ΧΑΡΟΚΟΠΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ
HAROKOPIO UNIVERSITY

Networked and Distributed Systems

System of admission of students to postgraduate program

Christos Lamprou

Table of Contents

Introduction

This is a simple implementation of a management system for students applying for admission to a postgraduate study program. The application is divided into two distinct parts, the internal (backend, api) and the external (frontend) for security reasons. The internal system is implemented with java spring boot framework and the external with reactjs and nodejs server.

Git Repository:

Frontend: https://github.com/clamprou/students-frontend

Branch: final

Backend: https://github.com/clamprou/springboot-students

Branch: final

Admissions

The system administrator will have the ability through the internal system to manage the entities of the application, for example to import, delete and edit objects and also to manage the roles of each entity. We consider that there is no reason for the administrator to connect to the external system to do the actions he wants, so we didn't implement it.

Every user who registers in the system will definitely have the user role. From then on, in order for the user to proceed in the logic of the system, the administrator must activate his account, in any other case the user will not be able to login to the system (he will simply be registered). By activating the user's account, the administrator also chooses what role he will have in the system (student or secretary). After the user's account has been activated (and therefore he has one of the two previous roles) by logging in to the application proceeds according to its role in every usage scenario.

For the student: A form appears for entering personal information such as (age, phone, address, nationality, studies, the postgraduate course he wants to enroll in and the reason he is applying). Also in the field of the form where the user selects the postgraduate course there is a link that displays detailed information about this course so that the user has a good idea of which course to choose. The user selects submit and the application is created in the system. Finally, he can choose from the menu: show off his application so that he can see its status.

For the Secretary: Two options appear.

Evaluation of student applications: A list of pending student applications appears. There the secretariat has the possibility to see in detail the application form of each student and decide whether to approve or reject it.

Importing a postgraduate course: A list of the postgraduate courses available in the system and their details is displayed. The secretariat has the ability to insert, delete, edit postgraduate courses.

Implementation

Internal System

In the internal system, only the administrator can log in. We chose to use thymeleaf to display the pages to the administrator with mvc since it is relatively easy. The Administrator by connecting to the system has the ability to edit the users, their roles and in general any entity. We also support the swagger interface of all our api controllers.

Regarding the security of the system, we have ensured through spring security with the filtering methods which specific addresses of the application need authentication and which authorization. However, the authentication of the system is done with basic authentication and not with some secure protocol such as Jwt, Pgp, etc. but as you mentioned in the context of this exercise it is not mandatory. Finaly, The Database is MySql and the tables are created automatically through Spring Jpa, i.e. by putting in the properties: spring.jpa.hibernate.ddl-auto=update.

External System

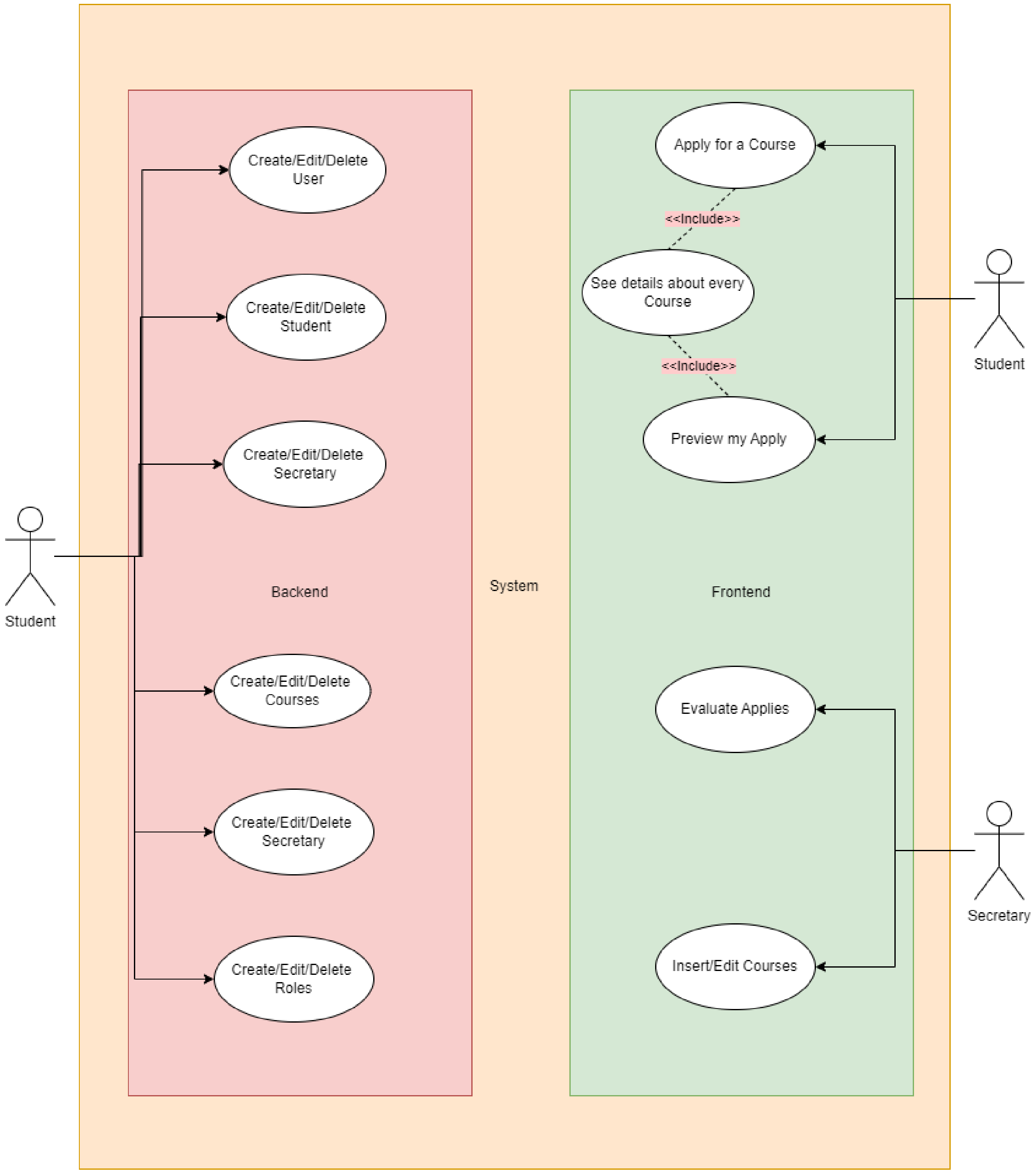
For the external system where students and the secretariat can connect, we have checked the role of each user so that the corresponding page is displayed. So, if a student tries to post a course then an error message will appear (unauthorized).

The technologies we use for the frontend are:

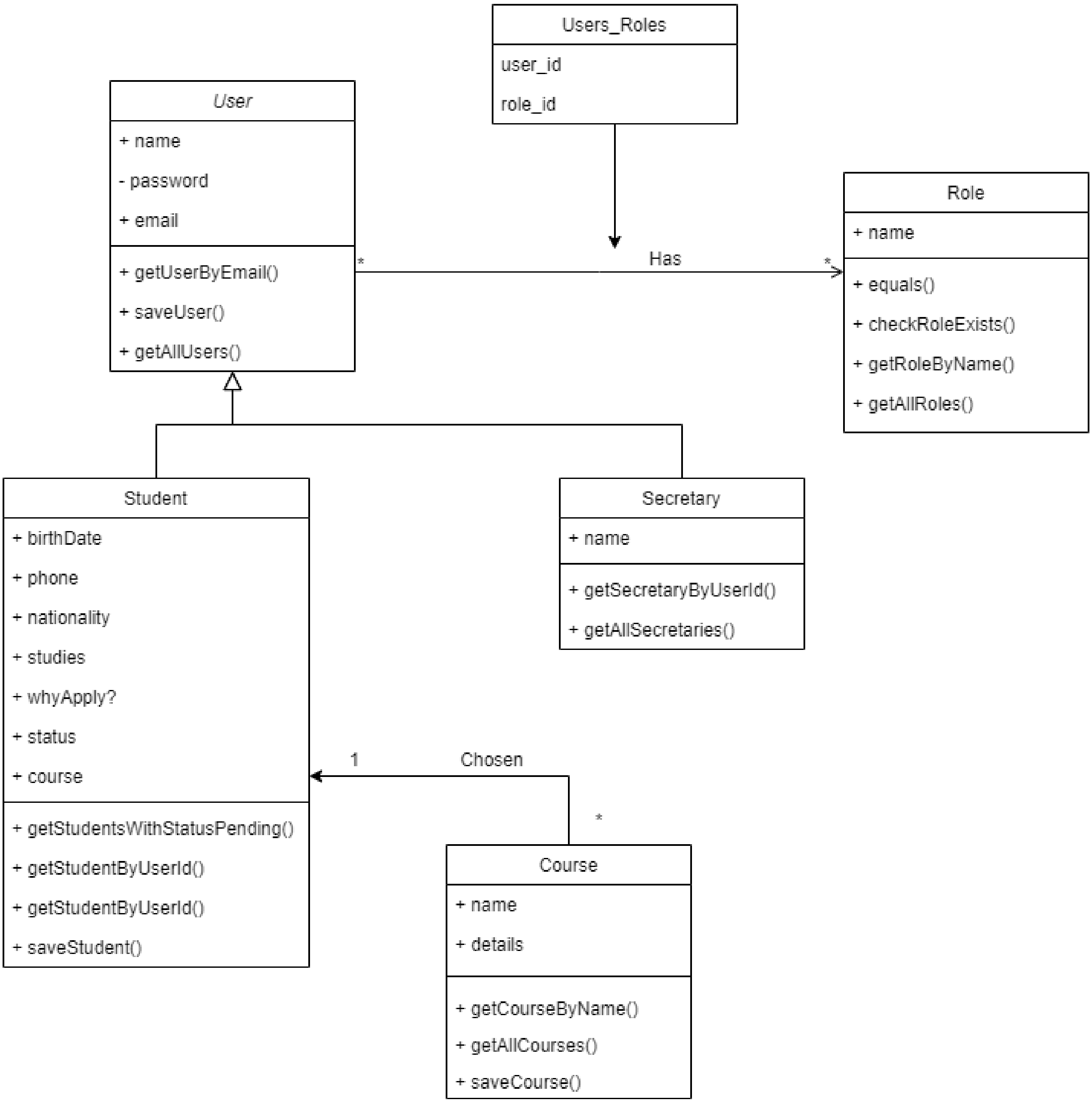
- Simple html pages
- javascript
- Nodejs
- Postman to get xmlhttpRequests auto generated request

Diagrams

Use Case Diagram

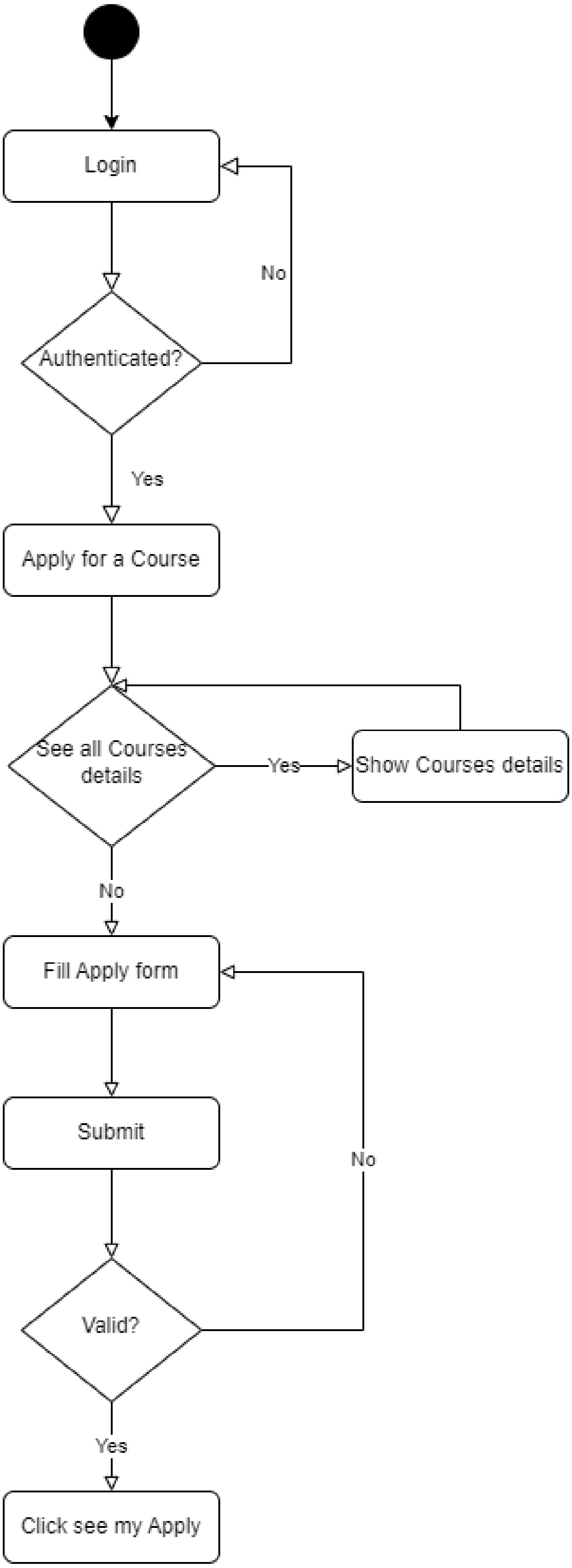


Class Diagram

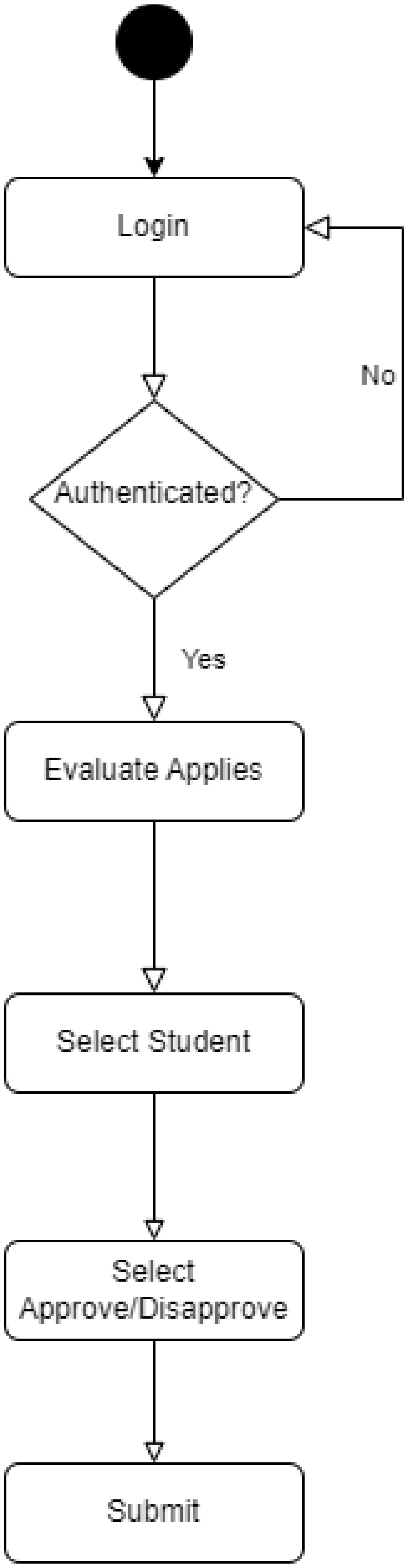


Activity Diagram

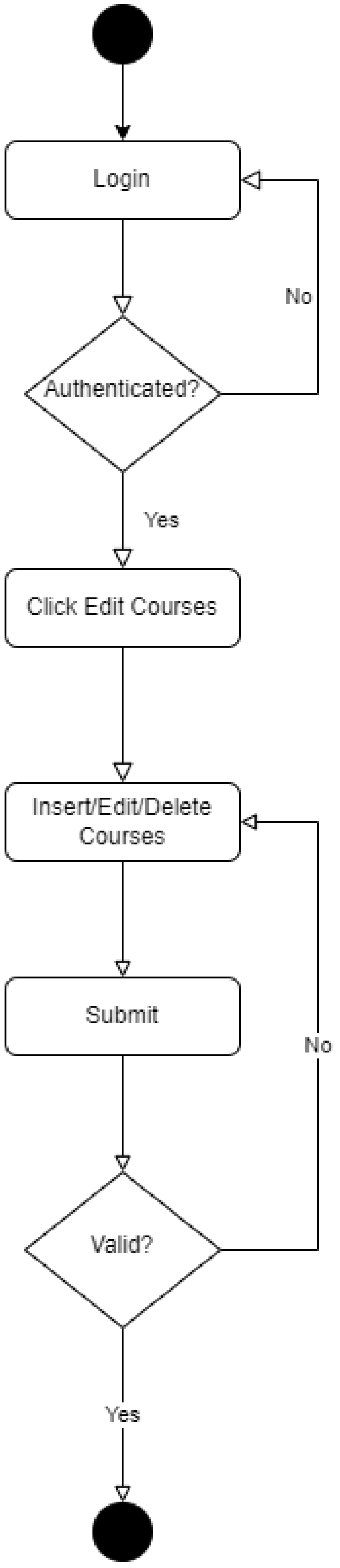
Student use case: Apply for a Course and See your Apply together



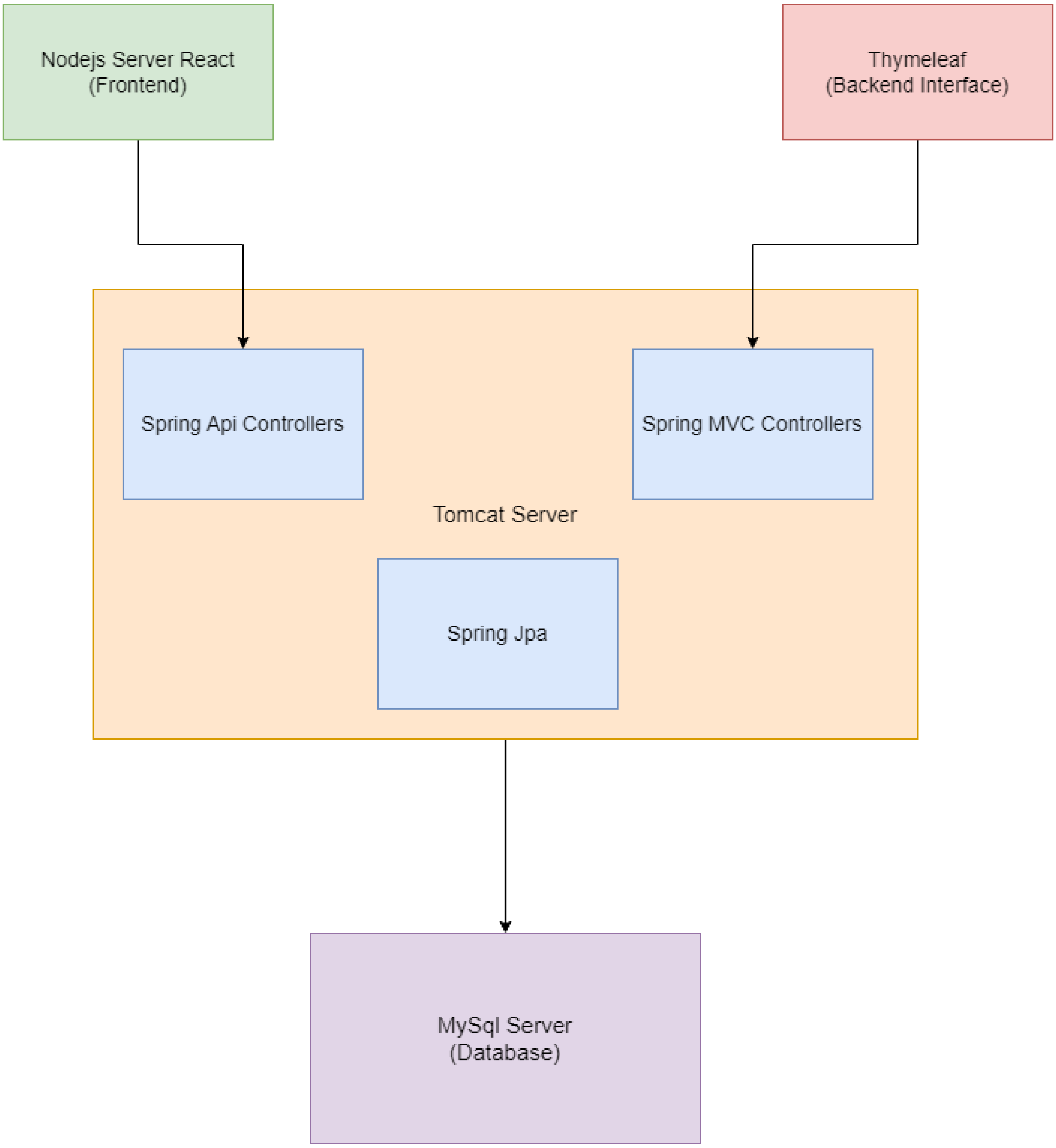
Secretary use case: Evaluate Students Applies



Secretary use case: Insert/Edit/Delete Courses


Login
No
Authenticated?
Yes
Click Edit Courses
Insert/Edit/Delete Courses
Submit
No
Valid?
Yes

Deployment Diagram



Application installation instructions

- **For the Backend:**

  To install the Database run the command via docker:

  ```
  docker run --name mydb -p 3306:3306 -e MYSQL_ROOT_PASSWORD=root -d mysql
  ```

  To create the base used by the application and to create some basic entities (such as admin) to run the application without any problems, run the commands through a sql client:

  ```
  create database mydb;
  use mydb;
  insert into users values(1,true,'admin@admin.com','admin admin','$2a$10$BwC.9HN.8qYF2jWihQzS8uA6VlooXWAwvNBdv4.l4l1UBAdKDEm3.');
  insert into roles values(1,'ROLE_USER');
  insert into roles values(2,'ROLE_STUDENT');
  insert into roles values(3,'ROLE_SECRETARY');
  insert into roles values(4,'ROLE_ADMIN');
  insert into users_roles values(1,1);
  insert into users_roles values(1,4);
  insert into course values(1,'Some details','Programming');
  insert into course values(2,'Some details','A.I.');
  insert into course values(3,'Some details','Web programming');
  ```

  To run the application, execute the command with **java 17 and maven(latest version) installed**:

  1. mvn clean install
  2. java -jar target/<....jar>

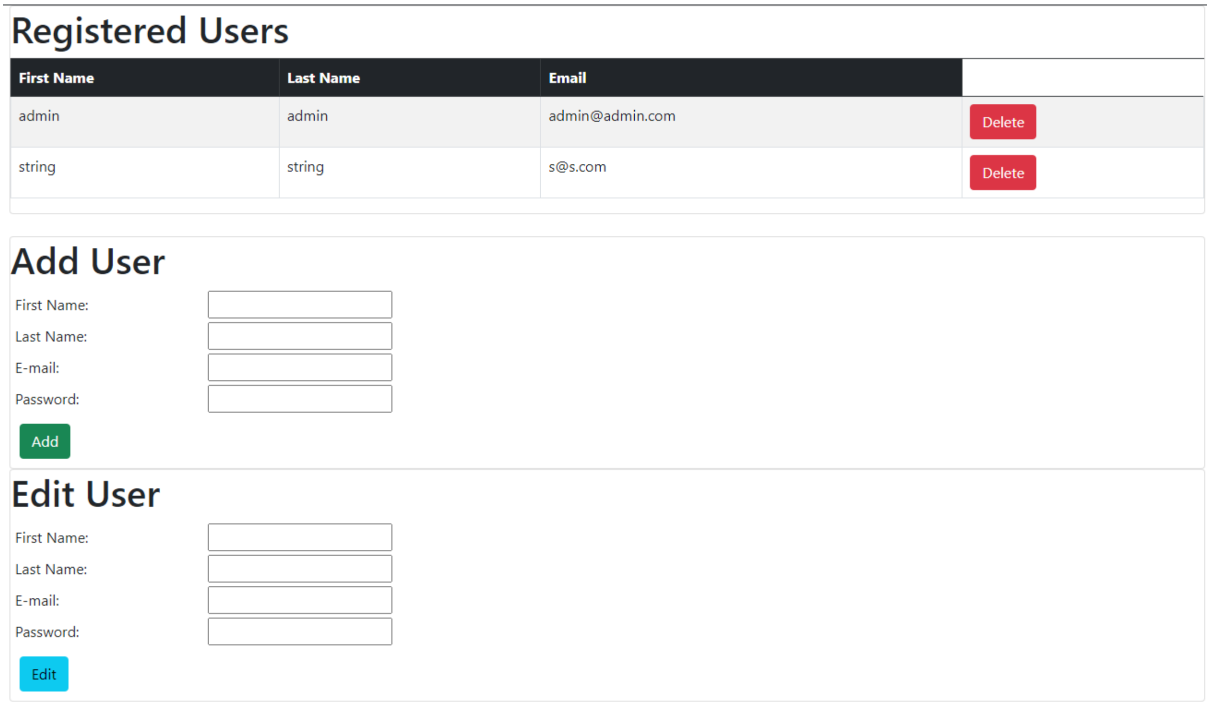
- **For the Frontend:**

  To install the frontend application after you have installed the nodejs server, run the commands:

  1. npm install
  2. npm start

Set of instructions

Backend System to manage users:



Registered Users

First Name	Last Name	Email	
admin	admin	admin@admin.com	Delete
string	string	s@s.com	Delete

Add User

First Name:

Last Name:

E-mail:

Password:

Edit User

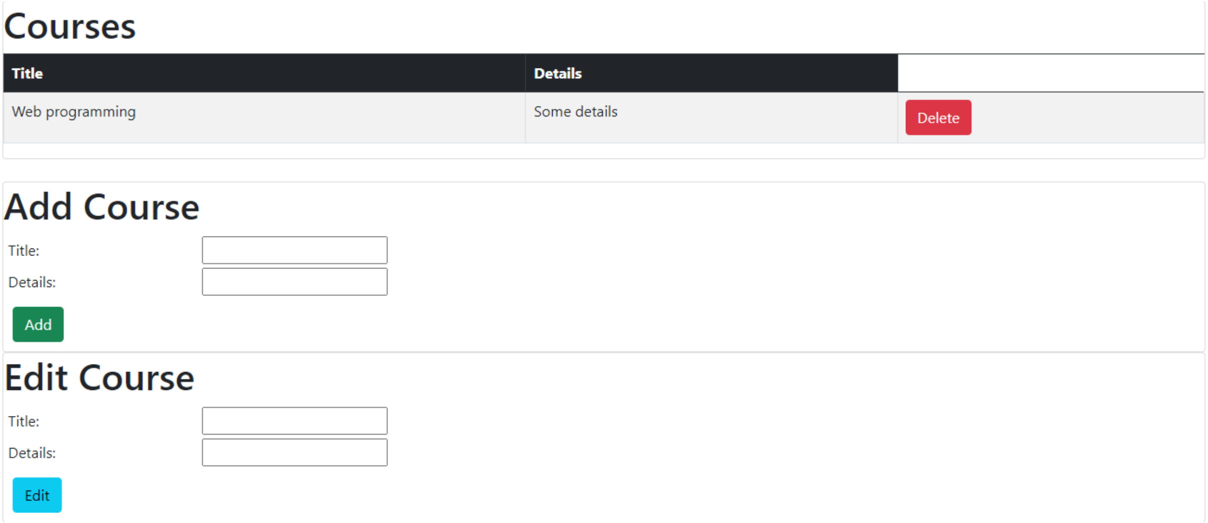
First Name:

Last Name:

E-mail:

Password:

Management of Courses:



Courses

Title	Details	
Web programming	Some details	Delete

Add Course

Title:

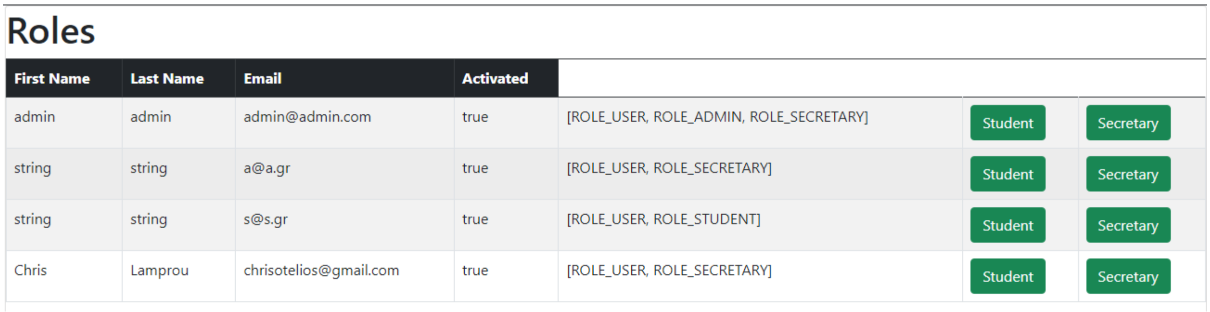
Details:

Edit Course

Title:

Details:

To manage Roles and activate the user (by pressing a role, either student or secretary, the account is also activated):

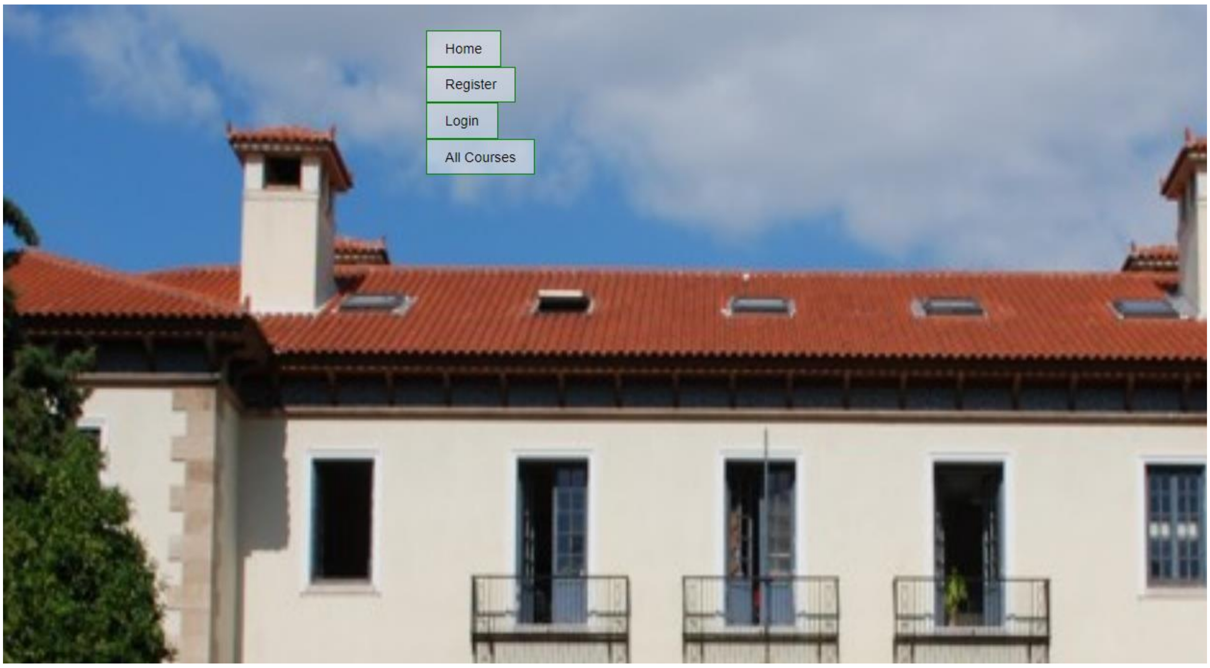


Roles

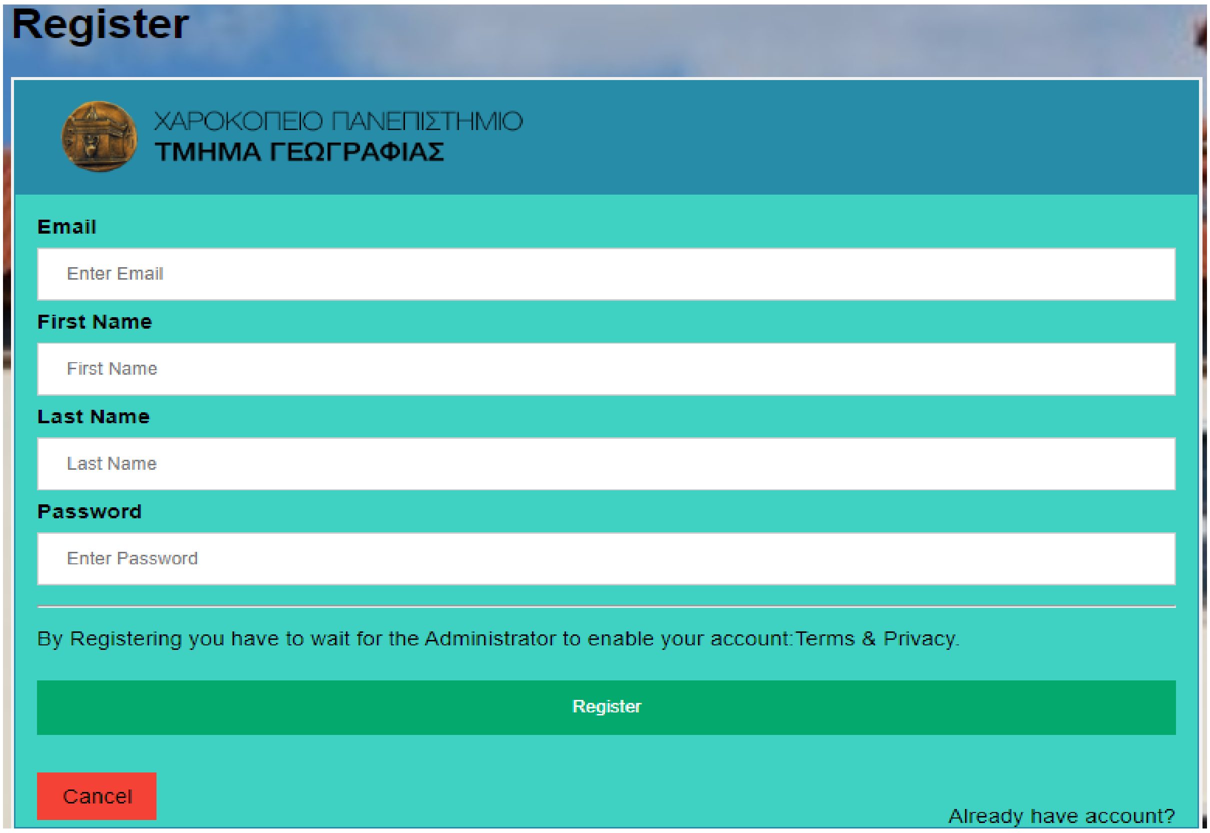
First Name	Last Name	Email	Activated			
admin	admin	admin@admin.com	true	[ROLE_USER, ROLE_ADMIN, ROLE_SECRETARY]	Student	Secretary
string	string	a@a.gr	true	[ROLE_USER, ROLE_SECRETARY]	Student	Secretary
string	string	s@s.gr	true	[ROLE_USER, ROLE_STUDENT]	Student	Secretary
Chris	Lamprou	chrisotelios@gmail.com	true	[ROLE_USER, ROLE_SECRETARY]	Student	Secretary

Frontend system for the management of Students, i.e. their applies:

Home Page:



System registration page:



Register

ΧΑΡΟΚΟΠΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ
ΤΜΗΜΑ ΓΕΩΓΡΑΦΙΑΣ

Email

Enter Email

First Name

First Name

Last Name

Last Name

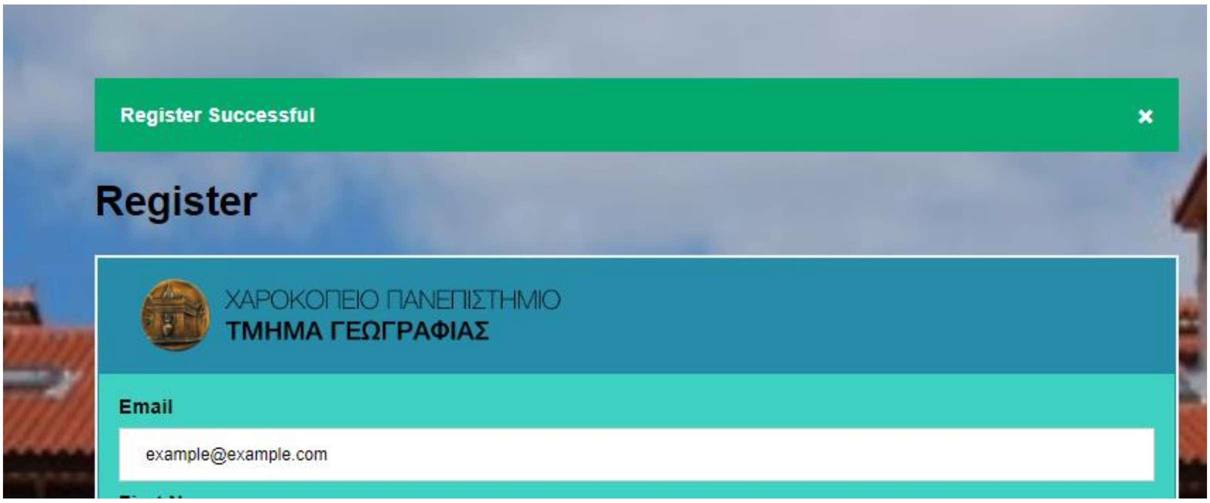
Password

Enter Password

By Registering you have to wait for the Administrator to enable your account:Terms & Privacy.

The Already have account link leads you to login to the system and the cancel button on the home page.

By pressing register successfully:



Register Successful ×

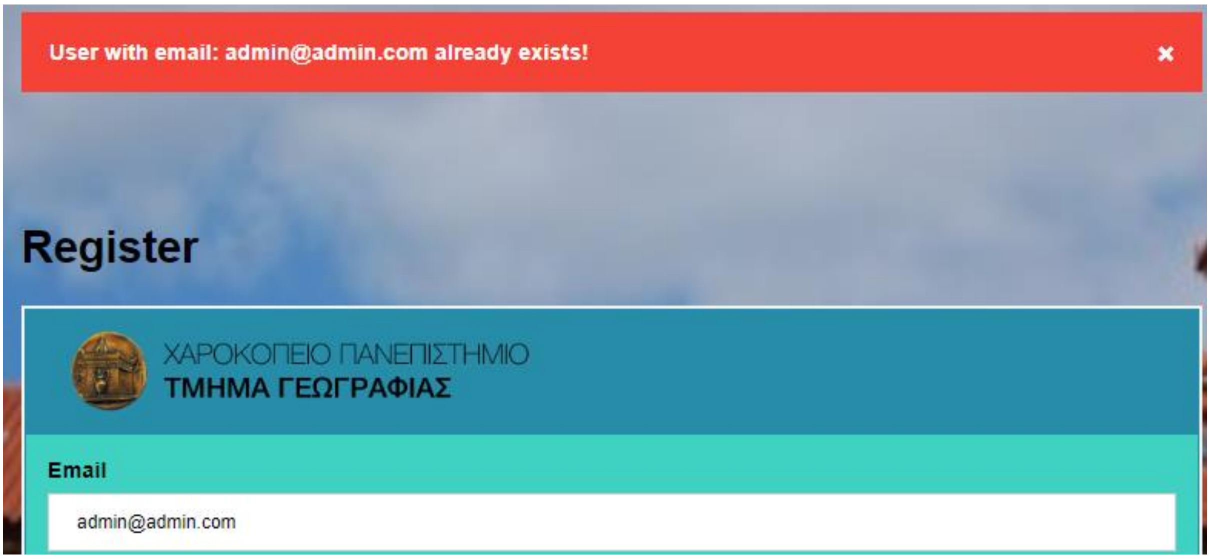
Register

ΧΑΡΟΚΟΠΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ
ΤΜΗΜΑ ΓΕΩΓΡΑΦΙΑΣ

Email

example@example.com

Pressing register with failure:

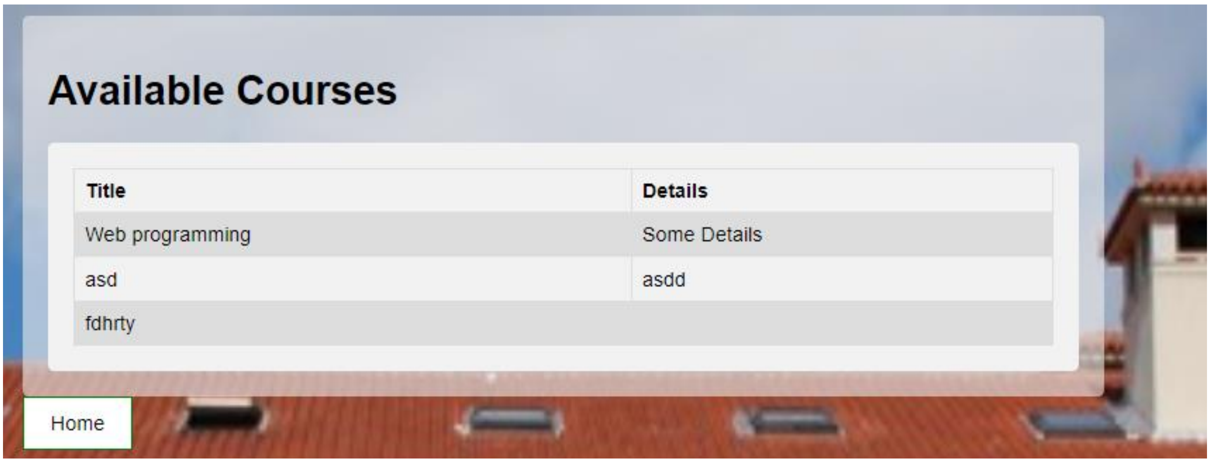


User with email: admin@admin.com already exists!

Register

Email: admin@admin.com

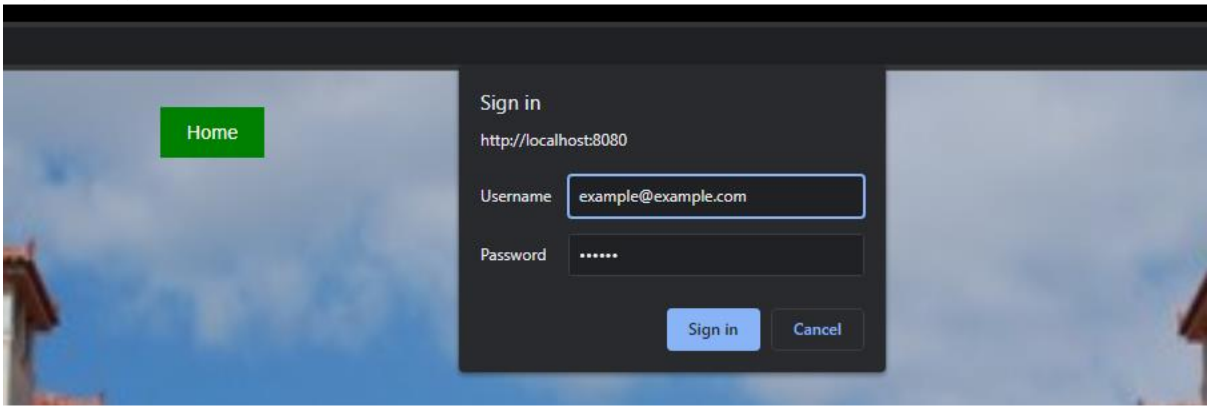
Page with all available Courses and their details:



Available Courses

Title	Details
Web programming	Some Details
asd	asdd
fdhrty	

Login Page:



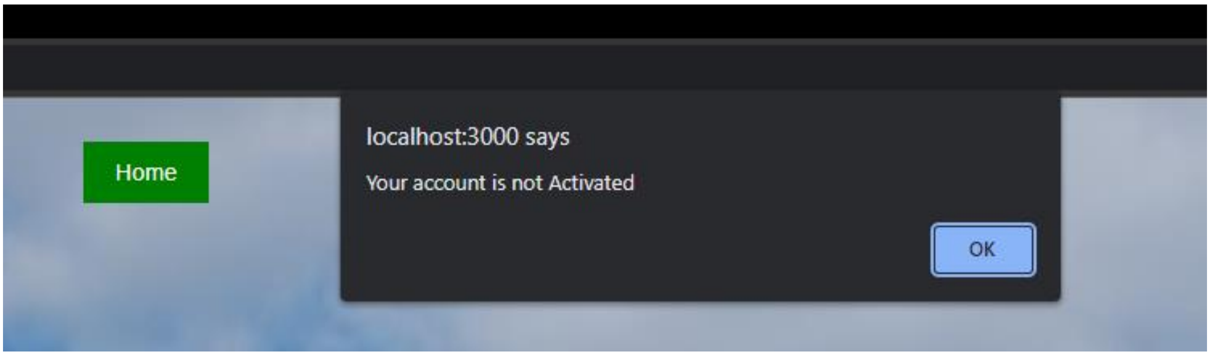
Sign in

http://localhost:8080

Username: example@example.com

Password: ••••••

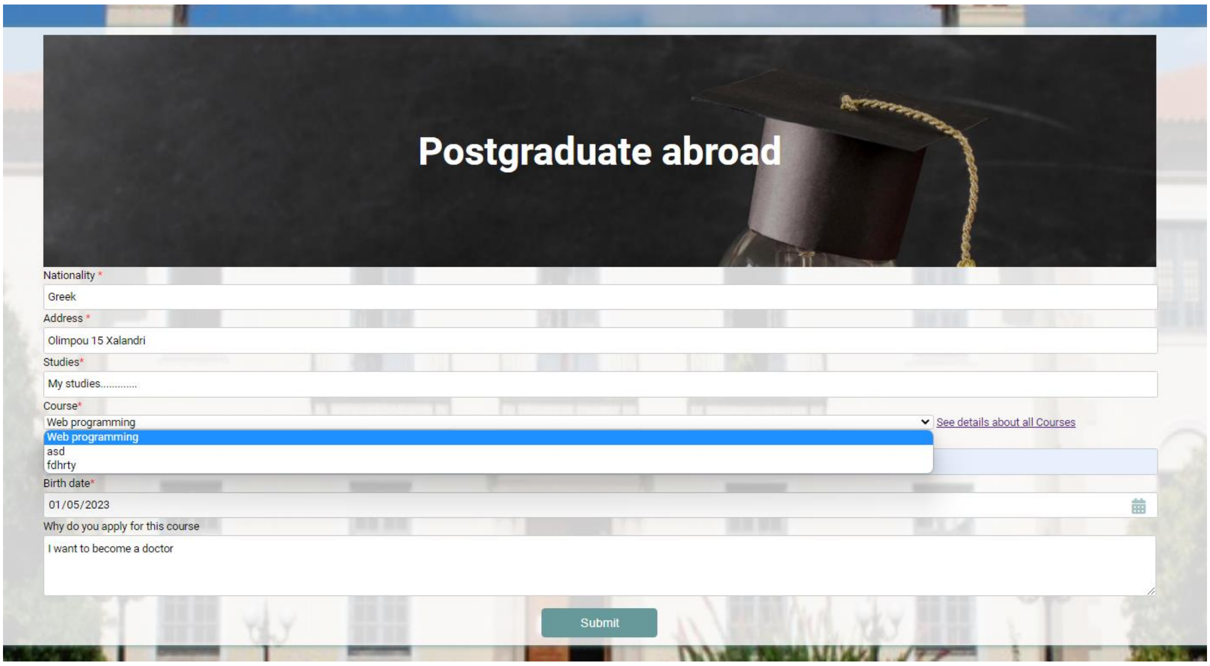
If the user has not been activated by the system administrator:



localhost:3000 says

Your account is not Activated

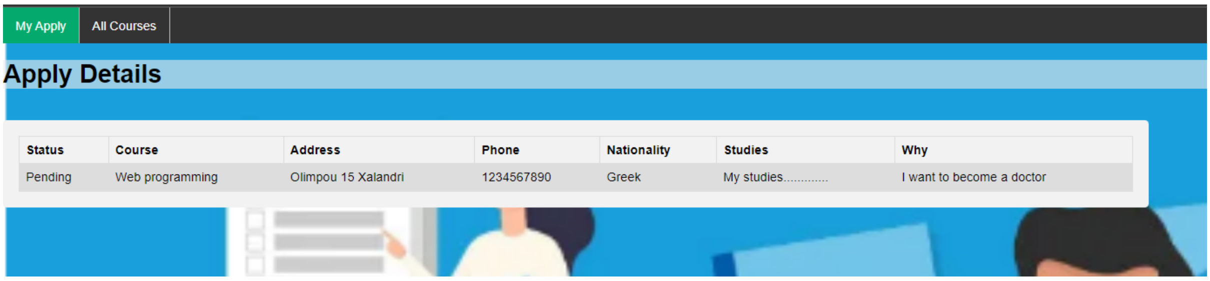
Entry of application details for the recruitment of the student by the secretariat:



Nationality *
Greek
Address *
Olimpou 15 Xalandri
Studies*
My studies............
Course*
Web programming
Web programming
asd
fdhrty
Birth date*
01/05/2023
Why do you apply for this course
I want to become a doctor

Note: the link “see details about all Courses” shows the same thing as the page with all courses and their details.

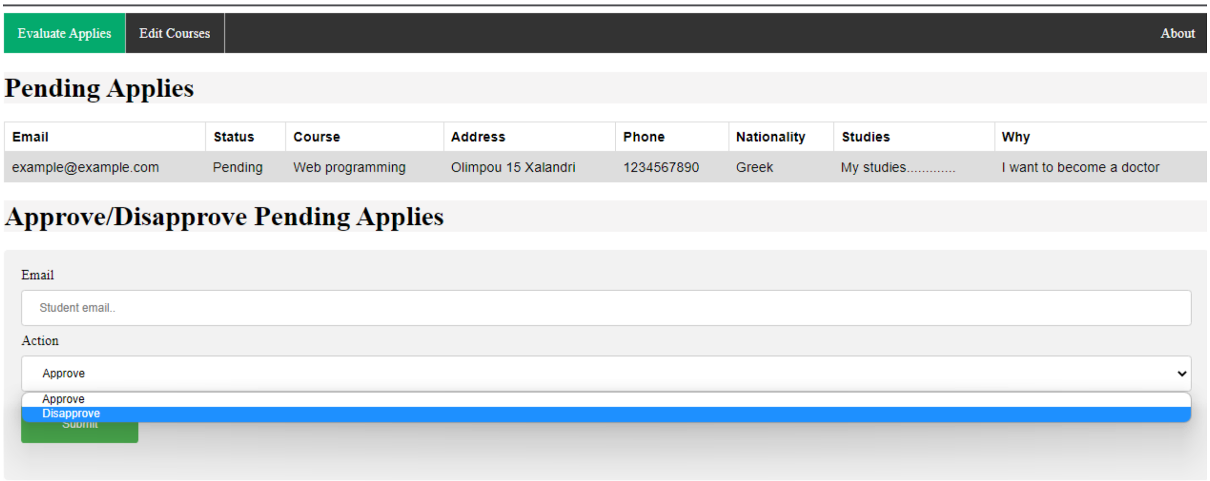
The Main page that the user sees every time he logs into the system (essentially he sees the status of his application):



Apply Details

Status	Course	Address	Phone	Nationality	Studies	Why
Pending	Web programming	Olimpou 15 Xalandri	1234567890	Greek	My studies............	I want to become a doctor

Note: The status as we see is Pending because it was just created, but as soon as the secretariat approves or rejects it, it will change to Approved or Disapproved. The Main page that the secretariat sees every time it connects to the system, where it can see all the Pending applications and by putting an email from those that appear and pressing submit approve or reject them.



Pending Applies

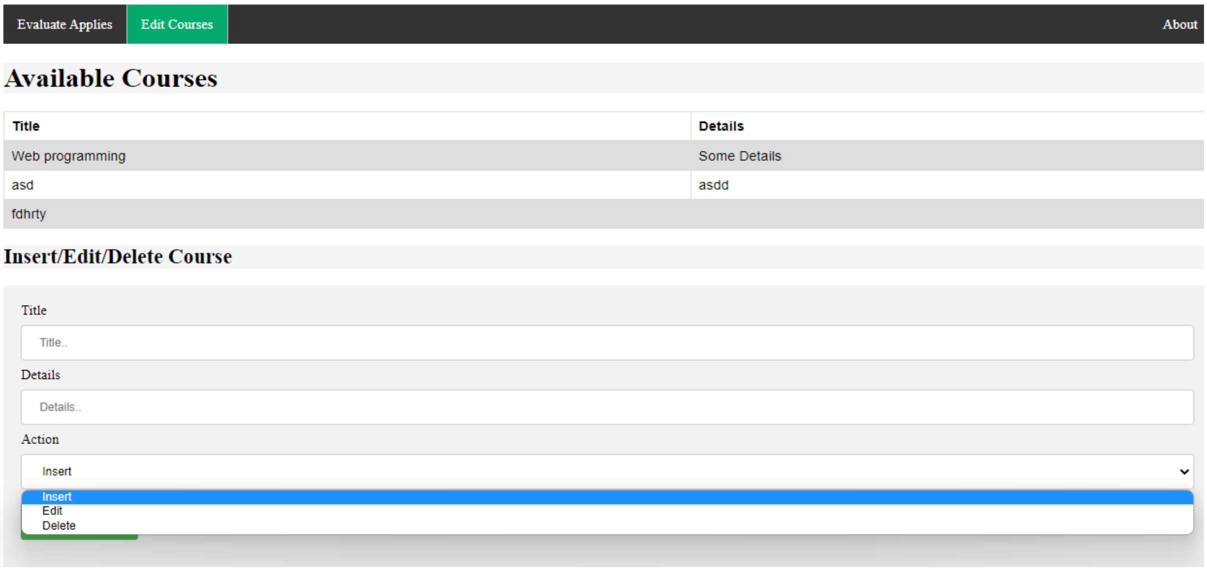
Email	Status	Course	Address	Phone	Nationality	Studies	Why
example@example.com	Pending	Web programming	Olimpou 15 Xalandri	1234567890	Greek	My studies............	I want to become a doctor

Approve/Disapprove Pending Applies

Email: Student email..

Action: Approve

Finally, the page with all the courses that the secretariat can edit:



Available Courses

Title	Details
Web programming	Some Details
asd	asdd
fdhrty	

Insert/Edit/Delete Course

Title: Title..

Details: Details..

Action: Insert

Depending on the action chosen by the secretariat and pressing submit, the courses are updated in the system.