Zewail City of Science, Technology and Innovation
University of Science and Technology
School of Computational Sciences and Artificial Intelligence

CSAI 203 – Fall 2025

Introduction to Software Engineering

Course Registration System

Documentation

Team Number: 23

Team Members:

Hazem Ahmed Refaat – 202401389

Youssof Amgad Abdelrahman – 202400370

Maya Sameh Aboamer – 202401080

Yaseen Mohamed Bilal – 202400463

Representative contact info:

s-hazem.refaat@zewailcity.edu.eg

20/12/2025

Testing Description:

Manual testing:

The non-functional requirements were manually tested.

Long login sessions were tested by logging into a user and closing the browser, after some time reopening the browser and running the app again, the website starts at the home page of the previously logged in user meaning the non-functional requirement was met successfully.

Intuitive UI was tested by asking users who have not seen this website before to perform each functional requirement without being guided through the website, they were able to successfully perform all of them in a short period of time meaning the non-functional requirement was met successfully.

Automated Testing:

Pytest and unittest were used to test 5 functional requirements by mocking certain objects in the system without writing to the database.

GitHub Actions was set up to automatically run the unit test with every push and pull request to make sure the new code did not break old working code.

If all the tests pass, a docker image is automatically built.

User Documentation:

Deployment:

Run these commands in the terminal:

```
1- docker build -t swe-project

2- docker run -p 5000:5000 swe-project
```

Then open the link 'http://localhost:5000'

Login:

- The user needs to be added by the admin.
- An admin needs to log in, choose Manage Users and add a new user.
- Every user type (Admin/Professor/Student) can choose their user type and login with the credentials given to them by the admin.

Student:

- To register for a course as student, login, click register for course and select a course from the available courses, write the course ID and click register course.
- To view grades, click on view grades.
- To view registered courses, click on view registered courses.
- To drop a course, click on drop course, write course ID and click drop course.

Professor:

- To assign a grade, login as professor, click assign grade and enter student and course IDs, select the grade and click assign grade.
- To view registered students, click on view registered students.
- To remove students, click on registered students and select remove on the required student.

Admin:

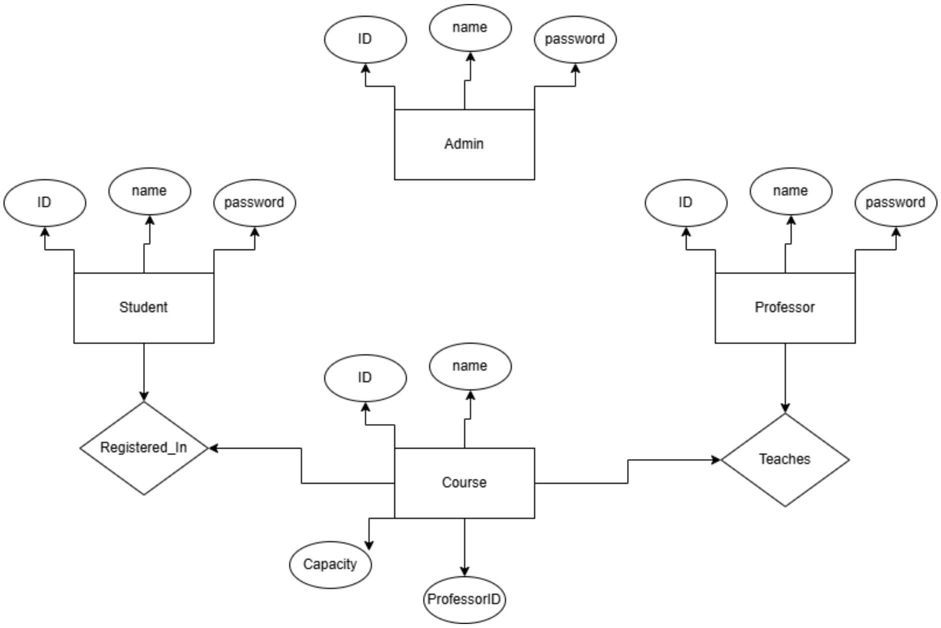
- To Add/Remove courses, login as admin, click add / remove courses, select the course to remove and click remove. To add new course, fill out the bottom form and click add course.
- To assign a professor to a course, course management, assign professor.

Technical Documentation:

Architecture: MVC pattern was used in this project for separation of concerns. Repository pattern was used to isolate data access from models and controllers.

API endpoints: Main routes used are: ("/"), ("/logout"), ("/Slogin"), ("/StudentHome"), ("/Plogin"), ("/ProfessorHome"), ("/Alogin"), ("/AdminHome")

Database Schema:


ID
name
password
Admin
ID
name
password
Student
ID
name
password
Professor
ID
name
Registered_In
Course
Teaches
Capacity
ProfessorID