

Student Code Online Review and Evaluation 2.0

*TEAM: SHAMIK BERA, DOROTHY
AMMONS, PATRICK KELLY, RAK ALSHARIF
ADVISOR/CLIENT: RAGHUVVEER 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

grade test

Assigned: Due

Highest Score: 11/10 [View Most](#)

Test Case tc1



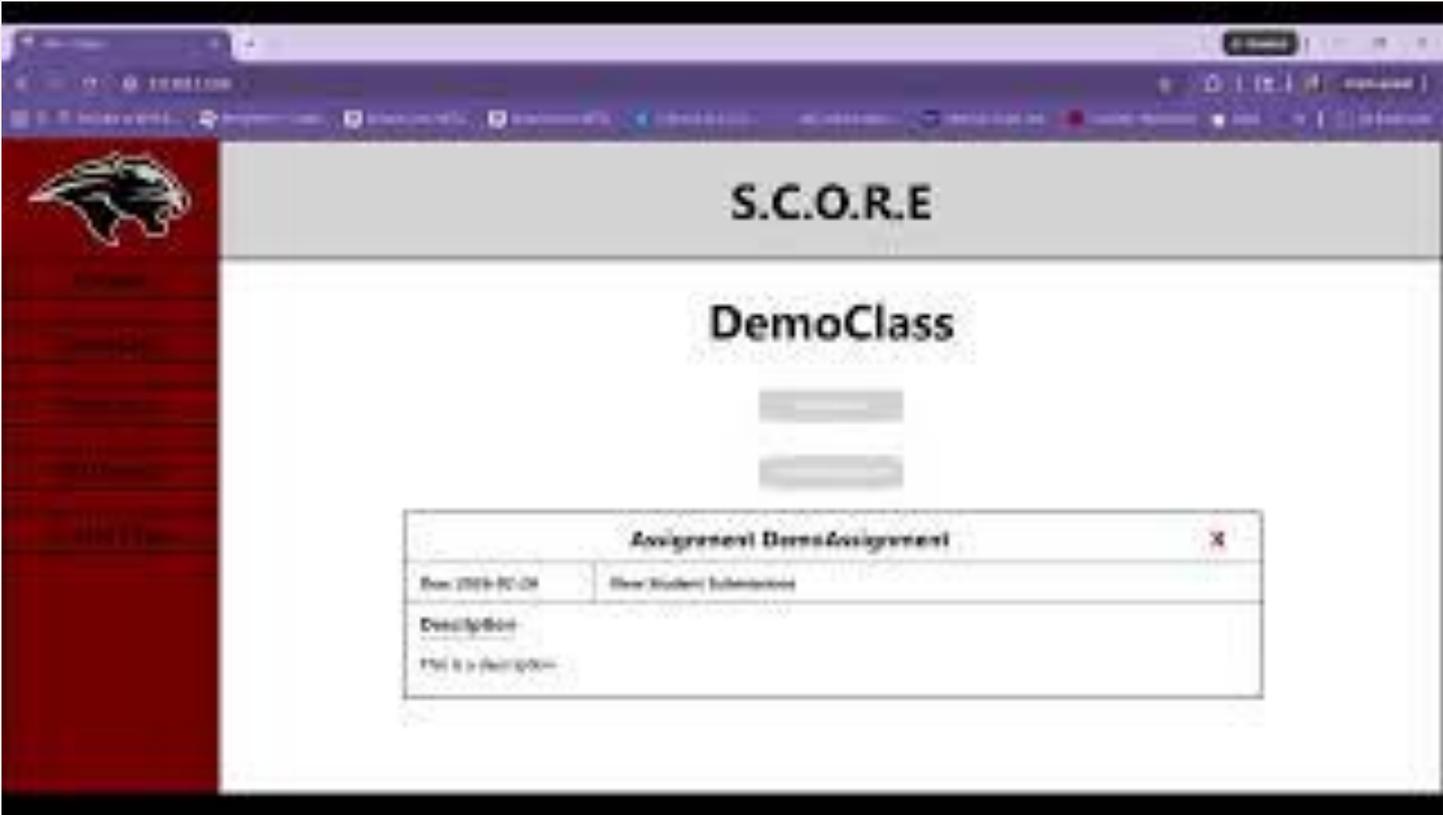
Description

Visible Test Cases

Sample Input	Sample Output
Hello World!	Hello World!

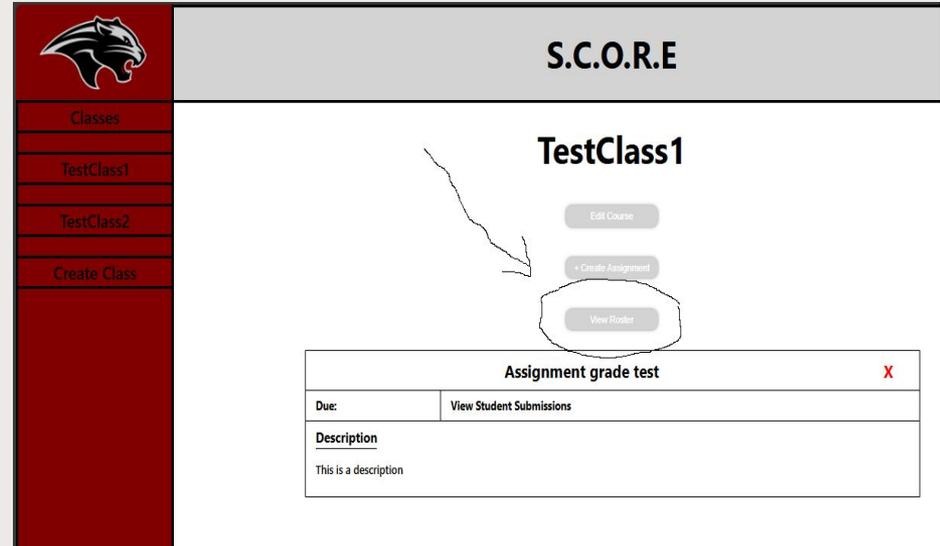
Test Cases					
Input	Output	Feedback	Points	Verifier	Visibility
<input type="button" value="Upload Input"/> <i>HelloWorldOutput.txt</i>	<input type="button" value="Upload Output"/> <i>HelloWorldOutput.txt</i>	<input type="text" value="This is feedback"/>	<input type="text" value="2"/>	<input checked=""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

Automatic Rubric Based Grading Demo



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 displays the S.C.O.R.E. (Student Course Overview and Reporting Environment) interface. On the left is a dark red sidebar with a white horse head logo at the top, followed by menu items: 'Classes', 'TestClass1', 'TestClass2', and 'Create Class'. The main content area has a grey header with 'S.C.O.R.E.' and a white body with 'TestClass1' in bold. Below the course name are three buttons: 'Edit Course', '+ Create Assignment', and 'View Roster'. The 'View Roster' button is circled in black, with a white arrow pointing to it from the left. Below the buttons is a table for an assignment:

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 displays the S.C.O.R.E. (Student Course Overview and Reporting Environment) interface. On the left is a dark red sidebar with a white horse head logo at the top. Below the logo are menu items: 'Classes', 'TestClass1', 'TestClass2', and 'Create Class'. The main content area has a grey header with 'S.C.O.R.E.' and a white sub-header with 'TestClass1'. Below the sub-header are three buttons: 'Edit Course', '+ Create Assignment', and 'View Roster'. A 'Current Roster' button is circled in red, with a 'Close' button next to it. Below the 'Current Roster' button, it says 'No students found.' Below this is a table for 'Assignment grade test' with a red 'X' in the top right corner. The table has a 'Due:' field and a 'View Student Submissions' button. Below the table is a 'Description' section with 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?