National University of Computer & Emerging Sciences

Department of Software Engineering

Course: Web Engineering - Spring 2025

Assignment 2

Total Marks	55
Assignment Deadline	20 March, 2025

Instructions:

1. This assignment involves building a university course registration system with real-world functionality. You must provide complete and functional HTML, CSS, JavaScript, Node.js, and MongoDB code.
2. Submissions must be in a ZIP file containing your HTML, CSS, JS, server, and database files.
3. The assignments will be checked through **MOSS**, so avoid plagiarism among each other.
4. Submit your assignment as a single ZIP file named 22F-XXXX_Web_Assignment_2.zip
5. Submissions not following the submission structure of the assignment will receive **deductions.**
6. Directly copying from peers/online sources/AI will result in 80% marks deductions in the assignment for both students.
7. The task is mainly of 50 marks. However, there are **5 bonus marks** written in the end.

8. Submissions **must follow the given file structure:**

```
22F-XXXX_Web_Assignment_2/
|
├── public/                         # Frontend assets
|   ├── css/
|   |   ├──  //your css files here
|   |
|   ├── js/
|   |   ├── common.js                # Shared scripts
|   |   ├── admin/
|   |   |   ├── // your admin js files here
|   |   ├── student/
|   |   |   ├── // your student js files here
|   |   └── auth.js                  # Login and registration scripts
|   |
|   ├── images/                      # Static images
|   |   └── logo.png
|   |
|   └── index.html                   # Entry point (login page)
|
├── views/                           # EJS templates for dynamic pages
|   ├── admin/
|   |   ├── // your admin views here
|   ├── student/
|   |   ├── // your student views here
|   ├── auth/
|   |   ├── login.ejs                # Login form
|   └── error.ejs                    # Error page
|
├── routes/                          # Express route handlers
|   ├── // your routes here
|
├── models/                          # MongoDB schemas
|   ├── // your model files here
|
├── controllers/                     # Application logic
|   ├── // your controllers here
|
├── middleware/                      # Access control and validation
|   ├── // your middlewares here
|
├── config/
|   └── db.js                        # MongoDB connection setup
|
├── .env                             # Environment variables
├── server.js                        # Express server entry point
├── package.json                     # Dependencies and scripts
└── README.md                        # Project overview and setup instructions
```

Problem Statement:

The university administration has approached your development team with an urgent request. Their outdated course registration system is causing significant frustration among students and administrative staff. The Registrar's Office has received numerous complaints highlighting specific issues:

- Students cannot visualize their weekly schedule when selecting courses, leading to time conflicts.
- The system crashes during registration due to excessive page refreshes.
- Students struggle to find courses that meet their degree requirements and fit their schedules.
- Seniors are unable to graduate on time because they miss prerequisites.

Your task is to create a modern, interactive prototype that addresses these critical problems using your web engineering skills. The system must support **two types of users**: Students and Admins, each with their own functionalities. Students can **only log in using their roll numbers**(obviously yours only), and there is no registration feature for students.

Detailed Scenario:

Student Features:

- **Login**:
  Students can log in using their **roll number(yours)**. No registration is required; the roll number must already exist in the database.
- **Scheduling Nightmare**:
  Students need an interactive weekly calendar view that updates dynamically as they add or remove courses. The system must highlight scheduling conflicts in real-time.
- **Registration Refresh Frenzy**:
  The system must display real-time seat availability without requiring page refreshes, reducing server load and improving user experience.
- **Finding the Right Fit**:
  Students need a filtering system to search for courses based on criteria like department, course level, time of day, days of the week, and open seats. Results should update instantly as criteria change.
- **Lost Progress**:
  The system must maintain the state of a student's potential schedule throughout their session, allowing them to build their schedule incrementally.
- **Prerequisite Puzzles**:
  The system must clearly show course prerequisites and relationships to help students plan their courses effectively.

Admin Features:

- **Login**:
  Admins can log in using a username and password.
- **Course Management**:
  Admins can add, update, or delete courses in the system. They can also set prerequisites for courses.

- **Student Management**:
  Admins can view and manage student registrations. They can also override registrations in special cases (e.g., adding a student to a full course).
- **Seat Management**:
  Admins can adjust the number of seats available for a course.
- **Reports**:
  Admins can generate reports, such as:
  - List of students registered for a specific course.
  - List of courses with available seats.
  - List of students who have not completed prerequisites.

Technical Requirements:

Using the technologies covered in class:

- **Frontend:** HTML, CSS, and JavaScript
- **Backend**: Node.js with Express (Use topics studied in class)
- **Database**: MongoDB

Create a single-page application prototype that demonstrates how these problems can be solved through thoughtful web engineering.

Bonus Tasks (5 marks):

- ### Notification Styles for Seat Availability (3 marks):
  A 'Subscribe' button next to each course which allow students to subscribe to a course and get notified when a seat becomes available.
- ### Push Code to GitHub and Update README (2 marks):
  Push your complete project to a GitHub repository and document it by adding the name and link of your GitHub repository in the README.md file.