Final Presentation of Plagiarism Checker

CS5500 Team 106

System Functionality

Register
Login
Logout

Why it is needed for customers?

Because customers want to see plagiarism checking history, and historical report is identified by its ID and the user who creates it.

Aka, keep tracking.

Register/Login/Logout

CS106-9: Log in functionality - front end

CS106-10: Log in functionality - back end

CS106-21: React unable to call end point on AWS

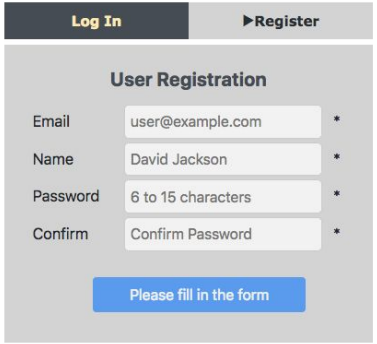
CS106-34: Front end refreshing automatically

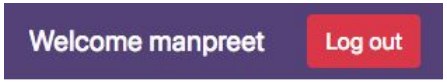
CS106-36: Fix the confirm password

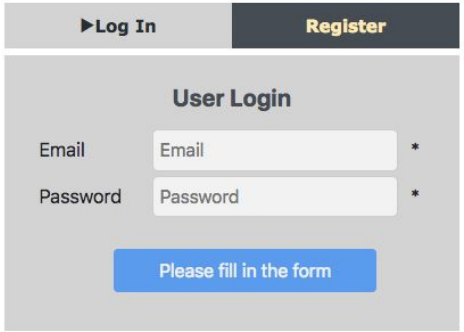
CS106-42: Securing login sessions

CS106-59: Login test case impacted

CS106-70: Login uid bug

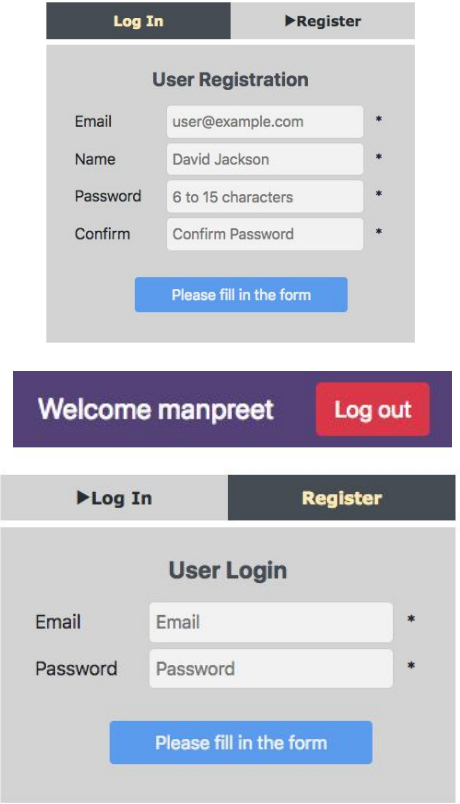






Features

- Email validation: not null, must be in the format like user@example.com
- Name validation: not null
- Password length validation
- Password confirmation validation
- User cannot register or login if the information is not complete



Start New Run

Plagiarism Checking → Run

Sample testing Git Repos:
https://github.com/bharat94/testRepo1.git
https://github.com/bharat94/testRepo2.git

The main and the most important functionality of the application.

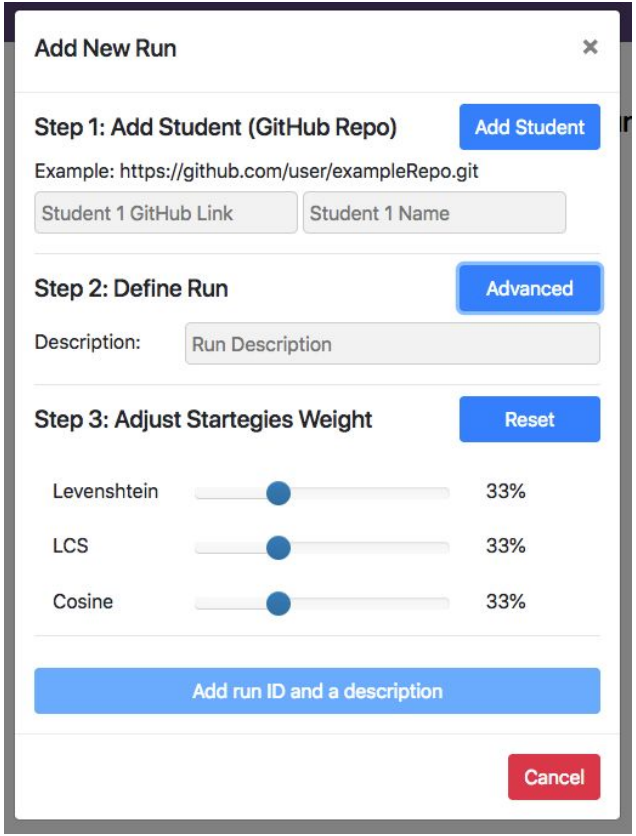
Customers must starting a new plagiarism checking in this user interface.

Start New Run

CS106-41: Plagiarism run framework

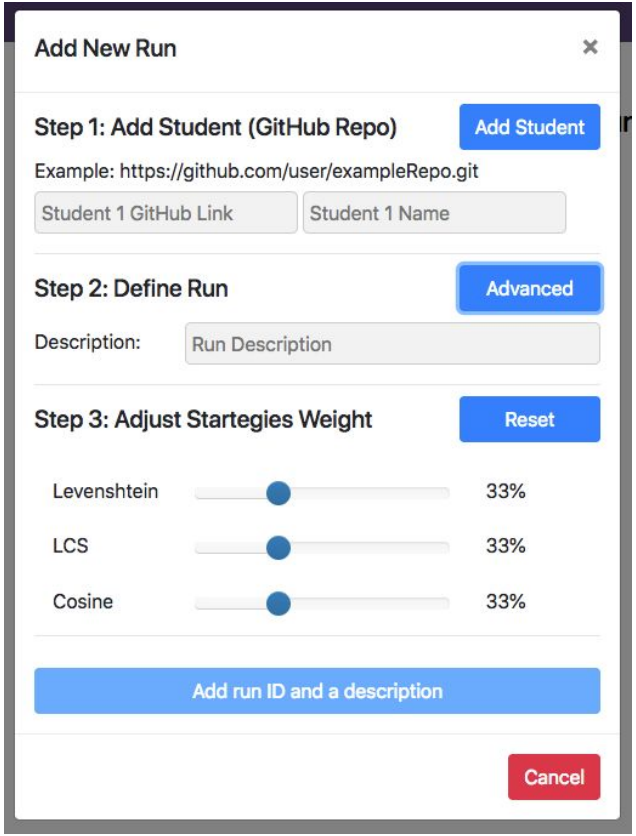
CS106-58: Plagiarism run service

CS106-71: Fix UI



Features

- Run service will be disabled if there are less than two students
- Must finish adding students then add description
- Users are able to adjust weight of strategies
- Information must be complete to start the run
- GitHub repo URL validation



View Reports

Another main and important functionality of the application.

Customers can get a report of the run they created in this user interface.

View Reports

CS106-32: Implement professors dashboard

CS106-46: A simple dashboard

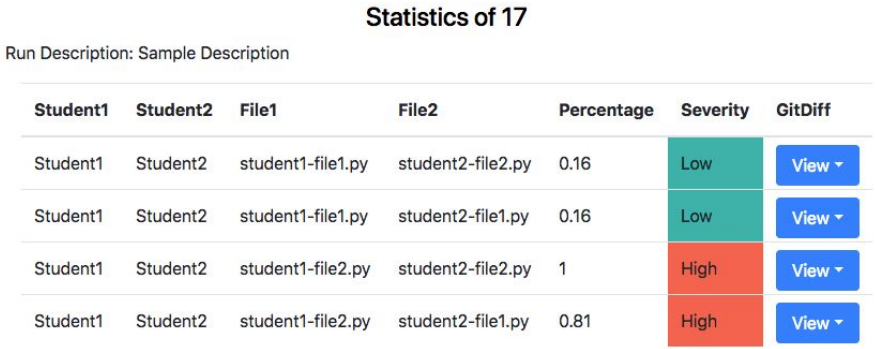
CS106-47: Reports services

CS106-48: An interactive dashboard

CS106-52: Pull Student submissions from git

CS106-41, 51, 53: Three complicated strategies

CS106-64: Call plagiarism and report backend endpoints



Statistics of 17

Run Description: Sample Description

Student1	Student2	File1	File2	Percentage	Severity	GitDiff
Student1	Student2	student1-file1.py	student2-file2.py	0.16	Low	View
Student1	Student2	student1-file1.py	student2-file1.py	0.16	Low	View
Student1	Student2	student1-file2.py	student2-file2.py	1	High	View
Student1	Student2	student1-file2.py	student2-file1.py	0.81	High	View

Admin Dashboard

Sample Admin credentials:

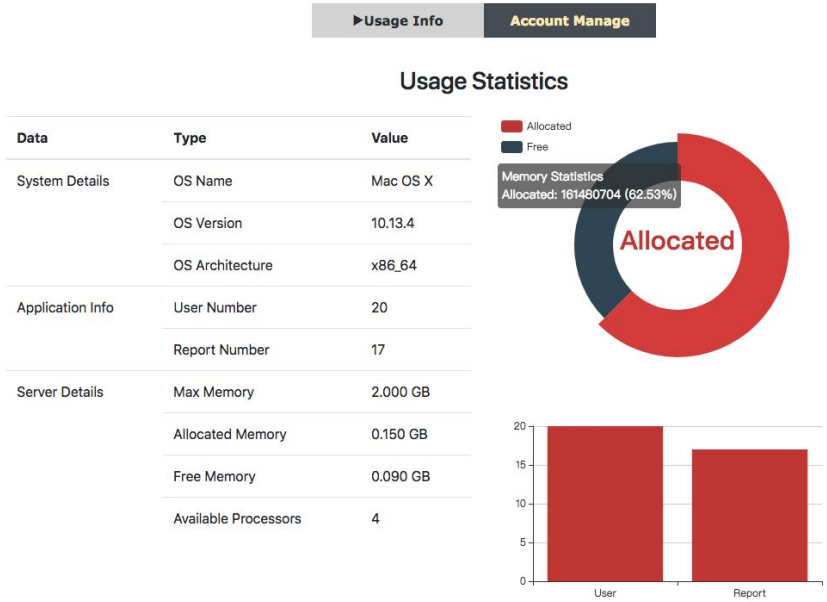
admin2@example.com
123456

Why customers need this?

An application must involve an administer who is responsible for getting usage statistics and managing users(blocking users).

Admin Dashboard

CS106-33: Implement admin's dashboard



Managing/Blocking Users

CS106-106

Job Quality

Defects Fixing

Sprint 1

CS106-21 *	React unable to call end point on aws	Bug
CS106-34 *	Front end refreshing automatically	Bug
CS106-36 *	Fix the confirm password	Bug

Sprint 2

CS106-59 *	Login test case impacted	Bug
CS106-62 *	List entity error	Bug
CS106-70 *	login uid bug	Bug
CS106-71 *	Fix UI	Bug

Sprint 3

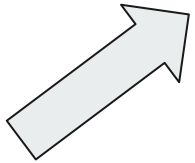
CS106-82 *	Fix to the weighted polynomial function.	Bug
CS106-106 *	Add restrictions to "add new run"	Bug

Defects Fixing

- CS106-158 No validation check to accept student repositories (Accepts fake repositories)
- CS106-161 Need to check whether the report object is null in js files.
- CS106-163 Exact same github url for 2 students, gives 85% plagiarism
- CS106-171 Empty test in Comparison Test class
- CS106-184 Unusual behavior of a system (log out issue)
- CS106-186 Add run functionality accepts any kind of repository -not only python
- CS106-204 Code review problems in testing classes
- CS106-206 Test case out unpredicted
- CS106-207 Maybe an issue in comparison algorithm

Feed back from TA

```
3   ------------------------------------------------
4   Sprint 2 Group Grade Team 106:
5   ------------------------------------------------
6
7   Functionality Expectations Achieved:
8   ------------------------------------------------
9
10  Comments on Functionality Expectations:
11
12  UI Comments:
13  1. The link ""View UI as non-wireframes"" does not work
14  2. Admin login does not work
15  3. UI needs prompts; labels and user messaging
16  4. Comparison UI (Add Run) has no labels or instructions for inputs.
```



```
3   ------------------------------------------------
4   Sprint 3 Group Grade Team 106:
5   ------------------------------------------------
6
7   Functionality Expectations Achieved:
8   ------------------------------------------------
9
10  Sets of Test Projects (out of 6): 6
11
12  Comments on sets of Test Projects: 6 sets provided ar
13
14  Comments on Functionality Expectations: "URL: http://
15  UI is clean and nice since Sprint 2. Need to tweak sc
16
```

Process and Teamwork

Work as a Team!

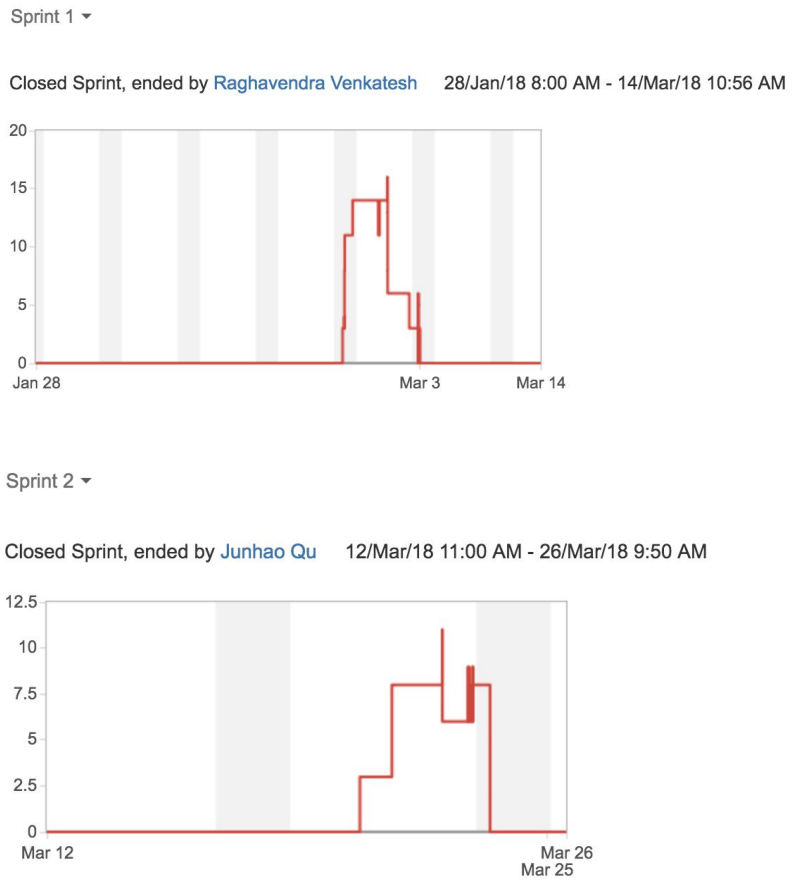
Ananta: GitDiff, two comparison strategies, MOSS training

AST Generation

Bharat: Back-end, end points, detection engine, dev enviror

Junhao: Front-end, one comparison startegy, paperwork

Raghu: Integration, end points and back end, use cases



Build & Test Automation

GitHub Master Branch

Technology Transfer

Hand Over to Clients

1. The application is well deployed on AWS and is able to access on internet. It is in good shape to deliver to customers.
2. We suggest customers to clone the code, deploy Database attributes, use maven to build and generate a jar file on local machine and deploy the jar file.