**CMPS 350 Project Phase 1 – Report**

**Education Platform**

**(10% of the course grade)**

**The report must be submitted in Word format only**

|  |  |
| --- | --- |
| **Group Members** | Aly Deyab Abdelsalam Aly (202008905)  Mahmod Amr Mostafa Abdelmawgood (202105318)  **Emails:** aa2008905@student.qu.edu.qa; ma2105318@student.qu.edu.qa |
| **GitHub link** | https://github.com/Mahmod-ma2105318/WebProject |

**Grades :**

**The student fills only the “Implementation Percentage”: Please specify either: *Working (completed x%)*, *Not Working (completed x%)* or *Not done*.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **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 | Working 100% | Good |  |
| Design and implement the Web API and data access repositories to read/write the app data JSON files. | 30 | Working 100%  (Local Storage) | Good |  |
| Application modeling (e.g. UML diagrams) to explain the data entities and the functionalities | 5 | Working 100% | Good |  |
| Testing documentation using screen shots illustrating the testing results. | 5 | Working 100% | Good |  |
| Team work quality. Contributions of team members - All members should collaborate and contribute equally to the project. | 5 | Working 100% | Good |  |
| Project report – description of the implemented app, what is implemented, what is missed .. | 5 | Working 100% | Good |  |
| **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.

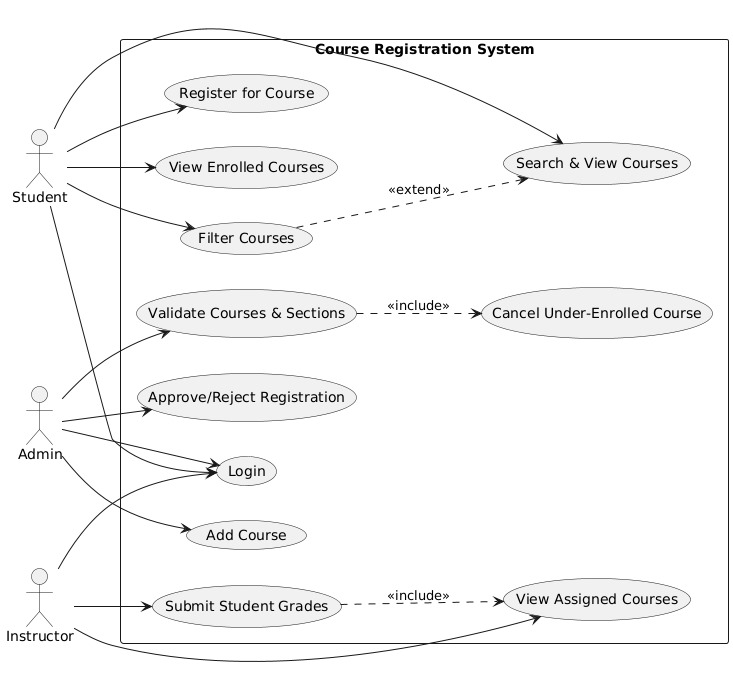
The grades are not negotiable. We expect that only a small portion (around 15%) of the class will be able to meet the criteria for the grades **[100-85]. You should work hard to and demonstrate the merits of your application to earn those grades.+**

# Description of your proposed platform

1. The first page is a login screen where users can log in to the platform.
2. If the user is a student, he will be navigated to the student screen (student.js) where he can find the courses (Registered, current and finished), logout will navigate back to the login screen.
3. If the user is an admin, he will be navigated to an admin screen (admin.js) where he can find the courses (currently taken courses and pending courses) and he can validate courses and add new courses.
4. If the user is an instructor, he will be navigated to an instructor screen (instructor.js) where he can find the courses (courses he is teaching) and he can give grades to students.

# Application Design

# Use case diagram



# Entities class diagram

A diagram of a software company

AI-generated content may be incorrect.

# Web API class

**Local Storage:**

A screenshot of a computer

AI-generated content may be incorrect.

**Json:**

A screen shot of a computer program

AI-generated content may be incorrect.

A screen shot of a computer program

AI-generated content may be incorrect.A screen shot of a computer program

AI-generated content may be incorrect.

A screen shot of a computer program

AI-generated content may be incorrect. A screen shot of a computer program

AI-generated content may be incorrect.

# Implementation

# Implemented use-cases

# Unimplemented use-cases and not functioning parts

# Testing

# Use case 1

# Use case 2

# Use case 3

# Use case 4

# Use case 5

# Discussion of the project contribution of each team member

|  |  |
| --- | --- |
| **Student name** | **Student contributions** |
| Aly Aly | 50% (instructor) |
| Mahmod Abdelmawgood | 50% (login) |
| For Student and admin (we work together where we distribute each use case to two and each one work on 50% of the use case) | |
|  |  |
|  |  |