E-Learning Platform

**Name:** Ayush Katoch **Supervisor:** Mr. Henry Forbes

**Student ID:** 2016XXXXX

# Project Description

Computer science is one of the major fastest-growing fields in the industry and many students are opting for this course for better employment opportunities. In the universities, it’s very common to teach programming to Computer Science students and also to students from non-technical courses. Specially for students from non-technical background it’s really challenging to complete programming assignment as they first have to download the assignment, code editor / IDE, compilers for the programming language that is to be used, additional libraries for that programming language before they can begin coding. Students often face a lot of problems in setting the development environment due to different versions or operating system. Not only this, when they complete their assignment, they have to upload the files so that they can be graded. For the professors, it’s a tiring process to grade these assignments, as they too will first have to setup the development environment which might not work with all the assignments due to different versions of software used by the students. After setting up the development environment professors have to download the student code and run them individually which is a very time-consuming process.

In order to solve this issue, this e-learning software will be developed that will allow the professors to upload assignment, lecture videos, slides, notes and much more for the student. This software is specially designed for the programming courses as it allows students to do their programming assignments directly on the website without downloading any extra software and they can even test their code along with the correctness using predefined test cases (designed by the professor). This software also reduces the workload of the professors as it can automatically grade the students code using predefined parameters like code compiles, code runs without any errors, testing student code against predefined test cases and hidden cases.

# Key Literature and Background Reading

In order to understand the features that should be implemented, a research on Computing Pedagogy was read. Pedagogy is basically a way of teaching a subject. Computing Pedagogy means effective ways to teach computing to people. It involves 12 principles, Lead with concepts, Work together, Get hands-on, unplug unpack and repack, model everything, foster program comprehension, create projects, add variety, challenge misconceptions, make concrete, structure lessons lastly, read & explore code first. This software could help with the three principles of Computing Pedagogy,

* Lead with concepts: This site can be used to upload video lectures and notes that can be used by the students to better understand the concepts.
* Work Together: Students can work together on a programming assignment using this software
* Get hands-on: This software could be the best way to practice programming, as it allows students to start coding without having to install / setup any 3rd party software.

# Aims & Objectives

Main aim of this project is to educate programming in a classroom. It also helps to achieve the following aims

* For students
  + Providing a platform / tool to the students to write and test their code without having to setup any development environment
  + Providing real time scores on submitted code.
  + Easy access to all the uploaded lecture notes and videos
* For Professors
  + Easy way to monitor class performance using visual tools like graphs
  + Automatic grading of programming assignments using predefined parameters
  + Plagiarism checker to ensure that academic integrity is maintained

In order to achieve the above aims, following objectives needs to be completed.

Objectives that are to be achieved are as follows:

1. Account Management System
   1. Allowing users to create a new account and log in to their existing account.
   2. There would be two types of users in the software, Teacher and Student. Both the account type will have different abilities / functionalities.
   3. On successful login, an authorization token will be generated that needs to be passed in the headers by the user in order to authenticate themselves.
2. Course Management System
   1. Allows professors to create, modify and delete existing courses (courses that they own)
   2. This system, allows students to join an existing code using a course invite token which is given to a student by the professors.
3. Module System
   1. This allows professors to upload lecture videos, notes and slides and also modify and delete the existing modules
   2. Students can use this to download / view the modules
4. Assignment System
   1. Allows professors to upload an assignment which can be of report type, programming type or a combination of both.
   2. Students can use this to submit their assignments. They can also code and test their written code on the website itself.
5. Grading System
   1. Allows professors to grade student assignments using a predefined rubric cube for marking.
   2. Students will be able to see their grades
6. Plagiarism Checker
   1. Checks for plagiarism in the submitted reports and the code
7. Code Executor
   1. Runs user code in the backend, maintains the virtual environment and the installed libraries.

# Development & Implementation

Multiple languages are being used to develop this software. The entire project is divided into two parts, Frontend and Backend and both the parts are using different programming languages.

**Technologies Used**

1. Backend
   1. Python (Django framework)
   2. PostgreSQL (Database)
2. Frontend
   1. JavaScript (ReactJS)
   2. Tailwind CSS (CSS Library)

**Backend**

It is much easier to develop a RestAPI using Python. Python also have thousands of libraries available which makes the development process very smooth. It is widely used in all fields of programming from Web development to Game development and even Machine Learning & AI making it one of the best programming languages. It also has a huge community which makes it easier to get help if there are any issues during the development of project.

Django is a Python framework used to write a web server. It has its own ORM that makes creating / modifying tables easier and enables querying the database.

PostgreSQL is a relational database management system (RDBMS). It’s highly scalable and quick making it one of the best options for both small-scale and large-scale projects. Compared to other databases like SQLite, PostgreSQL won’t slow down when populated with huge amount of data.

**Frontend**

Similar to Django, ReactJS is one of the most used JavaScript libraries for developing web apps. It was developed by Facebook and is being widely used in many websites like Twitter, Instagram, Facebook and many more. One of the main principles of ReactJS is reusability and it achieves that by using components which significantly reduces the number of lines of code that has to be written and allows reusing the same component in different section of the website without having to code it again.

In order to style the website and make it responsive, Tailwind CSS is being used. Alternatively, Bootstrap could have been used as well but unlike bootstrap, Tailwind CSS makes the website look very neat and modern without having to significantly modify the UI.

**Implementation**

In order to implement this project, first the backend will be developed. Backend development would involve creating a database. It also involves coding a RestAPI to handle different HTTP requests like creating/updating/modifying/deleting different resources example, courses. Backend have two major key component called, Code Runner and Plagiarism Checker

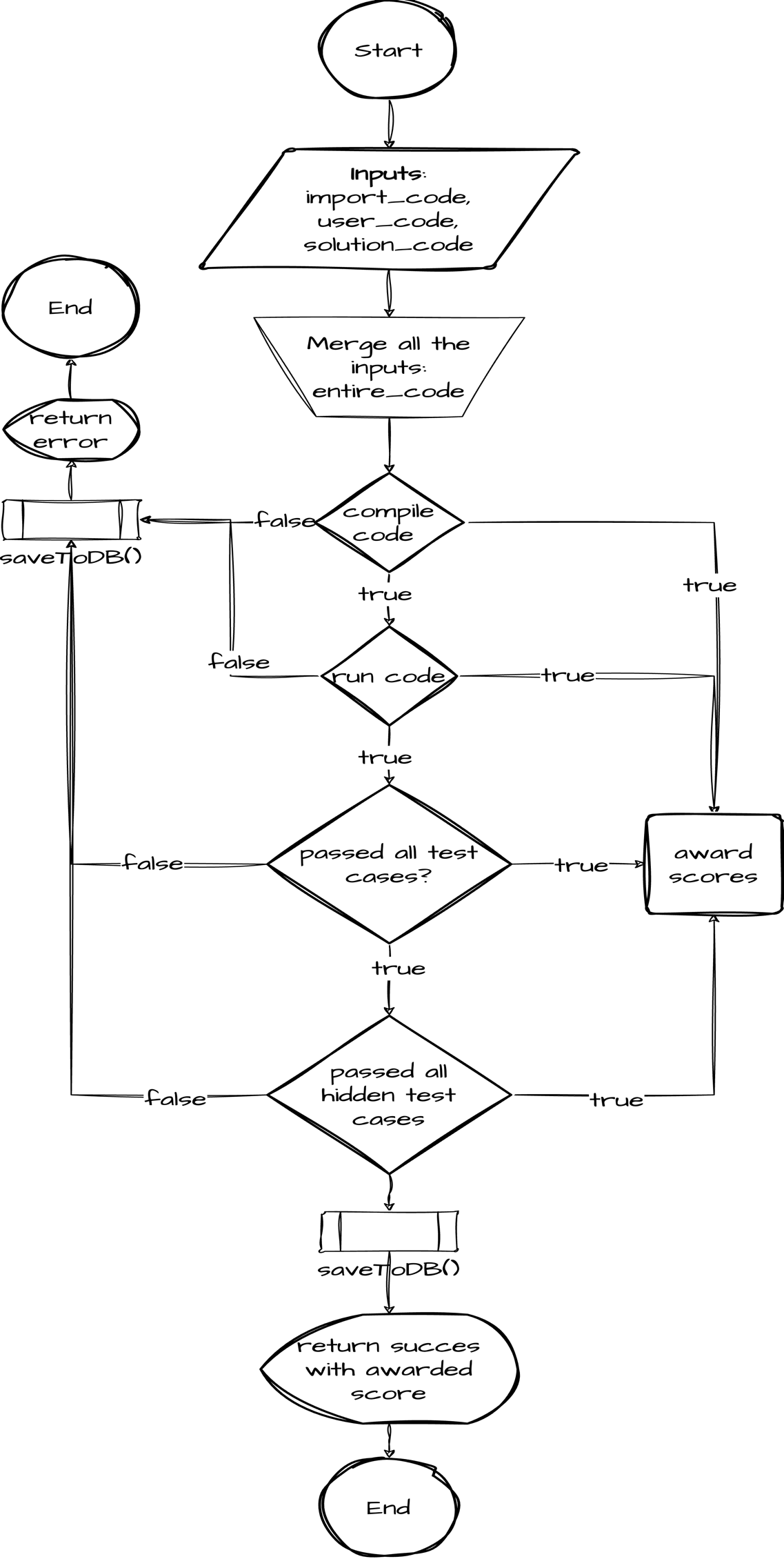
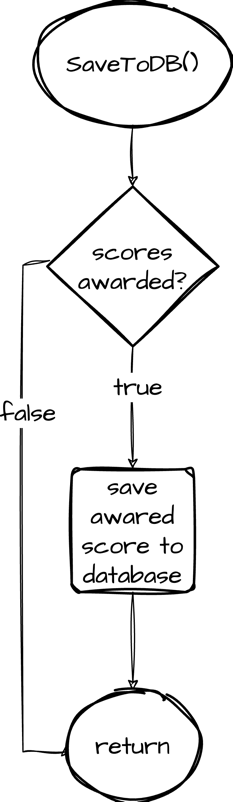
Code Runner module accepts the code from the user and runs it in the backend, providing appropriate error and success messages. This module is also responsible for grading the programming assignment of a student by granting marks to the student once they have reached certain stage of execution like code compiles, code runs without error, code passes all the tests (including the hidden tests)

Major drawback of the current Code Runnermodule is that it’s blocking. Meaning, when one user is running their code then other users will have to wait for that user to finish before their code starts executing. This problem can be fixed by implementing Django Consumers (WebSocket) and Multi-threading. WebSocket enables two-way communication between the server and the client. This way, even when the code takes a long-time user will receive their code output immediate as its available without worrying about server timeout issue. Multi-threading will be used for running long operations like executing user code. This way multiple users can run their code simultaneously without having to wait for the others. One issue with this approach is that it will have a significant impact on the performance of the server specially when multiple users are running their code.

Another approach could be by running the code in another thread and save the output to a file. This way users won’t have to wait for others code to complete executing. But since HTTP doesn’t support two-way communication, user will have to send a request to the server multiple times in order to receive the executed code output which could again slow down the server when large number of users are waiting for their code’s output.

Code runner will function by taking the code of the user and running it in the server using subprocess library. It first compiles the user code, if failed it will award scores to the users (if they earned any) and stops the executing while returning the error message. Similarly, this module then runs the code, tests the code against test cases perform the necessary steps based on the outcome. If code passes all the tests user will receive a success message and will be granted some marks as defined by the professor when creating the assignment.

Plagiarism Checker module will be used to detect the plagiarism in the reports and the code of the user

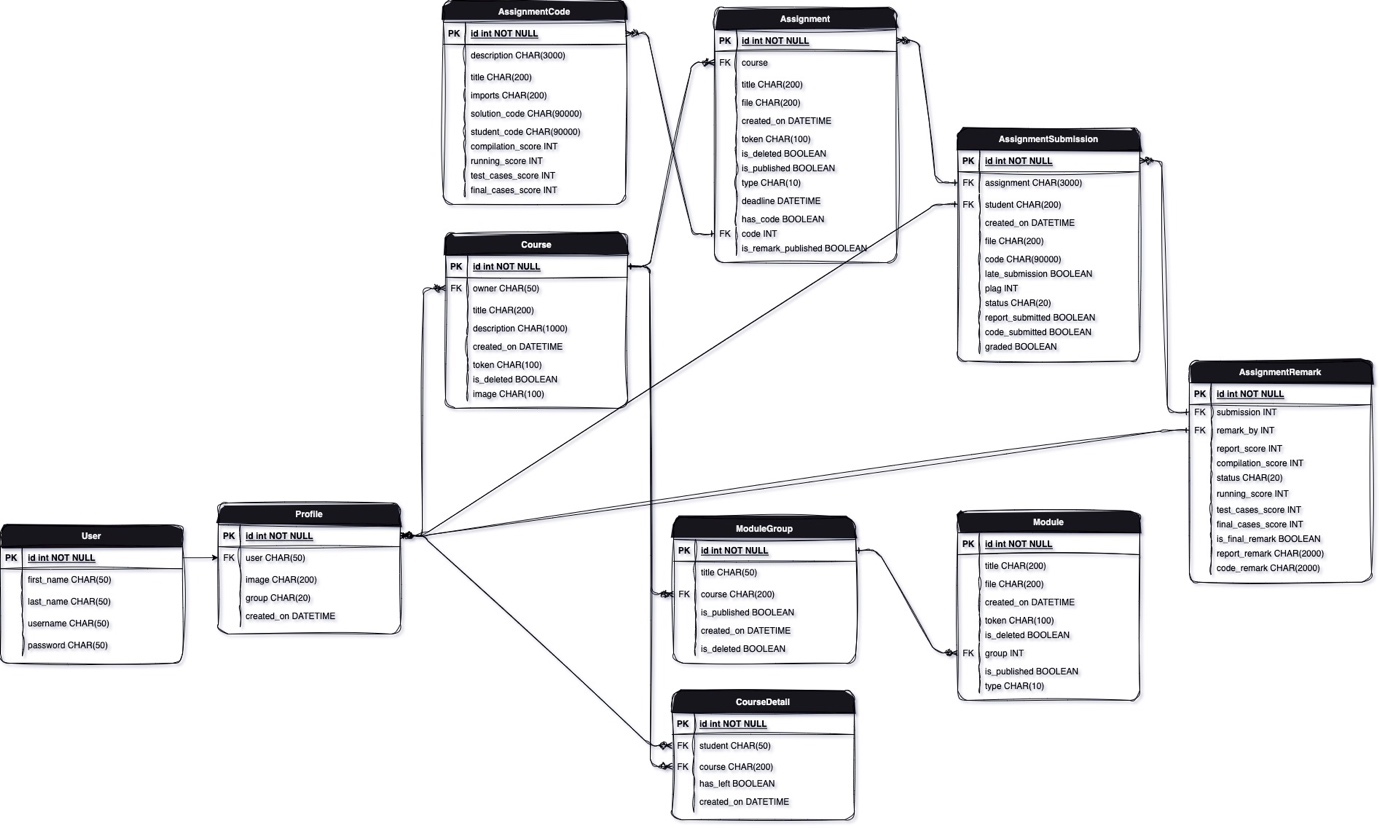


The database consists of 10 tables

|  |  |
| --- | --- |
| **Table Name** | **Description** |
| **User** | Holds information like first & last name, email, username and password |
| **Profile** | Holds users’ personal information like profile picture, user type (teacher/student) a reference to the user table and date & time of account creation. |
| **Course** | Used to store the details of the students who are in any course. |
| **CourseDetail** | Used to store the details of the students who are in any course. |
| **ModuleGroup** | Stores information about different groups for organizing modules like week 1, week 2 and so on.  Each group belongs to a Course |
| **Assignment** | Stores information about an assignment (report portion of the assignment)  It has a reference to AssignmentCode that stores the ‘programming’ portion of the assignment and to Course which indicate the course this assignment belongs to. |
| **AssignmentCode** | Used for storing programming part of an assignment |
| **AssignmentSubmission** | Whenever an assignment is submitted, the details are stored into this table. It has a reference to Assignment table and Profile Table (user who uploaded the assignment) |
| **AssignmentRemark** | Remarks submitted by the professor is stored in this table.  It has a reference to AssignmentSubmission and to Profile table (user who graded the assignment) |

Table 1

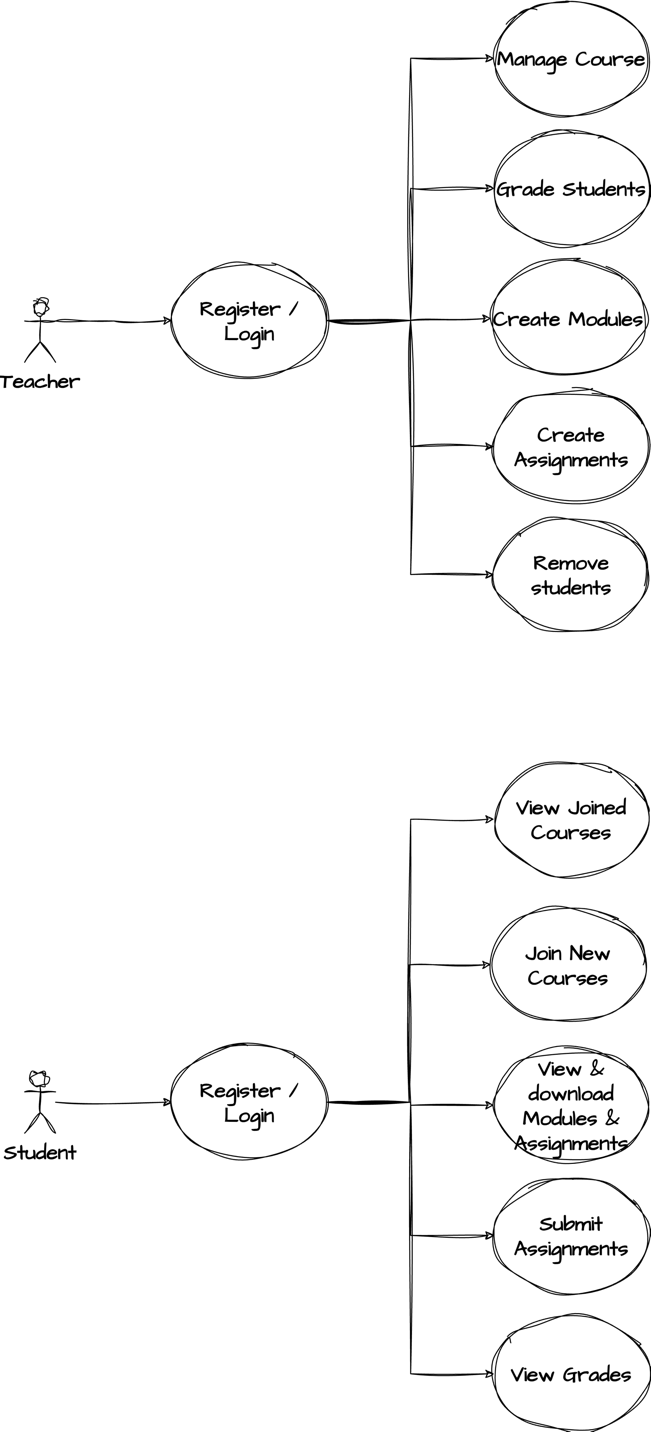
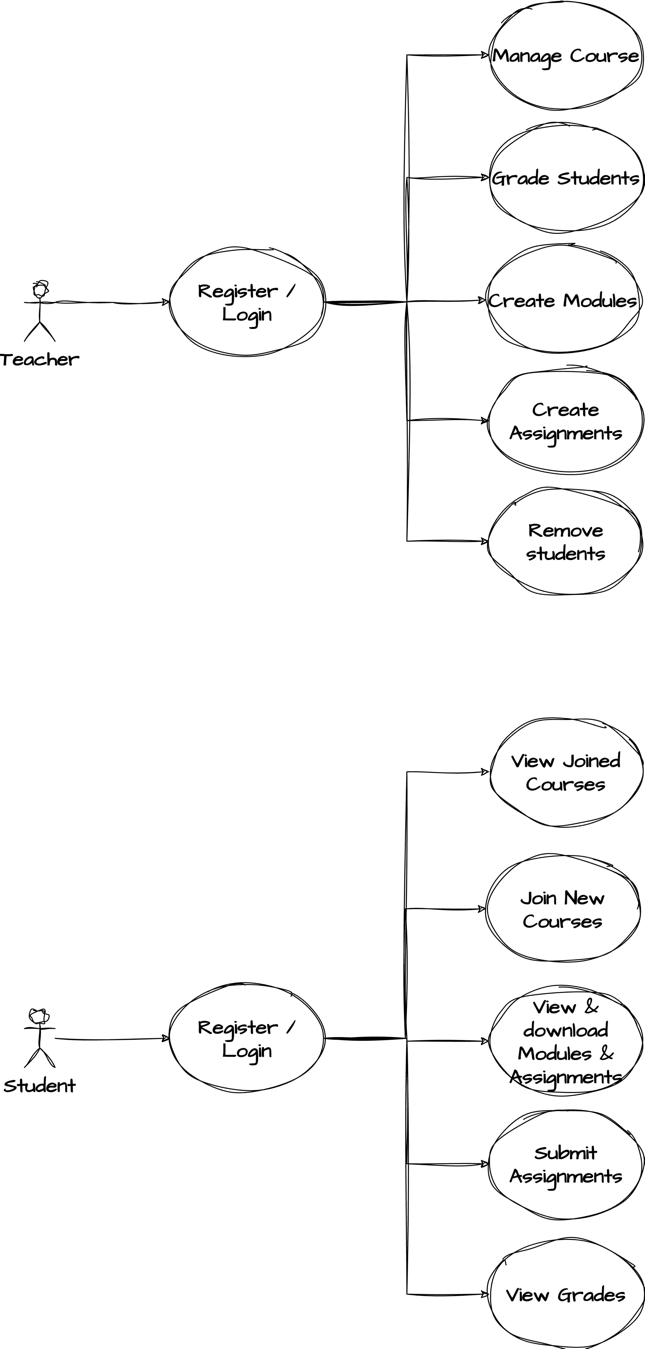
Database ER Diagram



Frontend development started with developing the registration and login page, during the development frontend elements that were being repeated in multiple pages were then turned into a component and were reused throughout the website. One of the major components of the frontend is FormGenerator. This project consists of large number of forms from login and registration forms to creating/modifying/deleting courses, assignments, modules, grading and much more. In order to make generating forms easier this component was designed.

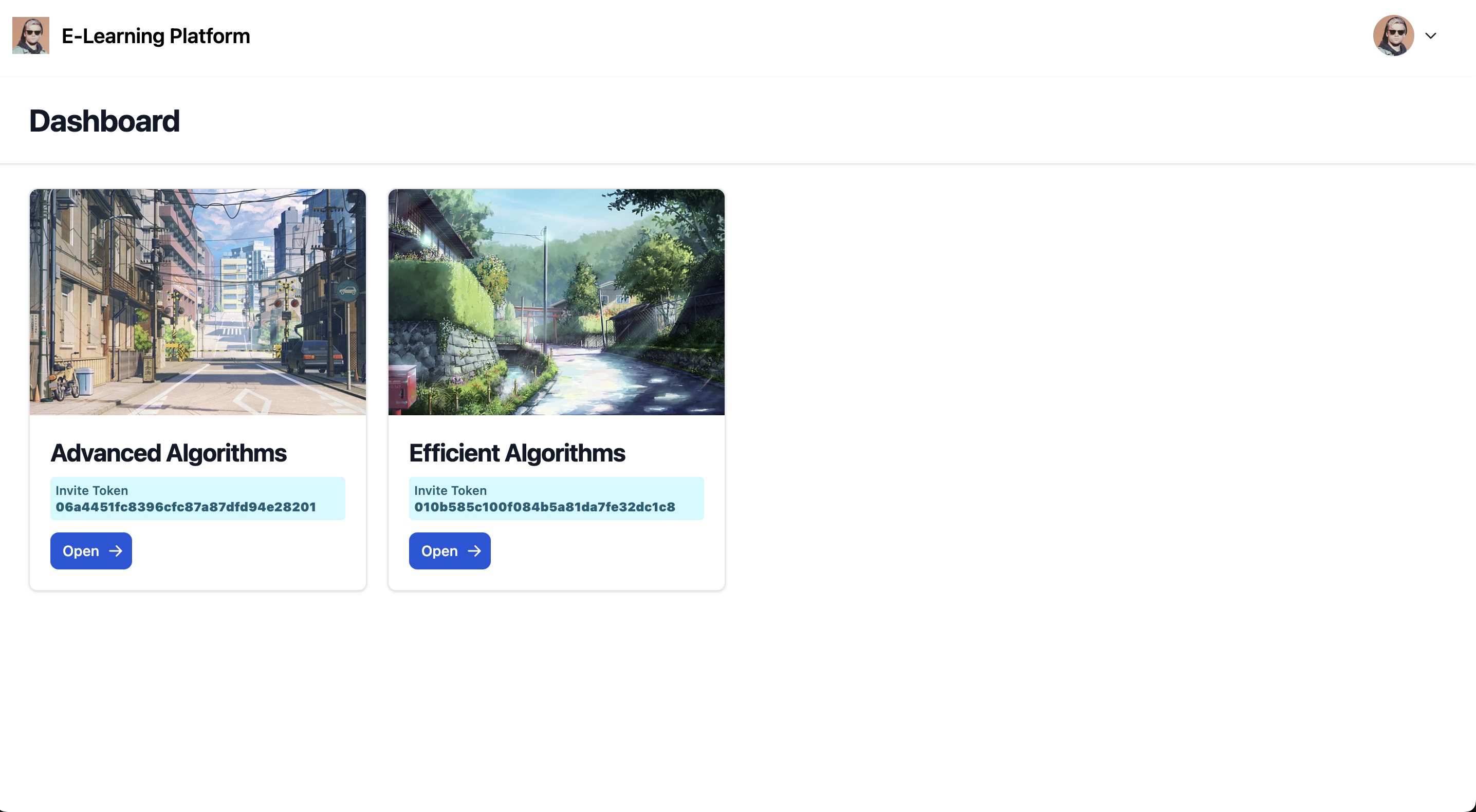
FormGenerator takes in a list of objects. Each object in the list defines an input type like email, text, number, radio box etc. It then loops through the list of objects and displays them on a webpage. This component makes the designing of forms very easy and significantly reduced the number of lines of code to be written.

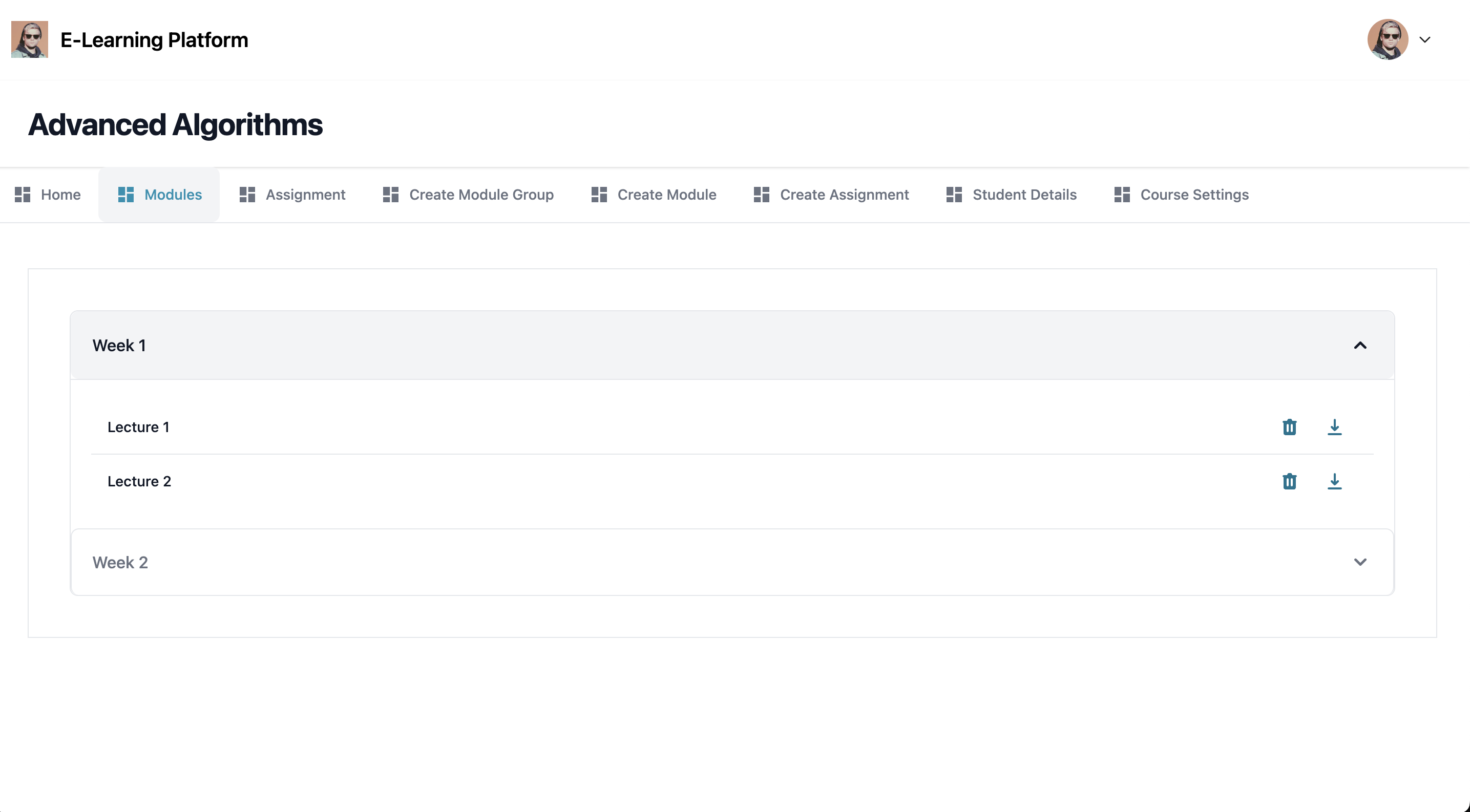
Use case diagram of the software based on different roles

# User Interface Mock-up

For designing the user interface, Canvas website was used as a reference. Couple screenshots of user interface are shown below. Final build will look similar but will have different color theme.





# Data Sources

All the data in this project is generated by the users using the forms in the website. Users’ personal information that will be stored are as follow:

* Email Address: For login / registration / resetting password / communication by the professors.
* First Name & Last Name: For personalized experience and identification.
* Profile Picture: For personalization and can be deleted by the user.
* Courses: Courses that have been joined
* Grades: Users score in every assignment

This software doesn’t store any other personal information like address, phone number, gender, age etc. All the data obtained is entered by the users themselves.

To protect the users and their data, all the confidential data like passwords are being stored after encrypting and hashing them making sure that even if the database gets hacked, hacker cannot access the user data like passwords.

While registering for an account, users would be informed that account once created cannot be deleted. As deleting the account could cause issues with the databases. Since there are a lot of references between the tables and most tables are connected to the Profile table, deleting a user account could affect the integrity of the database and could result in incorrect data being shown to the professors.

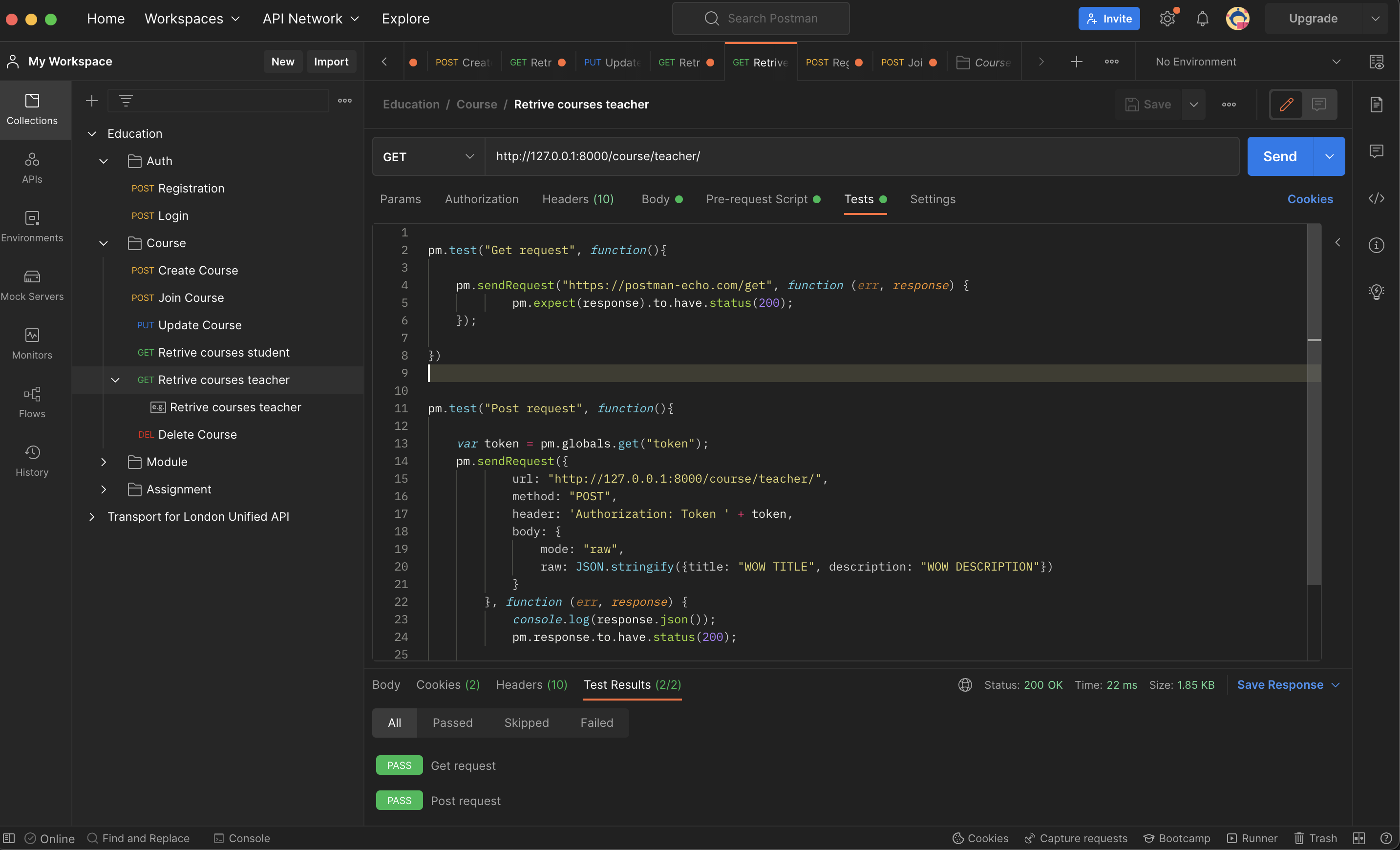
# Testing & Evaluation

Postman will be used to perform Unit Testing and help from friends would be taken to test the software All the modules in this software are pretty small and makes it very easy to implement Unit Testing.

In order to test a module, multiple HTTP requests will be sent to a specific module using Postman with different inputs to ensure it handles all incorrect inputs, duplicate data and invalid requests.

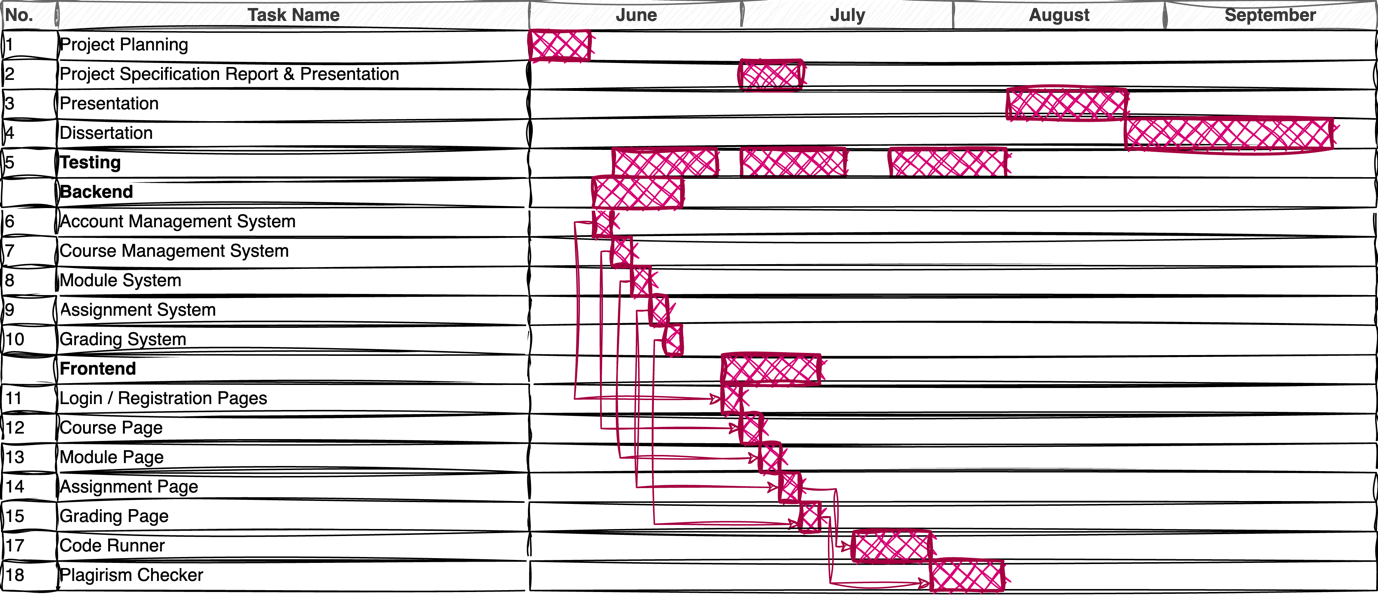
After performing Unit Testing, a friend will be asked to use the software. Without any interference from the developer their software usage would be noted and appropriate actions would be taken if the software breaks / crashes / unexpected behavior is observed during the test.

A module would be considered successful if it passes all the tests.



# Project Plan

Plan for this project is as follow, finish one part of the project (backend / frontend) then tests it and move on to the next item.



# Risks and Contingency Plans

|  |  |  |  |
| --- | --- | --- | --- |
| **Risks** | **Contingencies** | **Likelihood** | **Impact** |
| Long waiting time when running code | In order to solve this issue, WebSocket and Multi-Threading can be used but due to increased complexity this wasn’t implemented and was kept for future improvement. | Medium (depends on the number of people using the website at a time) | This will impact the performance of the website for all the user. This shouldn’t have any significant impact on the performance if small number of users are using the software or if the program that is being executed is pretty small. |
| Server crash | When hosting the website, precautions should be taken to handle the situation when the server crashes. | Low | Website becomes inaccessible to the users unless restarted by the admin. |
| Database offline | Creating scripts to ensure if server ever goes offline, it gets restarted | Low | Website will be accessible but it will won’t display any information from the backend. |

Table 2

# References

Teach Computing. (n.d.). *Pedagogy*. [online] Available at: <https://teachcomputing.org/pedagogy>.

Draw.io (2023). *Flowchart Maker & Online Diagram Software*. [online] app.diagrams.net. Available at: <https://app.diagrams.net/>.