**Web Development Project Phase 1**

**CMPS 350**

**Members:**

* **Lama Aldosari 202103338**
* **Nagham Alajmi 202002078**
* **Shaikha AlAli 202001976**
* **Maha kadkhodaei 202003044**

**Section : L51**

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

**Application Description:**

This is a simple web application for managing students, courses, instructors, and classes. It includes basic features like login, course search, registration, grade submission, and weekly schedule. The app was built using **HTML , CSS, and JavaScript**, and it uses **JSON files** to store and manage data.

**App UI and Navigation Design:**

The application follows a simple, clean, and responsive design. Navigation is role-based:

* **Login Page**: Users enter credentials.
* **Student Dashboard**: Search courses, register, and view progress.
* **Instructor Dashboard**: View classes and submit grades.
* **Admin Dashboard**: Manage courses, classes, schedule, and instructor-course assignments.

Each page includes clear buttons, minimal forms, and responsive layout using media queries for mobile and desktop.

**Use Cases Implementation**

**Use Case 1 : Login**

* Implemented login with credentials validated against users.json
* Redirects based on role after login.

**Use Case 2 : Course Search**

* Lists all available courses from courses.json
* Allows search by course name and category

**Use Case 3 : Course Registration**

* Students can register for open courses if prerequisites are met
* Registration checks instructor availability
* Adds student to class if successful

**Use Case 4 : View Learning Path**

* Displays three sections: completed courses, in-progress courses, and pending registrations.

**Use Case 5 : Create/Validate Courses and Classes**

* Admin can view all open and in-progress courses.
* Can validate or cancel classes based on registration numbers.
* Allows creation of new courses/classes.

**Use Case 6 : Grade Submission**

* Instructors can view their classes and submit grades
* Grades are updated in students.json

**Use Case 7 : Instructor-Course Assignment**

* Admin marks which courses are open for interest
* Instructors express interest in courses
* Admin assigns instructor based on submissions

**Use Case 8 : Weekly Schedule Display**

* Admin views a weekly table schedule of all in-progress classes.
* Uses schedule.json to render by day and time.

**Application Design Document**

**Entities:**

* **User**: ID, name, password, role, completedCourses
* **Course**: ID, title, category, prerequisites, status, classes
* **Class**: ID, courseId, instructorId, students, schedule

**Repositories:**

* users.json, courses.json, classes.json, schedule.json

**UML:**

**A diagram of a computer program

Description automatically generated with medium confidence**

**Screenshots:**

**Login Page:**

**A screenshot of a computer

Description automatically generated**

**Search and Display Available Course (main page) :**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**Register for a Course (student page) if student has passed all prerequisites:**

**A screenshot of a computer

Description automatically generated**

**If student has not passed all prerequisites:**

**A screenshot of a student registration form

Description automatically generated**

**View Learning Path :**

**A screenshot of a computer

Description automatically generated**

**Instructor Login:**

**A login screen with a building in the background

Description automatically generated**

**Instructor expressing their interests:**

**A screenshot of a computer

AI-generated content may be incorrect.**

**Instructor submitting students grade for non-existing student:A screenshot of a math test

Description automatically generated**

**Instructor submitting grades for students who exist:A screenshot of a computer

AI-generated content may be incorrect.Logging in as an admin:A login screen shot of a building

Description automatically generated**

**Adding a new course:**

**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.**

**Adding a class for the course:**

**A screenshot of a login form

AI-generated content may be incorrect.**

**A screenshot of a computer

AI-generated content may be incorrect.**

**Validating Classes (status change):**

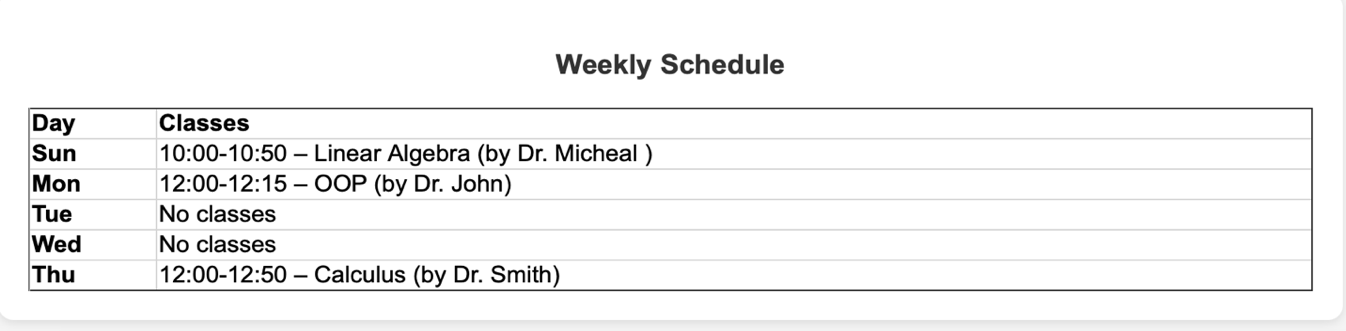
**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

AI-generated content may be incorrect.**

**Updated schedule:**

****

**Team Contribution :**

* **Lama Aldosari**: Developed admin interface for course/class creation, implemented class status management (in-progress, cancelled), and designed and tested the grade submission workflow for instructors.
* **Nagham Alajmi**: created JSON files, produced the login page that redirects to the particular users, created the main page, added to the student.html (course registration), also added to the student.html to view their learning path, worked on the instructor.html to grade students and express intrests.
* **Maha kadkhodaei**: implemented student course registeration with real-time avalibility checks
* **Shaikha AlAli**: Built instructor-course matching and weekly schedule functionality.

All members contributed to debugging, styling, and data testing.