Milestone Three Progress Evaluation

Project Title: Student Code Online Review and Evaluation 2.0
Names and email addresses of team members:
Dorothy Ammons dammons2022@my.fit.edu
Patrick Kelly pkelly2022@my.fit.edu
Shamik Bera sbera2022@my.fit.edu
Rak Alsharif ralsharif2021@my.fit.edu
Faculty advisor from CSE: Raghuveer Mohan, rmohan@fit.edu
Client name and affiliation: Raghuveer Mohan, CSE Professor

Task	Dorothy	Patrick	Shamik	Rak	To Do
1. Finalize backend and databases	100%	0%	0%	0%	
2. Set up hosting with Google Cloud Run	0%	0%	0%	0%	Set up hosting
3. Add the LLM for AI detection to the web application	0%	0%	0%	50%	Add functionality to front end
4. Add the MOSS functionality to the web application	0%	70%	0%	0%	Connect MOSS to the students' assignment submissions
5. Add the rubric page and functionality	50%	0%	0%	0%	Fix database storage, connect to autotest files
6. Add the import functionality for rosters	0%	0%	60%	0%	Connect roster to the created course to see the added students
7. Add the export functionality for grades	0%	0%	0%	0%	Coincides with rubric functionality, add export CSV option for student grades after

Tasks

1. Finalize backend and databases

   This task involved fixing the way some of the endpoints sent and retrieved data. Additionally, large files needed to be stored in a storage bucket so that they may not clog the database.

2. Set up hosting with Google Cloud Run

   This task was chosen with hopes we may launch a test across classrooms for the next milestone. We wanted to set up the server hosting so that students and professors could access and use our web application.

3. Add the LLM for AI detection to the web application

   This task involved connecting the AI detection functionality to the front and back end. This means creating an option to test submissions for AI as well as visualizing that data.

4. Add the MOSS functionality to the web application

   This task involved connecting the MOSS similarity score detection functionality to the front and back end.This means creating an option to test submissions for collusion as well as visualizing that data.

5. Add the rubric page and functionality

   This task involved adding the rubric creation button to the assignment creation page. Additionally, a rubric page should be created with options for total points, attempt, late penalties and more. Each test case should have a slot for how many points that test case is worth. Lastly, the autotest files should interact with the rubric to send back the new scores for each submission.

6. Add the import functionality for rosters

   This task involved adding a button on the course page to import a CSV file (exported from canvas roster) in order to add students to the course. The database needs to save the student name and email data in order for those student accounts to be connected to the SCORE(2.0) class.

7. Add the export functionality for grades

   This task involved adding a button to export all student grades for an assignment into a CSV file. The file needs to be in a format so that it may be uploaded to a canvas assignment easily.

Contributions

Dorothy Ammons: Dorothy finalized all endpoints and database structures. She moved test cases and submissions from Google Cloud Storage to Firestore Storage and replaced the bucket in all the endpoints. She updated the structure for the storage buckets and ensured stored files would remain under the free usage limit and are retrieved properly. She created the rubric functionality and updated the auto test wrapper class to work with the previous teams auto test and auto feedback files. Additionally, she added the rubric button and page. She added an endpoint to store all of the rubric data into the database. Lastly, she added a points column for every test case and adjusted the endpoint to include that in the database.

Shamik Bera: Shamik created the presentation slides and wrote the Milestone Three Evaluation document. He fixed a minor bug in the UI layout of the sign-in page. He implemented the import roster functionality and made a file input to ensure that the roster is in a CSV format. He achieved this by updating the EditCourse code for both the JSX file and the CSS file. He also ensured that the CSV roster file only accepts those that contain students' full names and students' emails. He updated the Firestore course document by adding another endpoint to parse CSV and store each student's information for each roster.

Patrick Kelly: Patrick focused on implementing the new MOSS similarity detection system and integrating it throughout the SCORE platform. Built a Python-based comparison tool that generates a similarity matrix for submitted code files and connects it to the backend through a new Flask route (/api/moss/demo). He resolved routing issues, added a sample submission directory, and verified all responses using cURL to ensure the API was returning correct and consistent results. On the frontend, Patrick created a dedicated React page to display the similarity matrix, added sidebar navigation, and connected the page to the backend using a fetch() call that dynamically renders the results in a clean table format. He also rebuilt the Vite frontend, fixed static file serving problems, and ensured the React build was properly served through Flask. Patrick's work fully integrates the demo MOSS system into SCORE and establishes the foundation for future expansion into real student submissions.

Rakan Alsharif: Rakan implemented the entire AI code detection subsystem for Milestone Three. He integrated multiple pretrained LLM models into the backend, including the RoBERTa-based classifier and a GPT-based detector. He removed outdated or non-functional models, created a clean and stable ensemble pipeline, and ensured that each model consistently and reliably returned probability outputs. Rakan also refactored the detection API endpoint, added logging and error-handling, and validated the results against multiple code samples. Finally, he prepared the AI detection functionality so that it can be connected to the frontend in the next milestone, including a standardized JSON output structure for visualization.

Next Milestone

Task	Dorothy	Patrick	Shamik	Rak
1. Fix rubric/assignment database clash	100%	0%	0%	0%
2. Connect rubric to autotest files	50%	0%	50%	0%
3. Add the export functionality for grades	100%	0%	0%	0%
4. Connect roster to the created courses	0%	10%	90%	0%
5. Integrate AI Detection (LLM) into the instructor dashboard	0%	0%	0%	100%
6. Set up hosting with Google Cloud Run	100%	0%	0%	0%
7. Connect MOSS to real student submission	0%	100%	0%	0%

Date(s) of meeting(s) with Faculty Advisor/Client during the current milestone:

10/29/2025, 11/05/2025, 11/12/2025, 11/19/2025

Faculty Advisor feedback on each task for the current Milestone

- Advised: Instead of connecting our hand made LLM, research pre trained models for more accurate results
- Recognized: We had some issues with connecting to the autotest files from the previous group, advised working over the winter break
- Advised: Meet with Dr. White to learn more about clustering algorithms and ways to visualize our data

Faculty Advisor Signature: _____ Date: _____

Evaluation by Faculty Advisor

Faculty Advisor: detach and return this page to Dr. Chan (HC 209) or email the scores to pkc@cs.fit.edu

Score (0-10) for each member: circle a score (or circle two adjacent scores for .25 or write down a real number between 0 and 10)

Dorothy Ammons	0	1	2	3	4	5	5.5	6	6.5	7	7.5	8	8.5	9	9.5	10
Patrick Kelly	0	1	2	3	4	5	5.5	6	6.5	7	7.5	8	8.5	9	9.5	10
Shamik Bera	0	1	2	3	4	5	5.5	6	6.5	7	7.5	8	8.5	9	9.5	10
Rak Alsharif	0	1	2	3	4	5	5.5	6	6.5	7	7.5	8	8.5	9	9.5	10

Faculty Advisor Signature: _____ Date: _____