

Student Code Online Review and Evaluation 2.0

TEAM: SHAMIK BERA, DOROTHY
AMMONS, PATRICK KELLY, RAK ALSHARIF
ADVISOR/CLIENT: RAGHUVeer MOHAN

Table of Contents

- Milestone 4
- Milestone 4 - Completion Matrix
- Automatic Rubric Based Grading Feature
- Roster Import Feature
- AI Detection Integration
- AI Detection Testing
- Milestone 5 – Task Matrix

Milestone 4

- Complete Automatic Grading Rubric
- Complete Google Cloud Run Hosting
- Complete Importing Roster
- Integrate AI detection results into the submission workflow
- Evaluate AI detection accuracy using sample student submissions
- Connect AI detection output to the web interface for professor review
- Create a Cluster Algorithm for COPS with visualization

Milestone 4 – Completion Matrix

Task	Dorothy	Patrick	Shamik	Rak	To Do
1. Rubric Autograder Completion	90%	0%	0%	0%	Add rubric to student assignment dashboard
2. Complete Google Cloud Run Hosting	0%	0%	0%	0%	
3. Import Roster Completion	0%	0%	80%	0%	Display the list of students from CSV file or manual add students
4. AI Detection Integration & Testing	0%	0%	0%	80%	Connect AI results to dashboard. Evaluate detection accuracy. Refine ensemble scoring.
5. Complete COPS Matrix	%	%	%	%	

Automatic Rubric Based Grading

- When a student makes a code submission, the autograder will use the professor-created rubric in order to give students a grade

Criteria	Points
Total	10
Compilation	5
Attempt	3
Under 100 Seconds	1
After 5 Days Late (deduction)	5

Test Cases

Input	Output	Feedback	Points	Verifier	Visibility
<input type="button" value="Upload Input"/> HelloWorldOutput.txt	<input type="button" value="Upload Output"/> HelloWorldOutput.txt	This is feedback	2	<input checked="" type="checkbox"/> Diff <input type="checkbox"/> Custom	<input checked="" type="checkbox"/> Visible <input type="checkbox"/> Hidden

grade test

Assigned:	Due
Highest Score: 11/10	View More
Test Case tc1	
Description	
Visible Test Cases	
Sample Input	Sample Output
Hello World!	Hello World!

Automatic Rubric Based Grading Demo

The screenshot shows a web application interface for "S.C.O.R.E". On the left, there's a red sidebar featuring a school logo of a panther head. The main area displays the title "DemoClass" above a table. The table has two rows. The first row contains two columns: "Due: 2019-02-04" and "New Student Submission". The second row contains a single column labeled "Description" with the text "This is a demo class".

Assignment Due	Assignment
Due: 2019-02-04	New Student Submission
Description	
This is a demo class.	

Roster Import

- Under each course, the button for seeing the current roster is implemented, so whatever students has been added or imported through CSV file will be shown in that panel.

The screenshot shows a user interface for managing classes. On the left, a sidebar has a logo of a panther head and navigation links: 'Classes', 'TestClass1', 'TestClass2', and 'Create Class'. The main area is titled 'S.C.O.R.E' and shows 'TestClass1'. It includes buttons for 'Edit Course', '+ Create Assignment', and 'View Roster' (which is highlighted with a red oval). Below this is a table for an assignment named 'Assignment grade test'. The table has columns for 'Due' and 'View Student Submissions'. A red 'X' is in the top right corner of the table. The 'Description' row contains the text 'This is a description'.

Assignment grade test		X
Due:	View Student Submissions	
Description		
This is a description		

Roster Import (P2)

- Inside of the panel for roster view, the professor would see the list of students' name and/or email that have recently been added.
- Currently, it's showing that no students have been added to the roster since displaying the student's information from the data is remaining.

The screenshot shows a user interface for a course management system. At the top, there is a logo of a panther and the acronym "S.C.O.R.E". Below the logo, the course name "TestClass1" is displayed. There are several buttons: "Edit Course", "+ Create Assignment", and "View Roster". A modal window titled "Current Roster" is open, showing the message "No students found." at the bottom. The main content area below the roster shows a table with one row, labeled "Assignment grade test". The table has columns for "Due:" and "View Student Submissions". A description box contains the text "This is a description".

AI Detection Integration

- Integrated AI detection into backend submission workflow
- Created API endpoint to analyze student code
- Returns AI probability score in JSON format
- Supports multiple detection models (ensemble system)

AI Detection Testing & Evaluation

- Tested using human-written and AI-generated code samples
- Verified probability consistency
- Handled API failures safely
- Preparing dashboard visualization for professors

Milestone 5 – Task matrix

Task	Dorothy	Patrick	Shamik	Rak
1. Complete Google Cloud Run Hosting	100%	0%	0%	0%
2. Work with our advisor to demo a release into classrooms	25%	25%	25%	25%
3. Test and correct security bugs	50%	0%	50%	%
4. Complete C.O.P.S	0%	100%	0%	0%
5. Complete AI detection	0%	0%	0%	100%
6. Add export grades functionality/finish import	0%	0%	100%	0%



Questions?