CMPS 350 Web Development Course Project Phase 1
QU Student Management Application

Weight of the project (phase 1):

- 10% of the course grade.

Important Dates

- The project phase 1 submission is due by **12 AM Friday 15 March 2024**.
- Demos are during the same week.

1. Description of the project:

The CSE department of Qatar University has tasked you with building a web application for the management of its students and courses. The following is a simplified description of the application needed:

- There are three types of users:
  - **Students:** They have a name, ID, list of completed courses along with their grades.
  - **Instructors:** They have a name, and expertise areas.
  - **Department Administrators:** They are responsible for creating courses and classes.

- The application has the following functionalities (i.e. use-cases):
  - **Use Case 1: Login**
    - It allows users to login to use the app using their username and password. Login should be verified using the users data in a JSON file (*users.json*) – you must create that file on your own and add users to it.
    - To keep the app simple, there is no need for the users to register to create an account to use the app.
    - Once a user is logged in, they will be redirected to the main page.
  - **Use Case 2: Search and display available courses**
    - On the main page, the student can search available courses by their names (e.g. JavaScript, OOP, etc.) or category (e.g. Databases, Programming, etc.).
    - By default, the main page displays all offered courses. When the user searches courses by *name* or *category*, only relevant courses should be displayed.
    - To keep the application simple, you can create and use a JSON file for available courses, i.e. *courses.json*.
  - **Use Case 3: Register in a course**
    - The student must be logged in to be able to register for courses.
    - The student can register for a new course only if they have passed successfully all prerequisite courses and if the course is open for registration.
    - If the conditions above are met, the student can register for the course with the desired instructor (if the instructor still has places available in their classes). The application should display an appropriate message if the student cannot register.
    - If the registration is successful, the courses / classes information are updated accordingly (to include the newcomer).
    - The registration will not be effective until the class is approved officially by the administrator.

- **Use Case 4: View their learning path**
  - The student should be able to view the list of all courses they have completed successfully (along with their grades), courses that are in progress and those that are still pending (i.e. the registration is still open, and the course has not started yet).

- **Use Case 5: Creating / validating courses and classes**
  - The administrator when they log on, they should see all courses that are currently in progress and those open for registration.
  - Displayed courses should be distinguished based on their *status* and *category*. If a course has several simultaneous classes, these should be displayed as well.
  - The administrator can officially validate the courses / classes that have received a sufficient number of registrations and cancel those that have not – a class is the materialization of a course with an instructor.
  - A course / class cannot start if not validated by the administrator.
  - The administrator should also be capable of creating new courses and classes.

- **Use Case 6: Grades submission**

  The instructors should be able to view their current classes and to submit the students' final grades.

2. Deliverables and Important notes

- Seek further clarification about the requirements/deliverables during the initial progress meeting with the instructor**.** Note that further important clarifications may be modified/added to the project requirements.
- You must define a project plan (including the distribution/assignment of tasks and deadlines). During the weekly office hours, you are required to present and discuss your design with the instructor and get feedback – Be also mindful that it is a personal project, in the sense that you should make your own design and implementation choices. The instructor will give you only high-level advice and acts as the client asking for the application, and you play the role of the application designer and implementer.
- You must follow the programming styles studied and use the same techniques and libraries covered in the course.

You are required to submit a project report by the deadline. While preparing the report, make sure to include information about (and respect) the following points:

I. **Design the App Web UI and navigation:** Design the app navigation to allow the user to navigate from one page to another in an intuitive and user-friendly way to achieve the app use cases. You may design the UI wireframe (sketch) to decide the UI components and the layout either on paper or use a design tool such as https://www.figma.com

II. For each use case, implement the app UI and navigation using HTML, CSS and JavaScript. The pages should comply with Web user interface design best practices. Also remember that

'*there is elegance in simplicity*'. **Each page should be responsive to support at least 2 layouts one for mobile and another for PC.**

III. For each use case, design and implement the Web API and the server-side data access repositories to read/write the app data from/to the data store. For phase 1, you can read/write to simple JSON files that you need to create and initialize with some sample data. **Note that this phase will be focused only on a fully working client-side and server-side implementation that read/write data in JSON files.**

IV. Application design documentation should include the Entities, Repositories and Web API class diagrams.

V. Document the app testing using screen shots illustrating the results of testing. You must test the functionality of your applications as well as the non-functional properties such as the user experience, the performance, etc. **It is your duty to define the non-functional properties that apply to your application.**

VI. The report must contain details about the contribution of every team member. Every team member should demo their work and answer questions during the demo week.

VII. Push your implementation and documentation to your group GitHub repository as you make progress.

3. Grading rubric

Criteria	Points	Implementation Percentage	Implementation Quality	Your Grade
Design and implement the app Web UI and navigation using HTML, CSS and JavaScript. Including designing the App Web UI and navigation.	50			
Design and implement the Web API and data access repositories to read/write the app data JSON files.	30			
Application modeling (e.g. UML diagrams) to explain the data entities and the functionalities	5			
Testing documentation using screen shots illustrating the testing results.	5			
Team work quality. Contributions of team members - All members should collaborate and contribute equally to the project.	5			
Project report – description of the implemented app, what is implemented, what is missed ..	5			
Total	100			
Plagiarism, outsourcing, free riders	-100			
Delivery behind the deadline	-5			

Important remark: In case of copying and/or plagiarism or not being able to explain or answer questions about the implementation, you lose the whole grade.

*** Criteria for grading the functionality:**

- The functionality is working: you get 70% of the assigned grade.
- The functionality is not working: you lose 40% of assigned grade.
- The functionality is not implemented: you get 0.

- The remaining grade in all cases from above **is assigned to the quality of the implementation**,
- The grades are distributed on the various use cases, when the design/implementation is partial, you get only the grades of designed/implemented use cases.

Code quality criteria, include:
- Use of meaningful identifiers for variables and functions (e.g. using JavaScript naming conventions)
- Pages are responsive
- Clean code: simple and concise code, no redundancy
- Clean implementation without unnecessary files/code
- Use of comments where necessary
- Proper code formatting and indentation.

You lose marks for code duplication, poor/inefficient coding practices, poor naming of identifiers, unclean/untidy submission, and unnecessary complex/poor user interface design.

Important Remark:

[Grades: 100-85]: Will be given only to **fully functional application** with **all the quality criteria cited above met** and the project has excellent **design for the various functionalities**. **The report is professional**.

[Grades: 85-80]: Will be given only **to functional application with most of all the quality criteria cited above met** and the project has good design for the various functionalities. **The report is professional**.

[Grades: 80-75]: 80% of the application functionalities are functional. The project respects partially the quality criteria. **The report is professional** but misses some information.

The grades are not negotiable. We expect that only a small portion (around 15%) of the class will be able to meet the criteria for the grades **[100-85]. You should work hard to and demonstrate the merits of your application to earn those grades.+**