Milestone Two 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. Replace frontend/backend with Flask and Firestore	80%	0%	0%	0%	Finish endpoints for processing test cases
2. Replace rust server with Python	0%	10%	70%	0%	Create the CLI Client file
3. Add AI detection page to website without functionality	0%	0%	0%	5%	Add the buttons and page(s) for the AI detection
4. Create and test LLM for AI detections	0%	0%	0%	90%	Continue refining accuracy and model evaluation
5. Create visuals for AI detections	0%	0%	0%	30%	Create visuals from the LLM results
6. Built and tested MOSS integration with Matrix	0%	100%	0%	0%	Integrate into grading pipeline for Milestone 3 and connect to frontend for visualization

Tasks

Task one:

For task one, we aimed to transfer the system to something sustainable for server hosting. This involved switching the old Node.js and MongoDB backend with Flask and Firebase. These steps are necessary in order to host the server reliably on a system such as Google Cloud Run. Each endpoint in the backend was replaced with Flask API operations and data was migrated away from MongoDB and to Firestore database API / Google cloud buckets.

Task two:

For task two, we needed to set up the CLI Client file to interact with the Google Cloud Run server from the command line terminal. This involves creating the commands to be run in the terminal and connecting the commands to backend endpoints.

Task three:

For task three, we wanted to create some basic buttons or pages for our AI detection functionality. We believe adding the space for the functionality will make connecting the LLM easier later.

Task four:

For task four, we chose to create and test the LLM for AI detection. This LLM needs to take in code as an input and output a statistic in how likely the code was to be generated by AI. Testing is crucial and the predictions need to be as accurate as possible. The LLM will be used by professors through the SCORE (2.0) application to notify them of AI usage in assignments.

Task five:

For task five, we needed to decide how the data from the AI detections would be shown to professors. This includes integrating customizable thresholds for displaying the data as well as a graphical interface for the web application.

Contributions

Dorothy Ammons: Dorothy created the SCORE (2.0) email, Google Cloud Run account and server, and the Firebase account. She connected the accounts to the Flask API. She replaced the entirety of the backend files with Flask endpoints. She updated the frontend end files to interact with the backend.

Shamik Bera: Shamik created the presentation slides and the Milestone Two Evaluation document. He tested the progress of the Flask API connected with the Google Cloud Run account, server, and Firebase account. He checked whether the frontend files successfully

interacted with the backend. He also replaced the rust server with Python that includes Flask and connected the commands to the backend.

Patrick Kelly: Patrick developed and implemented the MOSS similarity detection prototype for the SCORE 2.0 backend. He built a Python-based system to analyze student submissions and generate a similarity matrix showing how closely each file matched others. He created a new Flask API route (`/api/moss/demo`) that returns the results in JSON format, allowing future frontend integration. Patrick successfully tested the endpoint locally with sample submissions, confirmed working similarity percentages, and prepared the foundation for connecting real MOSS reports in the next milestone.

Rak Alsharif: Rak developed and tested the initial LLM-based AI detection module for S.C.O.R.E. (2.0). He implemented a Python baseline model that analyzes student code and predicts the probability that a submission was generated by AI. The detector extracts key text-based and structural features such as docstrings, comment ratios, average line length, and AI-related keywords. Additionally, he contributed ideas for how this AI detection functionality could later be integrated into the Flask backend and displayed to professors in the web interface.

Next Milestone

Task	Dorothy	Patrick	Shamik	Rak
1. Finalize backend and databases	100%	0%	0%	0%
2. Set up hosting with Google Cloud Run	100%	0%	0%	0%
3. Add the LLM for AI detection to the web application	0%	0%	0%	100%
4. Add the MOSS functionality to the web application	0%	100%	0%	0%
5. Add the rubric page and functionality	25%	25%	25%	25%
6. Add the import functionality for rosters	0%	0%	100%	0%

7. Add the export functionality for grades	0%	0%	100%	0%

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

10/27/2025

Faculty Advisor feedback on each task for the current Milestone

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: _____