**CMPS 350 Project Phase 1 – Report**

**Education Platform**

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

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

|  |  |
| --- | --- |
| **Group Members** | Abdullah Jamali 202104080  Mohammed Elnajjar 202205158  Faisal Elbadri 202107288  **Emails:** aj2104080@student.qu.edu.qa; me2205158@student.qu.edu.qa; fm2107288@student.qu.edu.qa; |
| **GitHub link** | https://github.com/mhmdelnajjar/ProjectWeb |

**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 | 49 |  |  |
| Design and implement the Web API and data access repositories to read/write the app data JSON files. | 30 | 30 |  |  |
| Application modeling (e.g. UML diagrams) to explain the data entities and the functionalities | 5 | 5 |  |  |
| Testing documentation using screen shots illustrating the testing results. | 5 | 5 |  |  |
| Team work quality. Contributions of team members - All members should collaborate and contribute equally to the project. | 5 | 5 |  |  |
| Project report – description of the implemented app, what is implemented, what is missed .. | 5 | 5 |  |  |
| **Total** | 100 | 99 |  |  |
| **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

Our Education Platform is a web-based application that supports three user roles: **Admin**, **Instructor**, and **Student**. It is built using HTML, CSS, and JavaScript, and it relies on **JSON files** for data storage and interaction, with all data operations handled through the browser's **localStorage**.

**Data Initialization**

* Upon loading the login page, two JSON files (users.json and courses.json) are loaded into the browser's **localStorage**.
* After a successful login, the current user is also stored in localStorage, enabling role-based access and personalized data rendering.

**Student Functionalities**

* View all available classes.
* Register for classes **if prerequisites are met**.
* View a categorized list of:
  + **Courses in progress**
  + **Completed courses**
  + **Pending courses** (awaiting admin approval)
* Follow a personalized **learning path** that updates upon course completion.

**Instructor Functionalities**

* View assigned classes and course details.
* Manage enrolled students in each class.
* Submit grades for students registered under their courses.
* Track the progress and performance of students.

**Admin Functionalities**

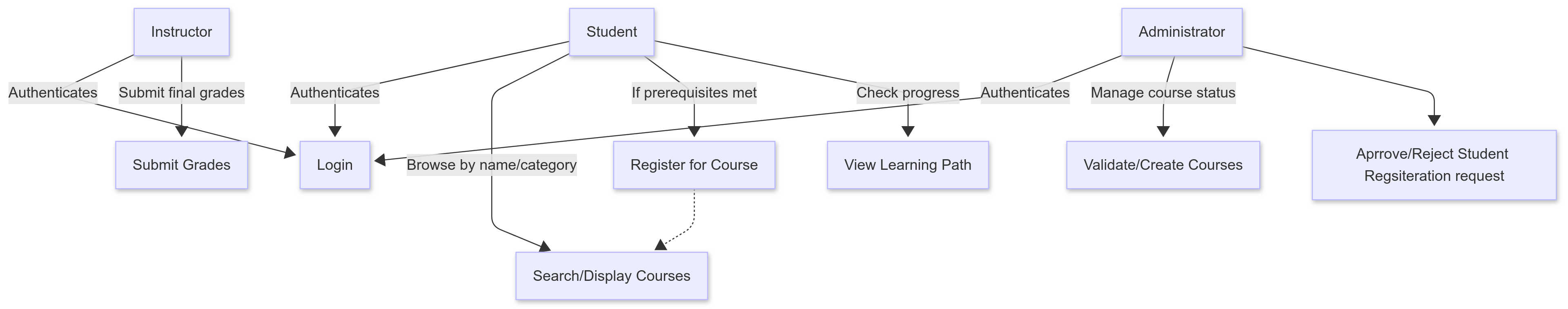
* **Approve or reject** student course registrations.
* **Create**, **edit**, or **delete** course offerings.
* Manage and update course details.
* Monitor student enrollments and class activities.

**Summary**

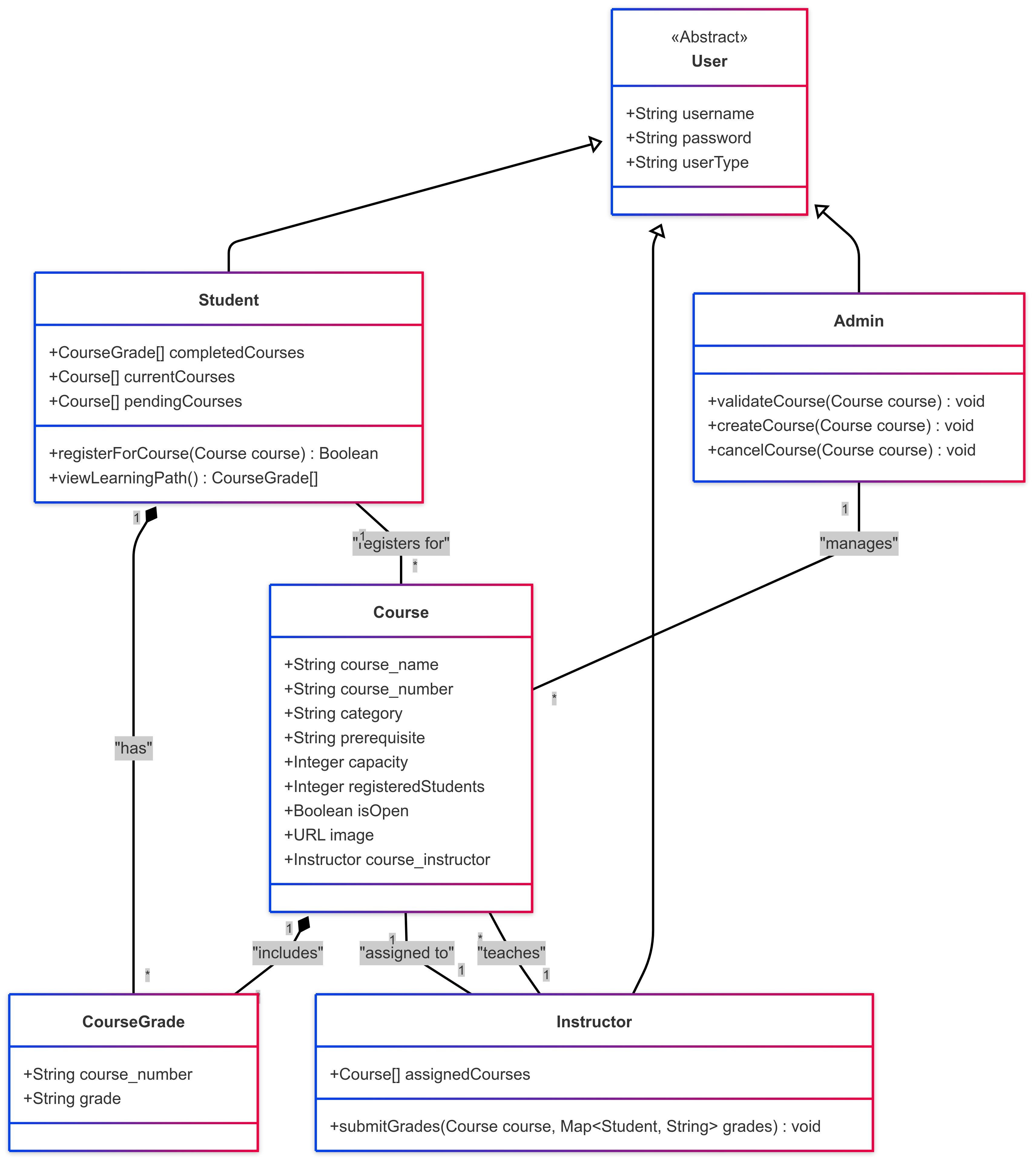
The platform ensures a **role-based experience** with dynamic data handling using localStorage. This approach simulates real-time database behavior while maintaining full functionality within the browser. Each user role is provided with features tailored to their needs, resulting in a flexible, responsive, and educationally productive web application

# Application Design

# Use case diagram



# Entities class diagram



# Web API class

We did not use the web api endpoint we will leave it for phase 2, However getting the data is done by a method called load which will load the users, courses from their json and upload and store it in the localstroge and then update the localstorge each time.

# Implementation

# Implemented use-cases

Our current implementation covers the majority of the core use cases defined in the project requirements. The following features have been fully implemented and tested:

* **Use Case 1: Login**  
  Users (Admin, Instructor, Student) can log in using a username and password verified against the users.json file. Upon successful login, the current user's data is stored in localStorage, and they are redirected to a personalized dashboard based on their role.
* **Use Case 2: Search and Display Available Courses**  
  Students can view all available courses on the main page by default and use a search bar to filter them by name or category. Courses are dynamically fetched from courses.json.
* **Use Case 3: Register in a Course**  
  Students can register for courses if prerequisites are met and seats are available. The system checks the student’s completed courses and validates eligibility. Successful registration is marked as “pending” until approved by an admin.
* **Use Case 4: View Learning Path**  
  Students can view a categorized display of courses: completed (with grades), in progress, and pending. This helps them understand their progress and what remains.
* **Use Case 5: Creating / Validating Courses and Classes**  
  Admins can create new courses and classes by filling out a form. They can also view pending classes, approve or reject them, and manage existing courses.
* **Use Case 6: Grades Submission**  
  Instructors can view their current classes and submit final grades for students who are enrolled.

# Unimplemented use-cases and not functioning parts

Everything working and tested.

# Testing

# Use case 1

Login: Once a user is logged in, they will be redirected to “their” main page.

Login Screen:

A screenshot of a computer

AI-generated content may be incorrect.

\*Admin after he logs in:

A screenshot of a computer

AI-generated content may be incorrect.

Instructor After he logs in:

A screenshot of a computer

AI-generated content may be incorrect.

Student after he logs in:

A screenshot of a computer

AI-generated content may be incorrect.

# Use case 2

Here in use case 2 as we can see we can search for classes, when we search “pr” the programming class shows : A screenshot of a computer

AI-generated content may be incorrect.

And as seen before, when student logged in , cards that show are the ones that are available for him only or the one he finished…..

# Use case 3

A screenshot of a computer

AI-generated content may be incorrect.

A pink card with black and yellow text and a black rectangle with a black rectangle with a red button

AI-generated content may be incorrect.

Now when I click on register, then it will be like this, now the admin should accept it to be registered

And now when we logout and go to the admin side, this will appear then he can approve the class

A pink card with black text and blue and yellow logo

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

AI-generated content may be incorrect.

And now when the student log back in this card will appear (the class will be registered) as shown above

# Use case 4

Learning path:

A screenshot of a computer

AI-generated content may be incorrect.

When I click course completed this will show:





And here is the current courses as shown above, and now when I click pending nothing will show cuz there is no pending course:

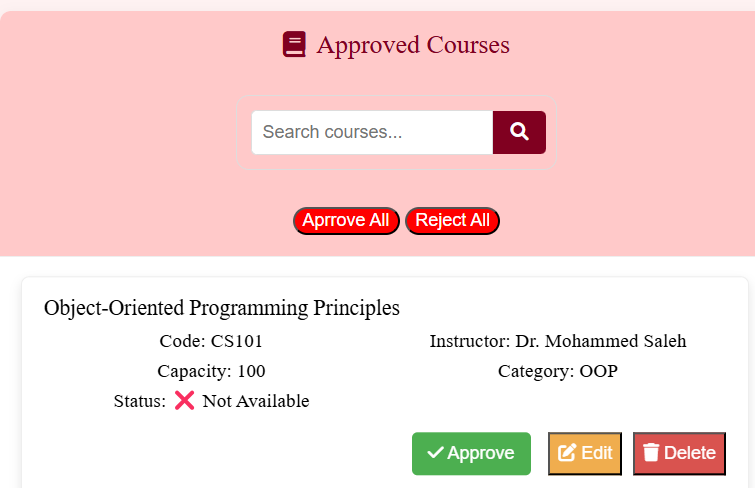
A screenshot of a computer

AI-generated content may be incorrect.

# Use case 5

When admin logs in, he can see the courses and approve what he wants

\*Before:



After :

A screenshot of a computer

AI-generated content may be incorrect.

Now it will appear in the student side as shown:

A screenshot of a computer

AI-generated content may be incorrect.

He can create classes :

A screenshot of a computer screen

AI-generated content may be incorrect.

When create classes clicked this window appears:

A screenshot of a computer

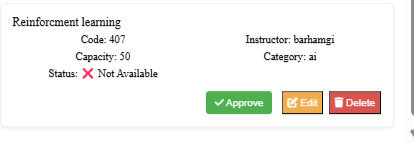
AI-generated content may be incorrect.

Here after filling:

A screenshot of a computer

AI-generated content may be incorrect.

Now we have it , we can edit and delete and here when we click edit , the form will appear with its information:



A screenshot of a computer

AI-generated content may be incorrect.

**And when deleted it will disappear…**

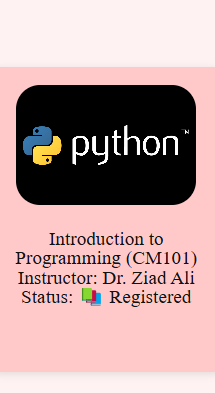
**But here is a picture of the class when approved and added :**

**A screenshot of a computer

AI-generated content may be incorrect.**

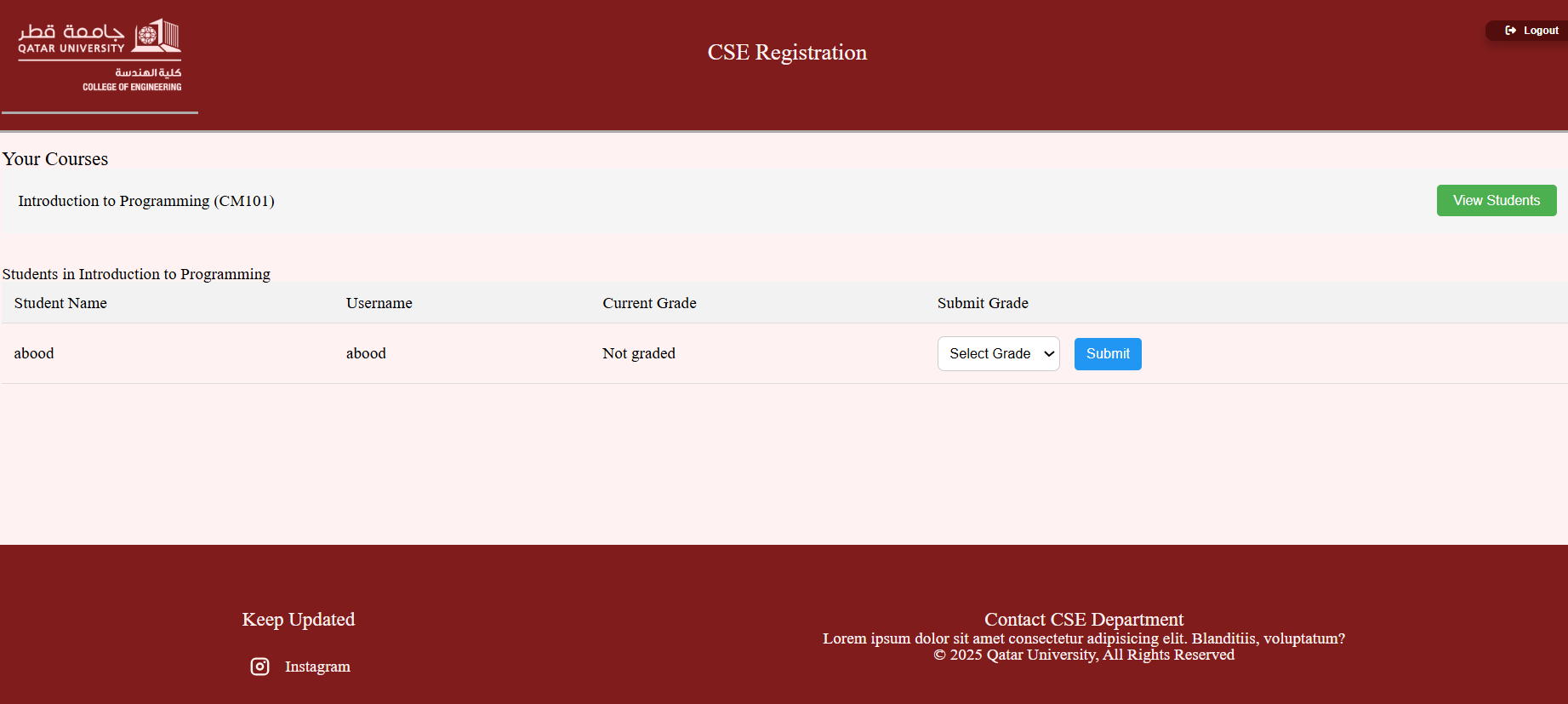
# Use case 6 : submit grades:

When student register for specific instructor, the instructor will have this student name and then can submit his grades as shown , then it will appear on the student side:



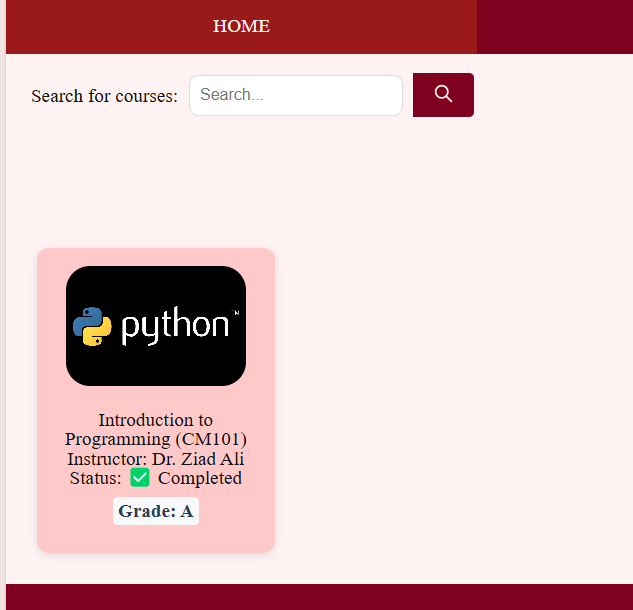
Now dr ziad ali should have aboods name since he registered to his class

Now he can choose one of the grades



**Lets say he chose A**

**And now this how the student class with ziad ali will look like when we go to the courses completed**

****

# Discussion of the project contribution of each team member

|  |  |
| --- | --- |
| **Student name** | **Student contributions** |
| Abdullah Jamali | 35% |
| Mohammed | 35% |
| Faisal | 30% |
|  |  |
|  |  |