E-Learning Platform

Name: Ayush Katoch
Student ID: 201647931

Supervisor: Mr. Henry Forbes

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 (Cheang, B., Kurnia, 2003), 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.

Due to this time-consuming process Universities tend to give few programming assignments even though it's the best way to teach programming to the students. All this has been discussed in a Tufts University paper dated 2001 which talked about the issues with manual grading in more depth (Tufts, 2001 [3]).

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.

Aims & Objectives

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

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

Objectives that were completed are as follows

1. Account Management System
    a. Allows creation and management of student and teacher accounts.
2. Course Management System
    a. Allows creation and management of courses by the teachers.

    b. Allows students to join a course
3. Module System
    a. Allows uploading of files like lecture notes, ppts and video lectures by the professors
    b. Allows students to download these files.
4. Assignment System
    a. Allows professors to upload an assignment which can be of report type, programming type, hybrid or a Group Assignment.
    b. Students can upload their assignments
5. Grading System
    a. Allows professors to grade student assignments using a predefined rubric cube for marking.
    b. Students will be able to see their grades
6. Plagiarism Checker
    a. Checks for plagiarism in the submitted reports and the code
7. Code Executor
    a. Runs user code in the backend, maintains the virtual environment and the installed libraries.

Objectives that were implemented but were not mentioned in the previous reports

1. Course Management System
    a. Professors can also provide a specific domain name that will allow only those students to join the course who have the domain mentioned in their email address.
    b. Allows teachers to clone their courses which also copies all the lectures and assignments.
2. Assignment System
    a. Professors can now also create group assignments in which multiple students can work together.
3. Live Code Share
    a. Allows users to share code with each other when working in a group assignment which will make collaboration very convenient.

Outputs

An E-Learning website was developed similar to Canvas. This website allows professors to share notes, video lectures, class ppts and assignments (code, report and group) to the students. Professors can also assign groups for the assignments. It also allows professors to grade the student report and ease their job by providing automatic grading of student code based on parameters defined by the professors while creating the assignments.

As for the students, this software allows them to upload their assignments, write and test their code, view their grades and download the resources like notes, lecture videos uploaded by the professors. Some interesting implementation aspects of this project include.

Code Runner

This module accepts the code from the user and executes it in the backend and based on the output, reports the important information like the line where an error occurred, test case that was failed and so on.

On receiving the code, this module creates a temporary file that holds the user code and then executes the file in the backend, providing appropriate error and success messages. After executing, the temporary file is deleted and the code is saved in the database. This module is also responsible for grading the programming assignment of a student by granting marks to the student once they have reached a certain stage of execution like code compiles, code runs without error, code passes all the tests (including the hidden tests).

Live Code Share

This module is responsible for live code share between the team members when doing coding assignments.

Live codeshare uses HTTP Long Polling to accomplish this task. Whenever a user makes changes to the code those changes are sent to the backend, where backend saves the changes in the database. In order to retrieve the changes, the frontend sends a request to the backend every 5 seconds. If there are any changes detected, those changes are merged with the current code. This way both the users can continue to code together without having to worry about losing their progress.

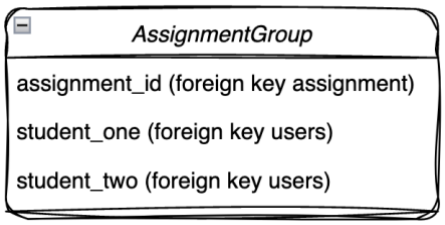
Plagiarism Checker

This module reads the submitted reports for plagiarism among all the other submissions. Based on the amount of plagiarism detected this information is converted into a percentage and shown to the professors when grading the assignments.

String Comparison compares the strings between the two documents and returns a score between 0 to 1 where 1 means 100% identical and 0 means no plagiarism detected. In order to compare the strings, the submitted reports by the students are first converted to vectors using the TF-IDF algorithm. These vectors are then compared using the cosine similarity formula. Advantage of this method is that it's completely free and also doesn't rely on a dataset to function. Disadvantage of this method is that it only looks for plagiarism between the submitted reports and doesn't check the online websites for the plagiarism.

Evaluation

All the objectives that were mentioned in the previous report have been achieved. Instead, additional functionalities like Live Code Share and Group Assignments were also implemented. Due to addition of some functionalities, changes had to be made in the Database. Changes in the Database are limited to adding some additional fields in the existing tables and also adding an extra table called 'AssignmentGroup' to implement the new Group Assignment feature.



With that being said, this project also has some limitations

- **Live Code Share:** Currently this module can cause various issues when dealing with a large number of active users writing code all at once. Live Code shares utilizes Long Polling technique

to achieve its goal. It means that it will send a HTTP request to the server every 5 seconds even if no changes have been made / received. This also means that a Handshake will occur between the server and the client every 5 seconds. Handshake is used to establish a connection to the server and is much slower. Server shouldn't have any issues handling hundreds of active users at once but when dealing with thousands of active users, this can cause performance issues and might even crash the server. One possible way to fix this issue is to use WebSocket. Unlike Long Polling technique that utilizes one way communication protocol HTTP. WebSocket are persistent connections enabling two-way communication between client and server. Making it the perfect option for real time data sharing. Reason for not implementing WebSocket is to reduce the project complexity.

- **Code Runner:** Only limitation with the code runner module is that it only supports the Python programming language. Further programming languages support can be added in the future but to reduce the complexity of the project, only one programming language was implemented.

References

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

Cheang, B., Kurnia, A., Lim, A. and Oon, W.-C. (2003). On automated grading of programming assignments in an academic institution. *Computers & Education*, 41(2), pp.121–131. doi: https://doi.org/10.1016/s0360-1315(03)00030-7