**CMPS 350 Project Phase 1 – Report**

**Education Platform**

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

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

|  |  |
| --- | --- |
| **Group Members** | Mohammad Hassan (201807929)  Ahmed Mohammed Abu Arrah (202108841)  Omar amdadullah (202007896)  Mostafa Youssef - 202106932  **Emails:**  [**mh1807929@qu.edu.qa**](mailto:mh1807929@qu.edu.qa)  [**aa2108841@student.qu.edu.qa**](mailto:aa2108841@student.qu.edu.qa)  [**oa2007896@student.qu.edu.qa**](mailto:oa2007896@student.qu.edu.qa)  [**my2106932@student.qu.edu.qa**](mailto:my2106932@student.qu.edu.qa) |
| **GitHub link** | <https://github.com/mh1807929/Spring2025-Web-Development-Project.git> |

**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 |  |  |  |
| 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.+**

# Description of your proposed platform

The QU Student Management System is a website for students, instructors and admin management at Qatar University's Computer Science and Engineering department. It's a website through which the QU runs all its student and instructor interactions regarding grades. It's an online convenient method for students to find courses and register to keep track of their grades. Instructors access the website to report and manage courses as well as submit grades. The admin has access and the ability to create/validate courses as well as cancelling them in case not enough students have registered. The website can

# Application Design

# Use case diagram

A diagram of a student management application

AI-generated content may be incorrect.

# Entities class diagram

A diagram of a computer

AI-generated content may be incorrect.

# Web API class

Use Case 1: Login

- handleLogin(): Verifies login using users.json and sets session.

- handleLogout(): Logs out user and resets UI.

- loadData(): Loads users and courses from JSON.

- showMessage(): Displays messages during login/logout.

Use Case 2: Search and Display Available Courses

------------------------------------------------

- applyFilters(): Filters courses based on name, category, and user status.

- displayCourses(courses): Renders filtered courses.

- setupEventListeners(): Binds search bar and filters.

- loadData(): Loads course data initially.

Use Case 3: Register in a Course

--------------------------------

- handleRegistration(e): Handles course registration with checks.

- checkPrerequisites(student, course): Validates if prerequisites are met.

- handleCancelRegistration(e): Cancels pending course registration.

- showMessage(): Displays registration success/failure messages.

- displayCourses(): Disables registration buttons accordingly.

Use Case 4: View Learning Path

------------------------------

- displayLearningPath(): Displays student’s course progress.

- displayPathCourses(courses, status): Renders learning path by status.

- setupLearningPathTabs(): Enables tabbed navigation.

- loadData(): Loads data to display path.

Use Case 5: Create / Validate Courses and Classes

-------------------------------------------------

- setupAdminPanel(): Shows admin panel with options.

- setupAdminTabs(), setupAdminFilters(): Tab/filter setup for admin view.

- loadAdminCourses(): Loads courses for admin processing.

- displayAdminCourses(courses, container): Displays admin course view.

- validateClass(e): Approves class and student registrations.

- cancelClass(e): Cancels class and updates data.

- setupCourseForm(), addClassForm(): Initializes course/class creation form.

- createNewCourse(): Creates a new course and classes.

- loadDataCourses(): Loads raw course data from JSON.

- renderCoursesForPublishing(draftCourses): Displays draft courses.

- publishSelectedCourses(d): Publishes selected courses.

Use Case 6: Grade Submission

----------------------------

- setupInstructorView(): Sets up instructor dashboard.

- loadInstructorClasses(): Loads instructor classes pending grading.

- submitGrade(e): Submits student grades and updates data.

- loadData(), displayLearningPath(), applyFilters(): Refreshes views post-grading.

# Implementation

# Implemented use-cases

1, 2, 3 ,4 ,5 ,6 ,7 ,8

# Unimplemented use-cases and not functioning parts

---

# Testing

# Use case 1:

A screen shot of a login

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

# Use case 2:

A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

# Use case 3

A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.

# Use case 4

A screenshot of a computer

AI-generated content may be incorrect.

# Use case 5

A screenshot of a chat

AI-generated content may be incorrect.

# Use case 6

A screenshot of a computer

AI-generated content may be incorrect.

# Use case 7

A screenshot of a computer

AI-generated content may be incorrect.

A white rectangular object with a black border

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.A white rectangular object with a black border

AI-generated content may be incorrect.

# Use case 8

A screenshot of a computer

AI-generated content may be incorrect.

# Discussion of the project contribution of each team member

|  |  |
| --- | --- |
| **Student name** | **Student contributions** |
| Mohammad Hassan | * UC5 & UC6 * Report [Testing] |
| Ahmed Mohammed Abu Arrah | * UC3 & UC4 * Report [Project Description] |
| Omar amdadullah | * UC1 & UC2 * Report [Application Design] |
| Mostafa Youssef | * UC7 & UC8 * Report [Application Design] |