Project Title: Student Code Online Review and Evaluation
Names and email addresses of team members:

Charlie Collins, ccollins2021@my.fit.edu
Thomas Gingerelli, tgingerelli2021@my.fit.edu
Logan Klaproth, lklaproth2021@my.fit.edu
Michael Komar, mkomar2021@my.fit.edu
Faculty advisor from CSE: Raghuveer Mohan, rmohan@fit.edu
Client name and affiliation: Raghuveer Mohan, CSE Professor

Date(s) of Meeting(s) with the Client for developing this Plan:
August 27th, 2024, September 3rd, 2024

Goals:

- Develop a more robust submission platform
  - Automatically grades student submissions
  - Provides project-specific feedback based on test cases
  - Ability to accept and run multiple file/program types
- Streamline the submission and grade input process
  - Student authentication
  - Canvas integration
  - Uses plagiarism detection
- Develop skills in full-stack development
  - Interactive GUI on student and professor side
  - Back end for automatic grading
  - Storage of submissions and assignment files

Motivations:

From the student's perspective, the current authentication process is a pain point as we must register and save an access token, increasing the steps required to submit an assignment. From the student perspective, the current submit server does not allow for immediate feedback on our assignment performed in the autotests, which provides students with less knowledge about how they are performing.

From the professor's perspective, it can be a pain point to have to manually enter grades into canvas, once the submissions have been graded, so canvas integration eases this by automatically submitting the grades to canvas. Additionally, assignments submitted directly to canvas can be placed into folders or have their name changed by canvas. This can make running auto testing scripts difficult. However, other sites, like Kattis, that can have auto testing, don't allow for other necessary features like uploading assignments.

Approach:

Auto Testing

Student

- The student will have their assignment submissions tested against the hidden test cases created by the professor

Professor

- The professor can configure the auto-testing environment to run student submissions within
- The professor can supply both sample, and hidden test cases to be run by the auto test
- The professor will be able to view how many test cases a student's submission has passed
- The professor can set the final grades which will be uploaded to Canvas

Immediate Feedback

Student

- The student can view the exact number of test cases that were passed and how many were failed
- The student can view professor feedback associated with a particular test case that they failed
- The student can view previous submission attempts auto test scores to determine if their work has improved

Professor

- The professor can create helpful feedback to attach to a test case in order to help students understand where they went wrong
- The professor can select how much feedback their students can receive from an assignment submission

Grading Portal Integration

Professor

- The professor can modify scores to match their grading criteria
- The professor can have the assignment grades automatically uploaded to canvas after the due date has passed
- The professor can export the assignment grades as a .CSV file

MOSS integration

Professor

- The professor can view a student submission's MOSS plagiarism detection score
- The professor can set limits to the acceptable plagiarism score so that they are notified when a student's submission exceeds the limit

User Authentication

- The user can access the platform through a TRACKS Login, using the same credentials as other florida tech applications
- The user will not have to register and maintain a control code for assignment submission

Shell Client

- The user can interact with the application through a command line interface in which they can do the following:

  Student

  - The student can upload C, C++, Java, and Python files to a class assignment
  - The student can view the grade of a submission after the due date passes
  - The student can view the breakdown of hidden test cases that they passed or failed for each submission
  - The students can view written professor feedback associated with each hidden test case that they failed

  Professor

  - The professor can create new assignments and test cases, as well as associated feedback, for students
  - The professor can receive and view student submissions as well as their scores from the auto test
  - The professor can view the MOSS plagiarism score for each student submission

Web-App

- The user can interact with the application through a graphical interface on the web in which they can do the following:

  Student

  - The student can navigate through their classes and assignment descriptions
  - The student can upload C, C++, Java, and Python files to a class assignment
  - The student can view the breakdown of hidden test cases that they passed or failed for each submission as well as associated professor feedback

Professor

- The professor can create new assignments and test cases, as well as associated feedback, for students.
- The professor can receive and view student submissions as well as their scores from the auto test.
- The professor can view the MOSS plagiarism score for each student submission

Novel features/functionalities:

User Authentication

While a submit server has existed in the past, it relied on a control code to authenticate students. This was inconvenient as students had to not only register for the code, but they also had to make sure they stored it somewhere to be able to submit their code. By authenticating with TRACKS, we are providing a new feature that improves upon the previous submit server.

Assignment Creation

While other platforms, such as Kattis, allow for automated grading with instant user feedback, they don't provide the user (Professor) to upload new assignments for their students to work on.

Test Case feedback

From our survey of other relevant applications, we did not find any that allow the professor to attach feedback to specific test cases. We feel this feedback has the potential to increase a student's ability to learn from their errors.

Algorithms and Tools: Potentially Useful Algorithms and Software Tools

- MERN: Full Stack
    - MongoDB
    - Express
    - React
    - Node JS
- Django: Front End
- Canvas API
- MOSS API
- TRACKS API
- AWS Cloud Services
- Rust
- Python

<u>Technical Challenges:</u>

1. Long term secure storage of submissions
2. Canvas class integration
3. Student user authentication using TRACKS

Milestone 1 (Sep 30): itemized tasks:

Discuss what we are using for configuration of autograding

- Compare and select technical tools for the server implementation, web development stack, file transfer, and user authentication
- Provide small demos to evaluate tools for server implementation, web development stack, file transfer, and user authentication
- Resolve technical challenges: Secure long term submission storage, canvas class integration, and user authentication
- Compare and select collaboration tools for software development, documentation, communication, and task calendar
- Create Requirement Document
- Create Design Document
- Create Test Plan

Milestone 2 (Oct 28): itemized tasks:

- Implement, test, and demo the shell application
- Implement, test, and demo creating assignments in the shell application
- Implement, test, and demo submitting assignments to the shell application

Milestone 3 (Nov 25): itemized tasks:

- Implement, test, and demo auto grading
- Implement, test, and demo immediate feedback
- Implement, test, and demo server client interaction

Task Matrix for Milestone 1 (teams with more than one person)

Task	Tommy	Logan	Michael	Charlie
Compare and select Technical Tools	User Authentication	Web Development Stack	Server Implementation	File Transfer
Tool Demos	25%	25%	25%	25%
Resolve Technical Challenges	20%	20%	35%	25%
Compare and select Collaboration Tools	25%	25%	25%	25%
Requirement Document	35%	20%	15%	30%
Design Document	35%	25%	35%	15%
Test Plan	5%	35%	25%	35%

Approval from Faculty Advisor

"I have discussed with the team and approved this project plan. I will evaluate the progress and assign a grade for each of the three milestones."

Signature: ________________________________ Date: ________