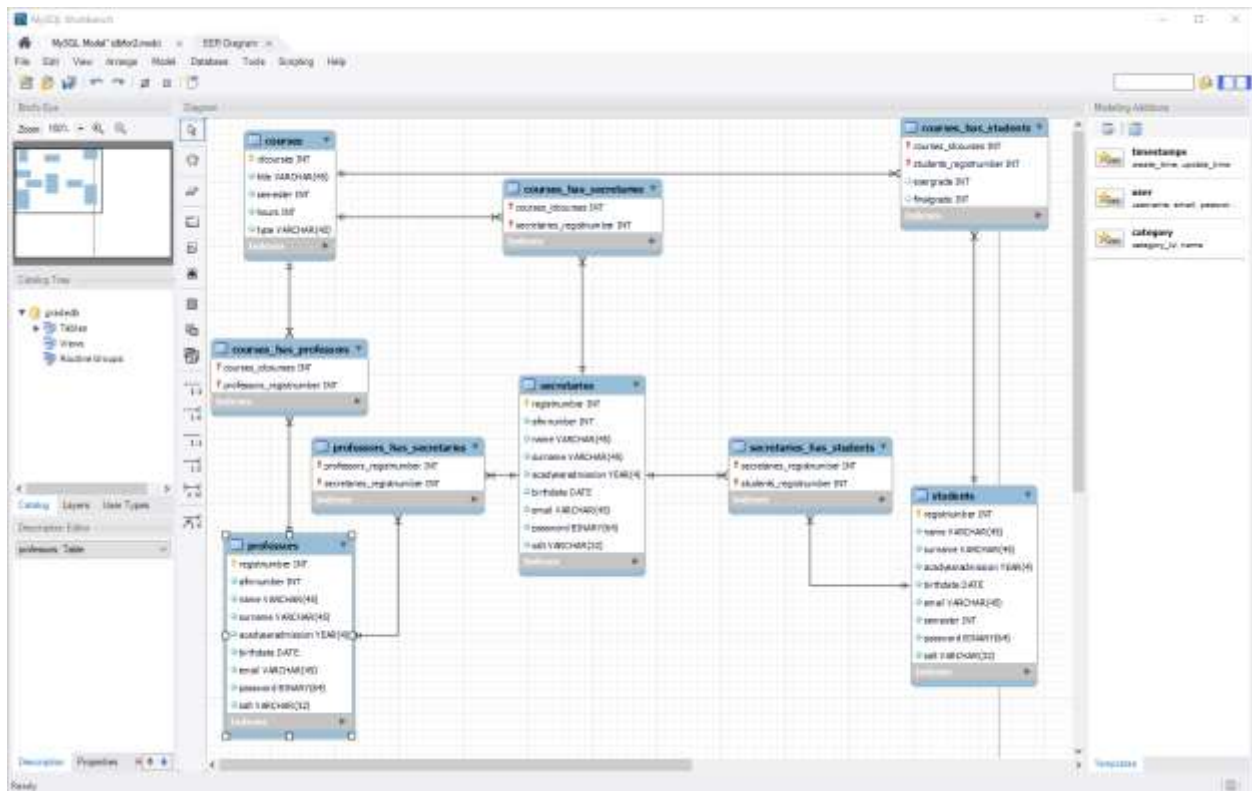
- **courses.jsp:** είναι η σελίδα που θα εμφανίζονται τα μαθήματα του φοιτητή.
(η λειτουργία αυτή θα υλοποιηθεί στην επόμενη εργασία)
- **grades.jsp:** είναι η σελίδα που θα εμφανίζονται οι βαθμοί του φοιτητή.
(η λειτουργία αυτή θα υλοποιηθεί στην επόμενη εργασία)

- **profile.jsp**: είναι η σελίδα που θα εμφανίζονται τα στοιχεία του φοιτητή.
(η λειτουργία αυτή θα υλοποιηθεί στην επόμενη εργασία)
- **errorpage.jsp**: είναι η σελίδα στην οποία οδηγείται ο χρήστης σε περίπτωση κάποιου σφάλματος από τον tomcat server. (η λειτουργία αυτή θα υλοποιηθεί στην επόμενη εργασία)
- **Index.jsp**: είναι η αρχική σελίδα που δίνει την επιλογή στο χρήστη είτε να μεταβεί στην σελίδα login για την υλοποίηση της σύνδεσης είτε να βγει από την εφαρμογή.
- **login.jsp**: είναι η σελίδα που υλοποιείται η σύνδεση με την εφαρμογή.
- **sessionPage.jsp**: είναι η σελίδα που χρησιμοποιείται ως μεταβατική για τον χρήστη της πλατφόρμας, από την σελίδα login.jsp στην αντίστοιχη profile.jsp, ανάλογα με την ιδιότητα που έχει.

3.4 Περιγραφή των Servlets του Package com.controller

1. **LoginServlet**: είναι το servlet που χρησιμοποιείται για τη λειτουργία login και τους ανάλογους ελέγχους για την υλοποίηση της.
2. **LogoutServlet**: είναι το servlet που χρησιμοποιείται για τη λειτουργία logout της εφαρμογής.
3. **SessionServlet**: είναι το servlet που χρησιμοποιείται για την λειτουργία του session management.
4. **RegisterProfessorServlet**: είναι το servlet που χρησιμοποιείται για την υλοποίηση της εγγραφής και δημιουργίας ενός χρήστη, ιδιότητας καθηγητή.
5. **RegisterSecretaryServlet**: είναι το servlet που χρησιμοποιείται για την υλοποίηση της εγγραφής και δημιουργίας ενός χρήστη, ιδιότητας γραμματείας.
6. **RegisterStudentServlet**: είναι το servlet που χρησιμοποιείται για την υλοποίηση της εγγραφής και δημιουργίας ενός χρήστη, ιδιότητας φοιτητή.
7. **ProfessorController**: είναι το servlet που χρησιμοποιείται για την υλοποίηση όλων των λειτουργιών μεταξύ του προγράμματος και της βάσης δεδομένων, των αντικειμένων τύπου καθηγητή.
8. **SecretaryController**: είναι το servlet που χρησιμοποιείται για την υλοποίηση της ανάθεσης μαθημάτων στους καθηγητές, καθώς και την προβολή των μαθημάτων μαζί με τον υπεύθυνο καθηγητή ανά μάθημα.
9. **StudentController**: είναι το servlet που χρησιμοποιείται για την υλοποίηση όλων των λειτουργιών μεταξύ του προγράμματος και της βάσης δεδομένων, των αντικειμένων τύπου φοιτητή.
10. **CoursesController**: είναι το servlet που χρησιμοποιείται για την υλοποίηση όλων των λειτουργιών μεταξύ του προγράμματος και της βάσης δεδομένων, των αντικειμένων τύπου μάθημα.

3.5 Βάση Δεδομένων Project:



Η βάση δημιουργήθηκε με την χρήση του προγράμματος MySQL Workbench

Έχουν δημιουργηθεί οι ακόλουθοι πίνακες:

courses, professors, secretaries, students

Όλες οι σχέσεις που δημιουργήθηκαν μεταξύ των πινάκων αυτών είναι σχέσεις **πολλά προς πολλά** (n:m) :

- courses με professors
-> δημιουργείται ενδιάμεσος πίνακας **courses_has_professors**
- courses με students
-> δημιουργείται ενδιάμεσος πίνακας **courses_has_students**
- courses με secretaries
-> δημιουργείται **courses_has_secretaries**
- professors με secretaries
-> δημιουργείται **professors_has_secretaries**
- secretaries με students
-> δημιουργείται **secretaries_has_students**

Φτάσαμε στο τέλος της περιγραφής της εργασίας, στην επόμενη ενότητα θα βρείτε οδηγίες για το πως να εκτελέσετε το project στον υπολογιστή σας και screenshots από το τελικό αποτέλεσμα της εργασίας.

4. Εκτέλεση Προγράμματος

Σημείωση: οι παρακάτω οδηγίες είναι μόνο για χρήστες που χρησιμοποιούν λειτουργικό Windows 10 21H1 ή νεότερη έκδοση. Το project αυτό μπορεί να δοκιμαστεί και σε linux os αλλά η εμπειρία από έκδοση σε έκδοση μπορεί να ποικίλλει.

4.1 Προϋποθέσεις εκτέλεσης προγράμματος

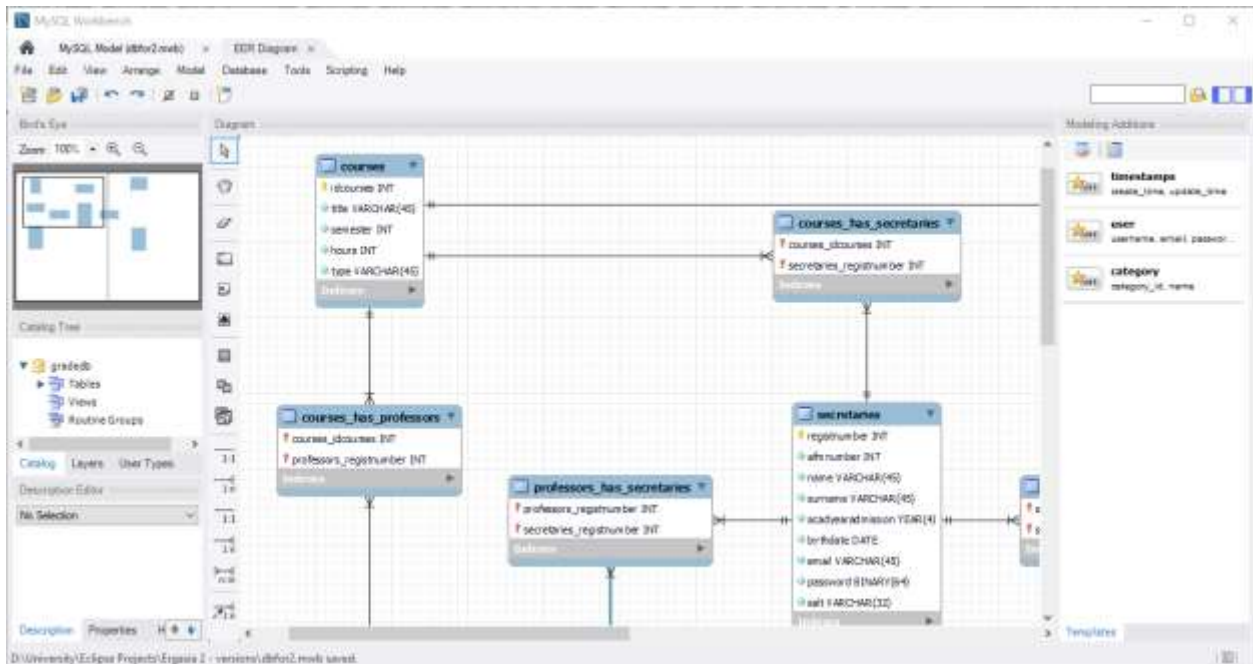
Να έχετε εγκαταστήσει στον τοπικό σας υπολογιστή τα εξής προγράμματα:

- ❖ [MySQL Workbench 8.0 CE](#) ή νεότερη έκδοση
- ❖ [Java v17](#) ή νεότερη (συμβουλευτείτε το documentation εδώ)
- ❖ [Eclipse IDE for Enterprise Java and Web Developers](#) 2022-3 ή νεότερη έκδοση
- ❖ [Apache-tomcat-9.0](#) (εγκαθίσταται εύκολα με την χρήση του eclipse)

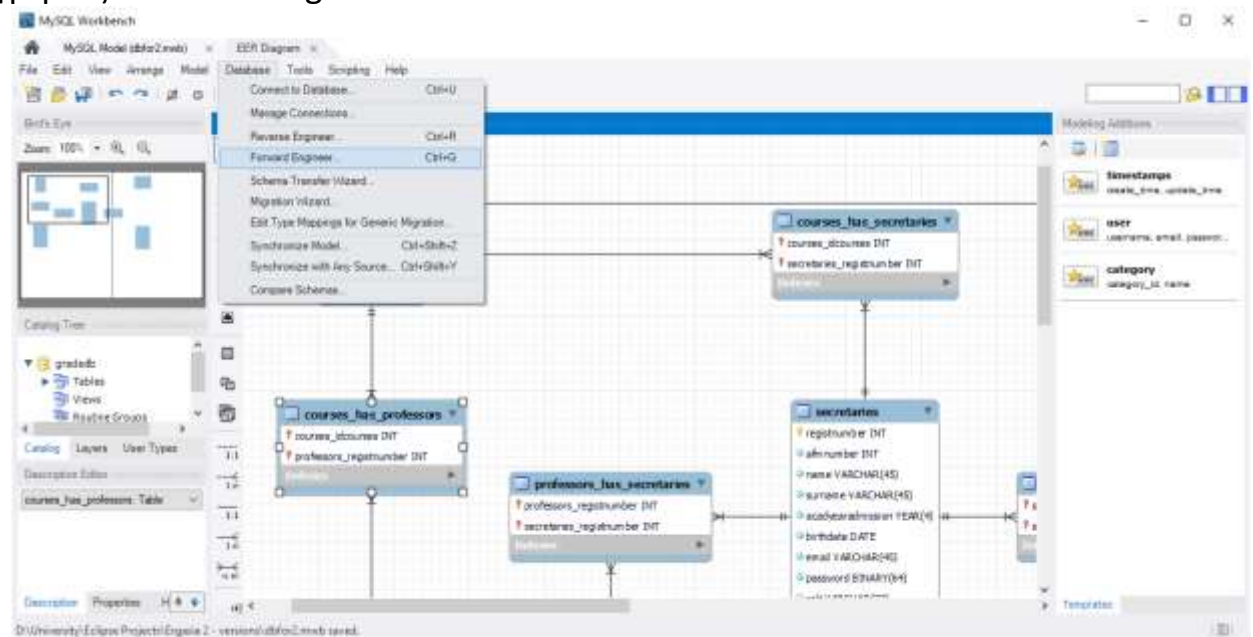
Αρχικά, κατεβάζετε το το αρχείο “Ergasia_2_Java.zip” και το κάνετε αποσυμπίεση με το εργαλείο αποσυμπίεσης του λειτουργικού.

Στη συνέχεια ανοίγεται τα MySQL Workbench και το Eclipse. Στην εφαρμογή Workbench ανοίγεται την βάση που θα βρείτε μέσα στο συμπιεσμένο αρχείο του project με τίτλο “dbfor2.mwb” που περιέχει την απαραίτητη βάση με τα δεδομένα της εργασίας.

Όταν ανοίξει η βάση θα πρέπει να σας ανοίξει ένα παράθυρο σαν αυτό που βλέπετε παρακάτω:



Ύστερα, επιλέγεται από το πάνω αριστερό μενού κατηγορία “Database” και από εκεί την κατηγορίας “Forward Engineer”:



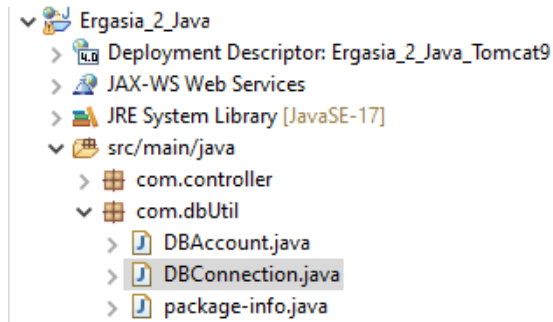
(Σημείωση: θα χρειαστεί να έχετε τον κωδικό που έχετε για τον server της mysql σας, γιατί θα χρειαστεί στο επόμενο βήμα)

Εκτελείται το wizard που θα εμφανιστεί. Όταν τελειώσει με επιτυχία, είσαστε έτοιμοι για το επόμενο βήμα.

Από εδώ και πέρα δεν θα χρειαστεί να έχετε ανοιχτό το πρόγραμμα Workbench.

Τώρα που τελειώσατε με την εγκατάσταση της βάσης σας μεταβείτε στο πρόγραμμα eclipse και ανοίξτε το project (είναι το ίδιο αρχείο που μόλις αποσυμπιέστε)

Με το που το ανοίξατε το project, κάτω από το μονοπάτι:
Ergasia_2_Java_Tomcat9\src\main\java\com\dbUtil



ανοίγεται την κλάση DBConnection και αλλάζετε το username, password με αυτό του χρήστη που θέλατε να συνδεθεί στην βάση.

```
import java.sql.Connection;

public class DBConnection {

    private static String dbURL = "jdbc:mysql://localhost/gradedb?useSSL=false";
    private static String dbUsername = "root";
    private static String dbPassword = "pass";
    private static String dbDriver = "com.mysql.jdbc.Driver";

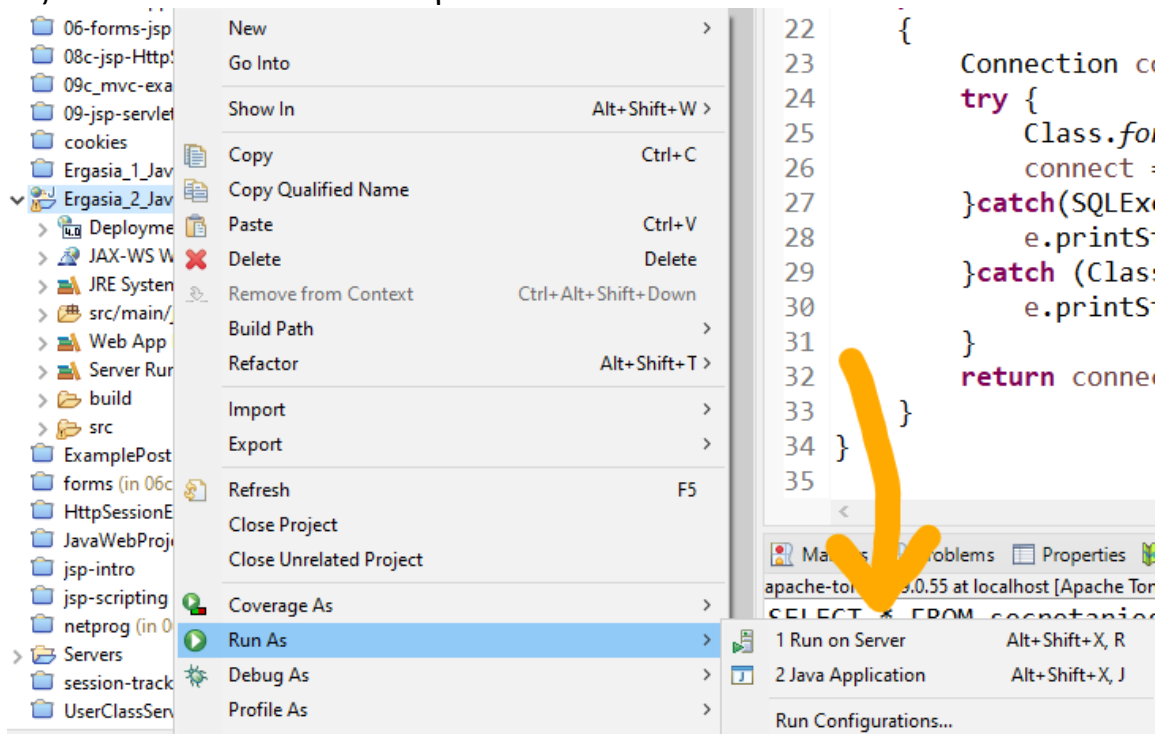
    public static Connection connectToDB()
    {
        Connection connect = null;
        try {
            Class.forName(dbDriver);
            connect = DriverManager.getConnection(dbURL, dbUsername, dbPassword);
        } catch (SQLException e) {
            e.printStackTrace();
        } catch (ClassNotFoundException e) {
            e.printStackTrace();
        }
        return connect;
    }
}
```

Πριν ξεκινήσετε το project, μην ξεχάσετε να τοποθετήσετε το mysql connector κάτω από το φάκελο lib του tomcat 9 server σας:

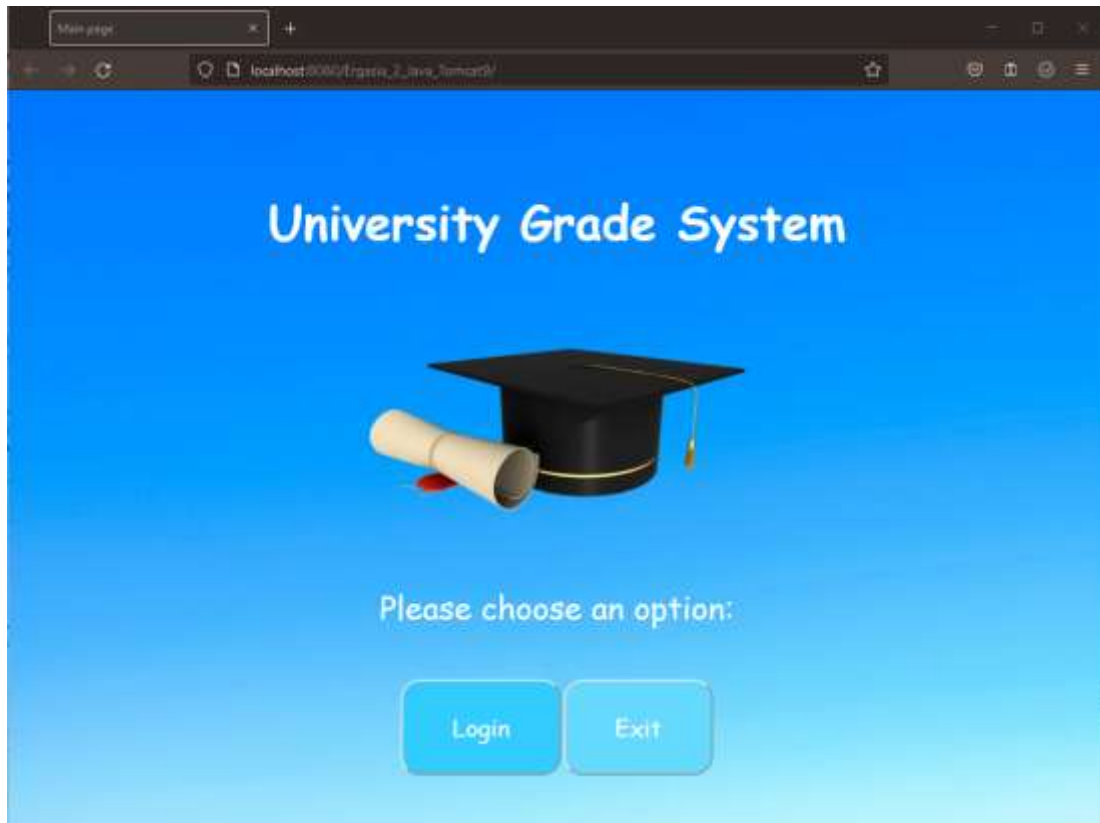
University > Tomcat > apache-tomcat-9.0.55 > lib

Name	Date modified	Type	Size
annotations-api.jar	14/06/2022 22:22	JAR File	13 KB
catalina.jar	14/06/2022 22:22	JAR File	1,705 KB
catalina-ant.jar	14/06/2022 22:22	JAR File	54 KB
catalina-ha.jar	14/06/2022 22:22	JAR File	122 KB
catalina-ssi.jar	14/06/2022 22:22	JAR File	62 KB
catalina-storeconfig.jar	14/06/2022 22:22	JAR File	77 KB
catalina-tribes.jar	14/06/2022 22:22	JAR File	331 KB
ecj-4.20.jar	14/06/2022 22:22	JAR File	3,061 KB
el-api.jar	14/06/2022 22:22	JAR File	88 KB
jasper.jar	14/06/2022 22:22	JAR File	558 KB
jasper-el.jar	14/06/2022 22:22	JAR File	169 KB
jspic-api.jar	14/06/2022 22:22	JAR File	27 KB
jsp-api.jar	14/06/2022 22:22	JAR File	63 KB
mysql-connector-java-8.0.29.jar	08/03/2022 18:20	JAR File	2,461 KB

Έτοιμοι! Τώρα μπορείτε να τρέξετε το project μέσω του eclipse.
Κάνετε δεξί κλικ πάνω του και επιλέγεται:



Θα πρέπει να ανοίξει ο προεπιλεγμένος φυλλομετρητής που έχετε ορίσει στο σύστημά σας, και υποδεχτείτε από αυτή την "index.jsp":



Από δω και πέρα, μπορείτε να κάνετε Login στην πλατφόρμα ως Secretary, Professor ή Student (στην τελική εργασία θα περιλαμβάνεται txt αρχείο με defaults usernames και password για κάθε κατηγορία χρήστη στην πλατφόρμα)

Όταν κάνετε login μπορείτε να μεταφερθείτε στην βασική σελίδα του χρήστη (profile.jsp), όπου από αυτή μπορείτε να περιηγηθείτε στις δυνατότητες που έχει ο χρήστης από το πλαϊνό μενού.

4.2 Φωτογραφίες από το UI:

Ακολουθούν φωτογραφίες από το UI της εφαρμογής.

University
Form login

Username

Password

Login

Παράδειγμα login χρήστη

Secretary's Profile

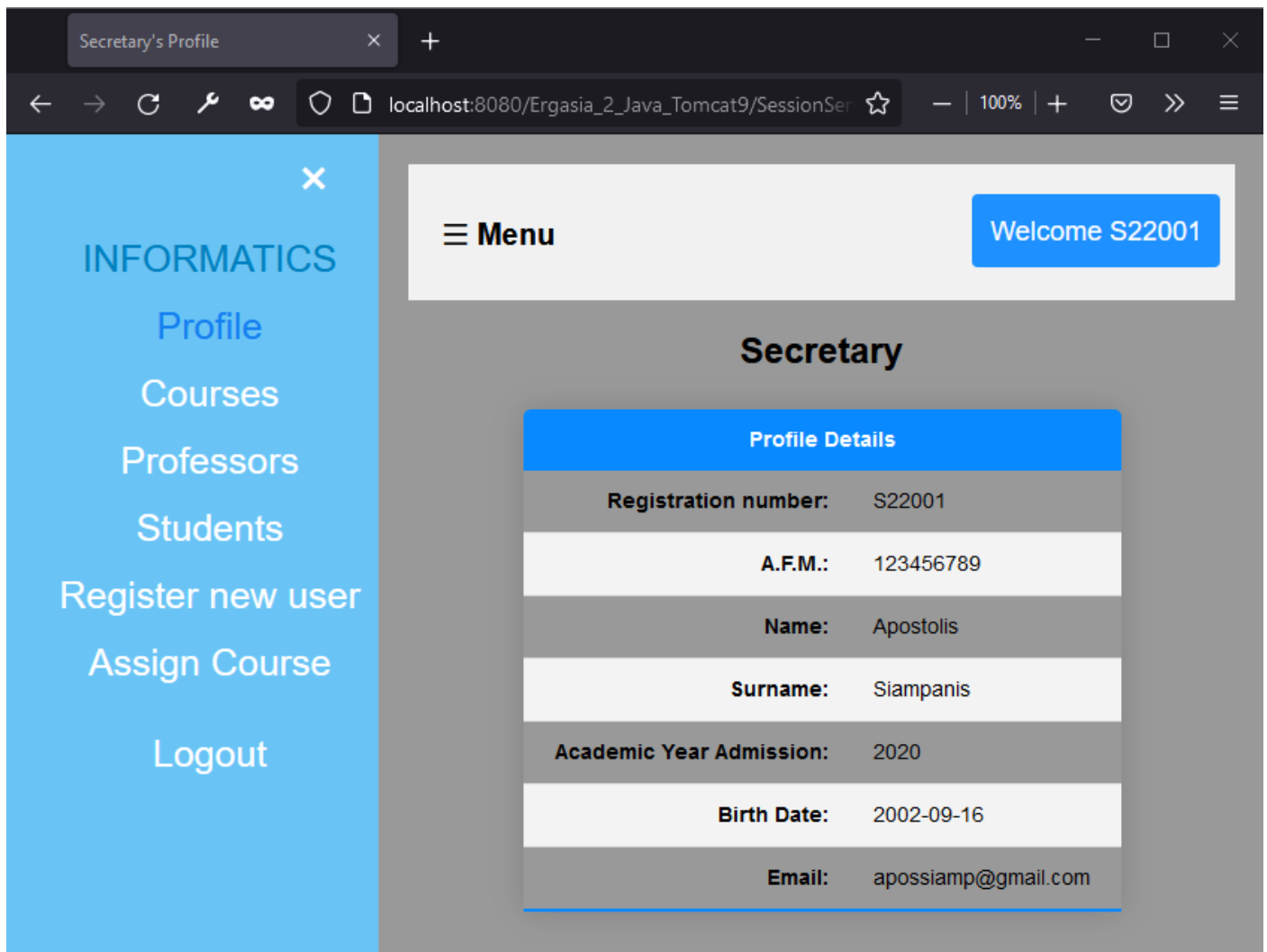
Menu

Welcome S22001

Secretary

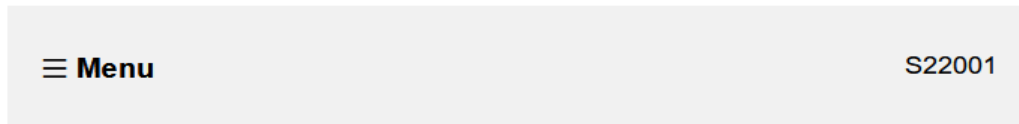
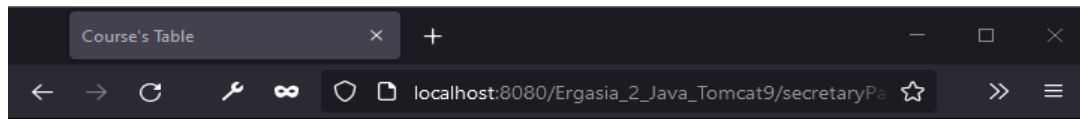
Profile Details	
Registration number:	S22001
A.F.M.:	123456789
Name:	Apostolis
Surname:	Siampanis
Academic Year Admission:	2020
Birth Date:	2002-09-16
Email:	apossiampan@gmail.com

Παράδειγμα dummy χρήστη secretary profile



Πλαϊνό μενού με επιλογές

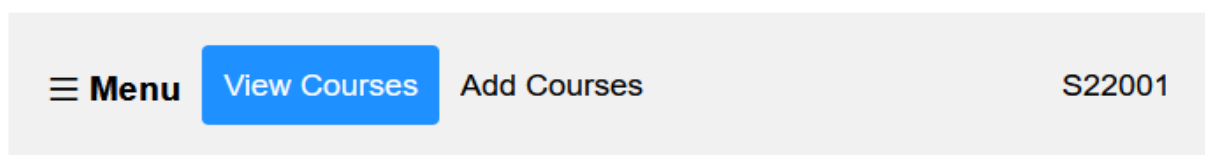
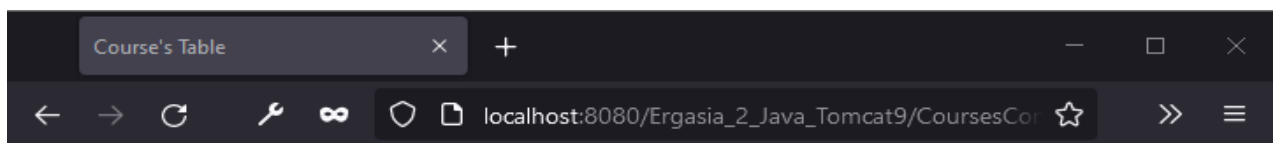
Όταν ο χρήστης θελήσει να βγει από την πλατφόρμα, επιλέγει logout και οδηγείται αυτόματα στην αρχική σελίδα.



View
Courses

Add Courses

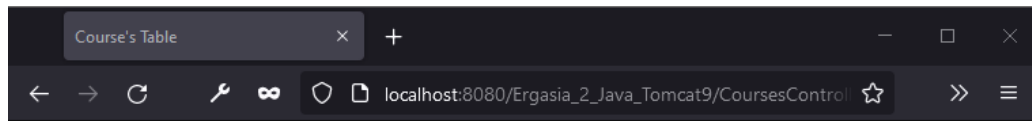
Επιλογή Menu courses



Course's Table

Course ID	Title	Semester	Hours	Type	Action
1	Analysi	1	6	mandatory	Edit Delete
2	Chemistry	3	8	mandatory	Edit Delete

Προβολή όλων των Courses



≡ **Menu** View Courses **Add Courses** S22001

Add a course

ID Course

Title

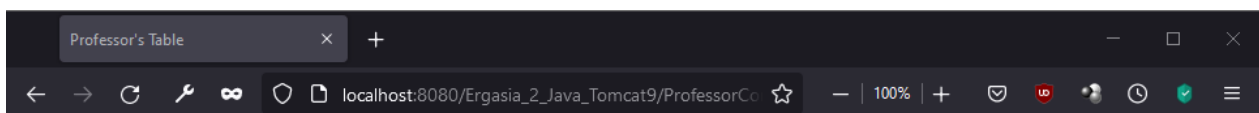
Semester

Hours

Type

Submit

Προσθήκη ενός Course

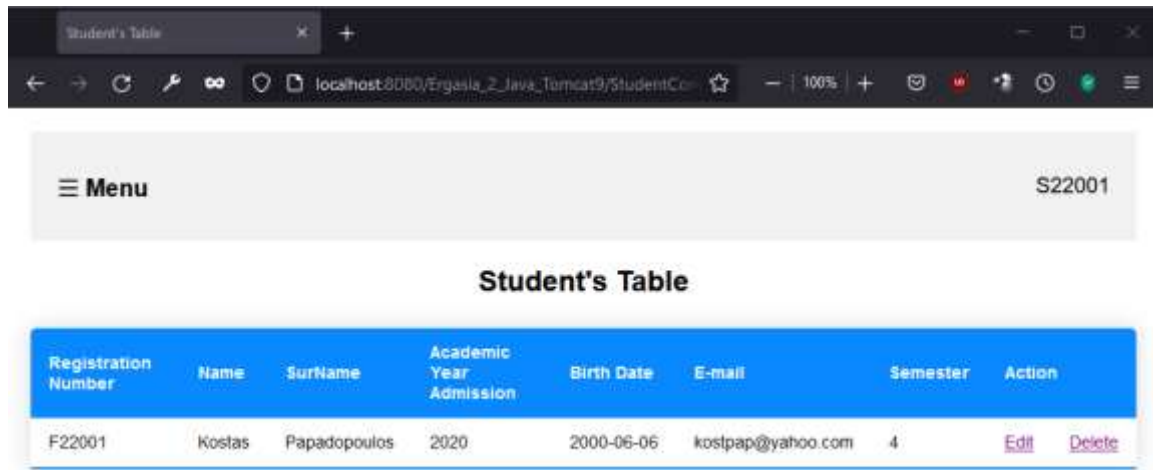


≡ **Menu** S22001

Professor's Table

Registration Number	A.F.M.	Name	SurName	Academic Year Admission	Birth Date	E-mail	Action
P22001	234567891	Kostas	Paraskevopoulos	2016	1980-12-02	kostparask@gmail.com	Edit Delete

Προβολή όλων των Professors



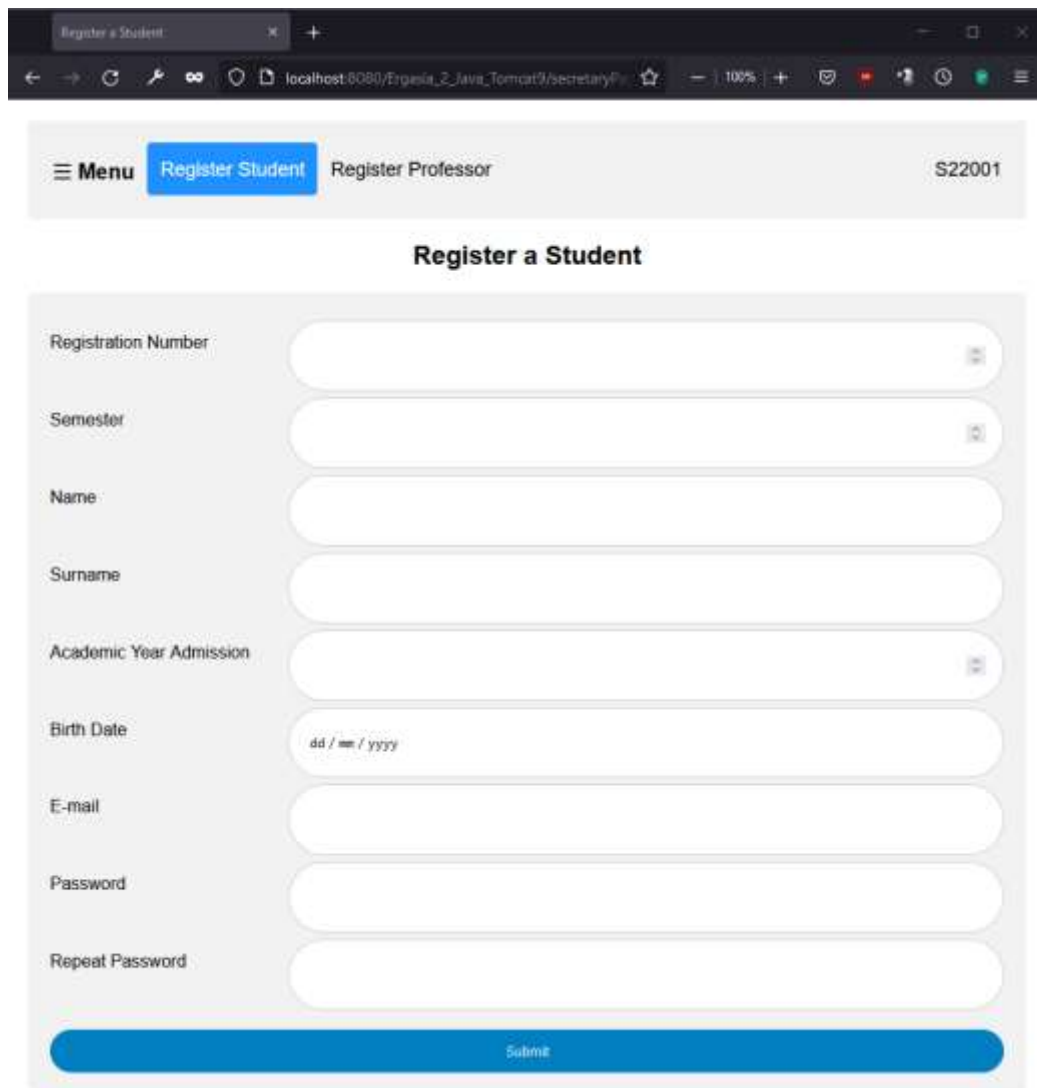
Student's Table

Menu S22001

Student's Table

Registration Number	Name	SurName	Academic Year Admission	Birth Date	E-mail	Semester	Action
F22001	Kostas	Papadopoulos	2020	2000-06-06	kostpap@yahoo.com	4	Edit Delete

Menu students



Register a Student: Menu Register Student Register Professor S22001

Register a Student

Registration Number

Semester

Name

Surname

Academic Year Admission

Birth Date

E-mail

Password

Repeat Password

Submit

Register Student

Register a Professor

Menu Register Student Register Professor S22001

Register a Professor

Registration Number

Name

Surname

AFM number

Academic Year Admission

Birth Date

E-mail

Password

Repeat Password

Submit

Register Professor

Register Secretary

localhost:8080/ergasia_2_java_tomcat9/adminPage

Secretary Form

Create new secretary account! This is ONLY available to administration.

Registration Number

Name

Surname

AFM number

Academic Year Admission

Birth Date

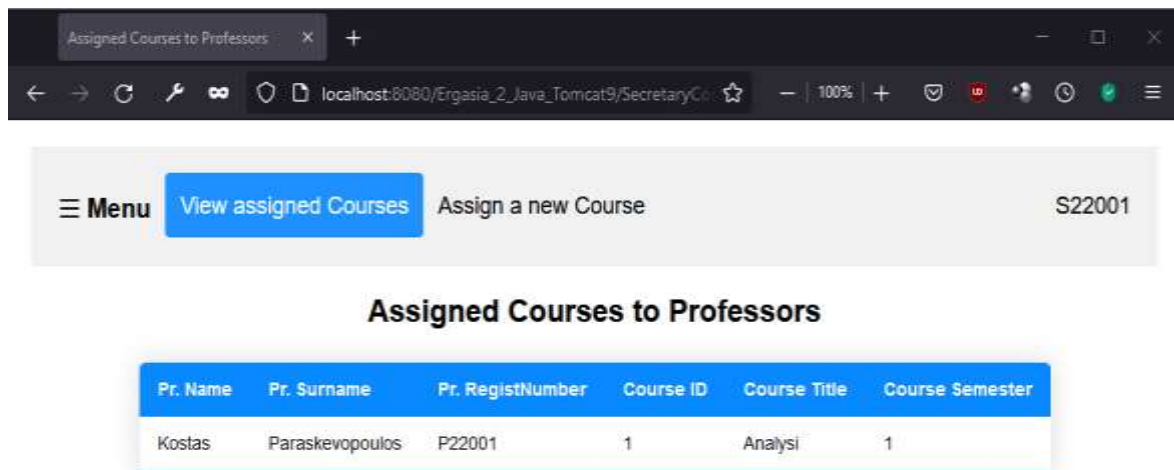
E-mail

Password

Repeat Password

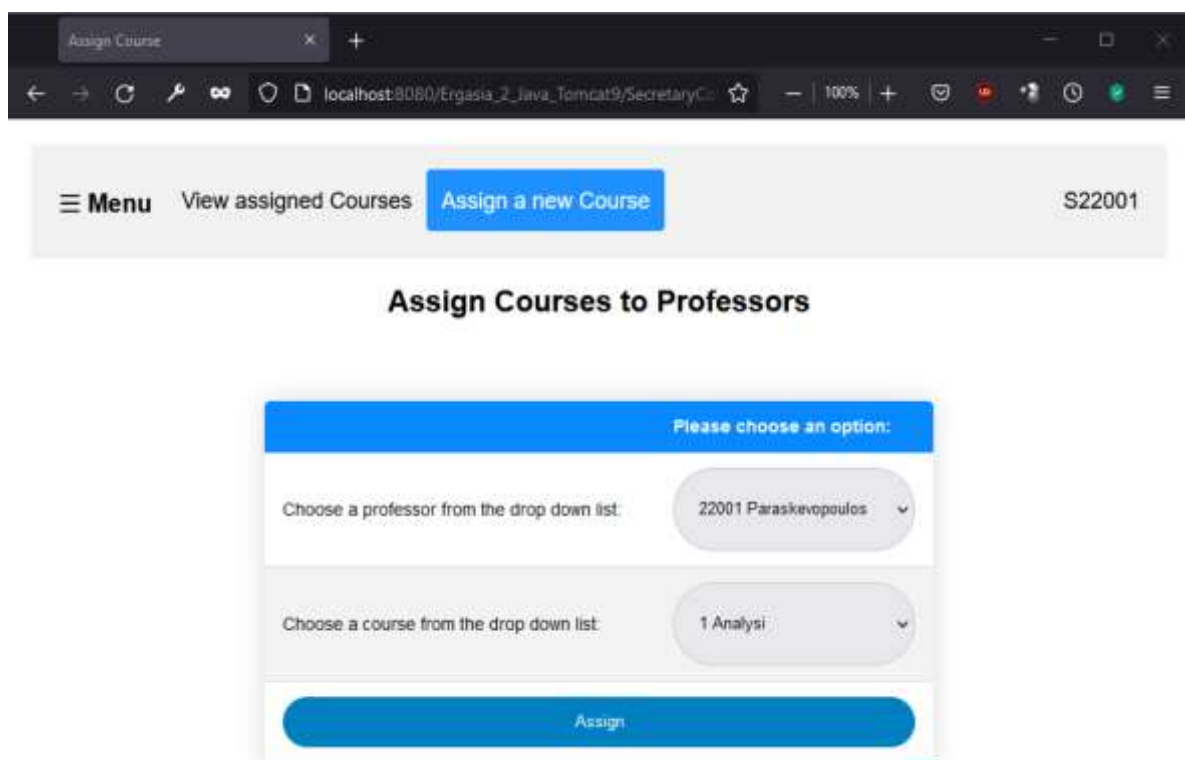
Submit

Secretary Registration Form, only available



Pr. Name	Pr. Surname	Pr. RegistNumber	Course ID	Course Title	Course Semester
Kostas	Paraskevopoulos	P22001	1	Analysis	1

Menu Assign Course



Assign Course to a Professor

Βασικό troubleshooting

Σε περίπτωση που δεν υπάρχει κάποια εγγραφή secretary στη βάση δεδομένων, αρκεί να ανοίξετε μέσα από το eclipse το registerSecretary.jsp (κάτω από τον φάκελο adminPages) και να προσθέσετε secretary account!

5. Βιβλιογραφικές Πηγές

1. **ΕΡΓΑΣΤΗΡΙΑΚΑ ΠΑΡΑΔΕΙΓΜΑΤΑ (Java)**, Retrieve 10/06/2022 from <https://gunet2.cs.unipi.gr/modules/document/document.php?course=TMB117&openDir=/201102161747563cexy8tj>
2. **ΠΑΡΟΥΣΙΑΣΕΙΣ**, Retrive 05/06/2022 from <https://gunet2.cs.unipi.gr/modules/document/document.php?course=TMB117&openDir=/20110216172215d4kp21g1>