

# Integrity: Plagiarism Detector

Darshan Panse, Samanjate Sood, Shail Shah, Vaibhav Dave

# Introduction

System Functionality

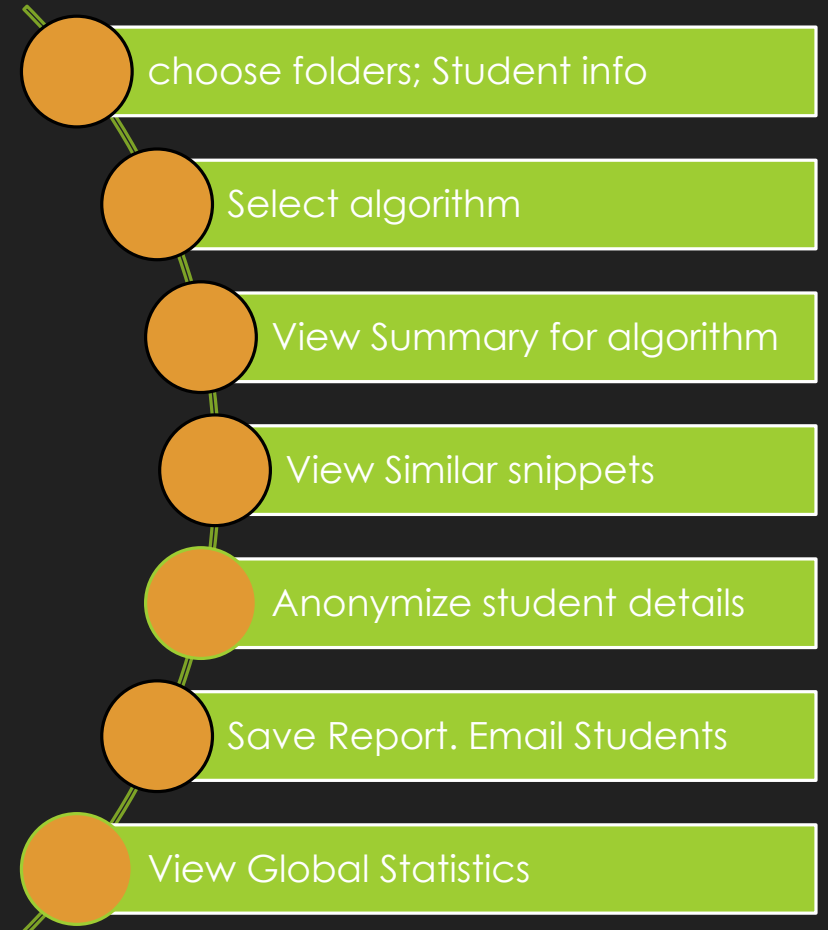
Job Quality

Process and Teamwork

Technology Transfer

# System Functionality

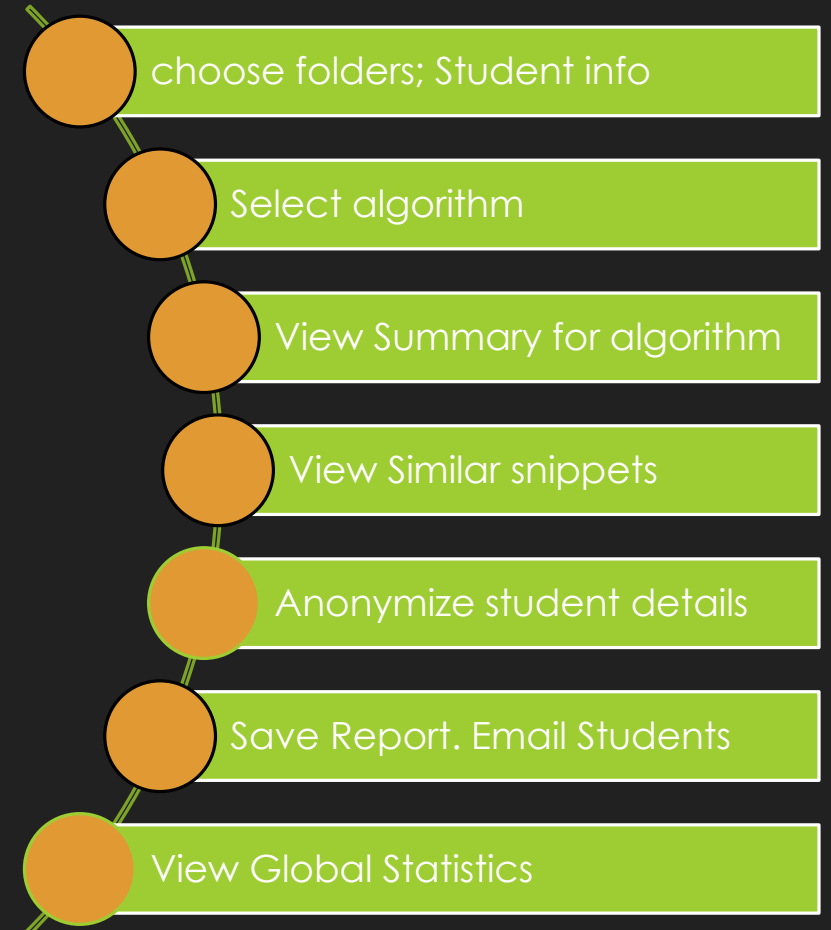
- Did the team accomplish a sufficient amount of functionality?
  - Yes, we delivered more than promised!



# System Functionality

○ Did the team achieve what it set out to do?

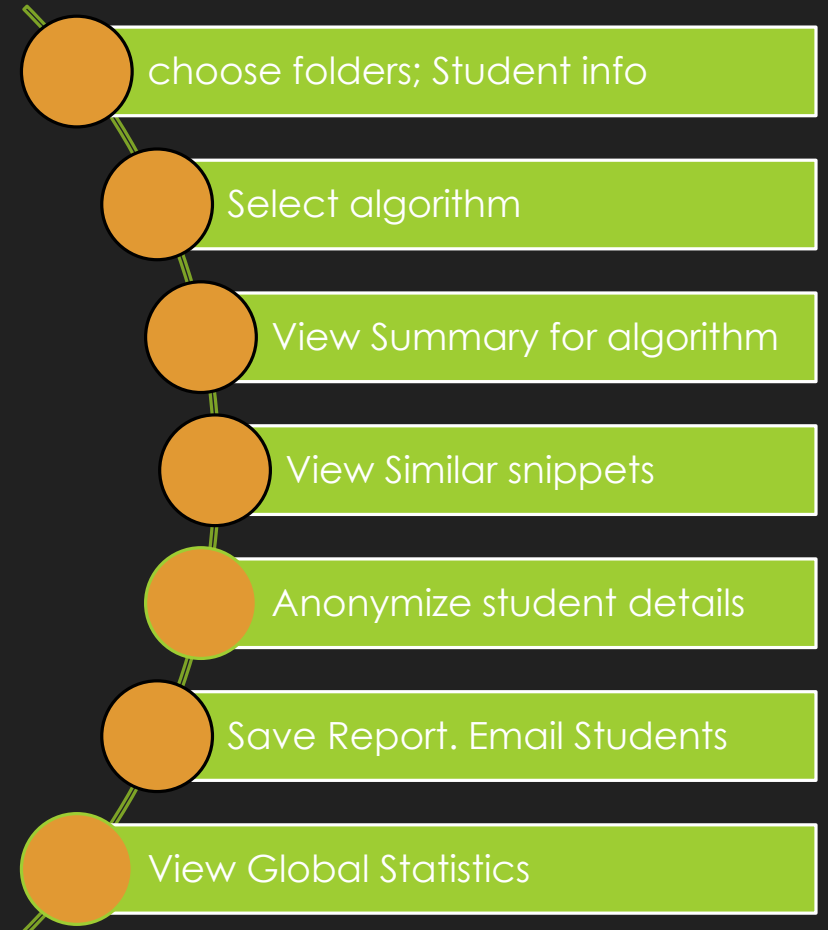
- Yes, we achieved more!



# System Functionality

○ Is it useful for the client?

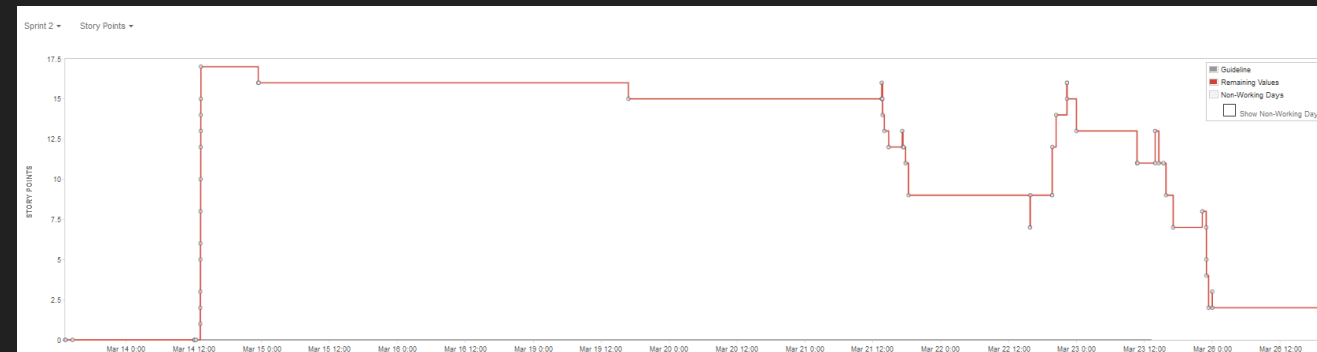
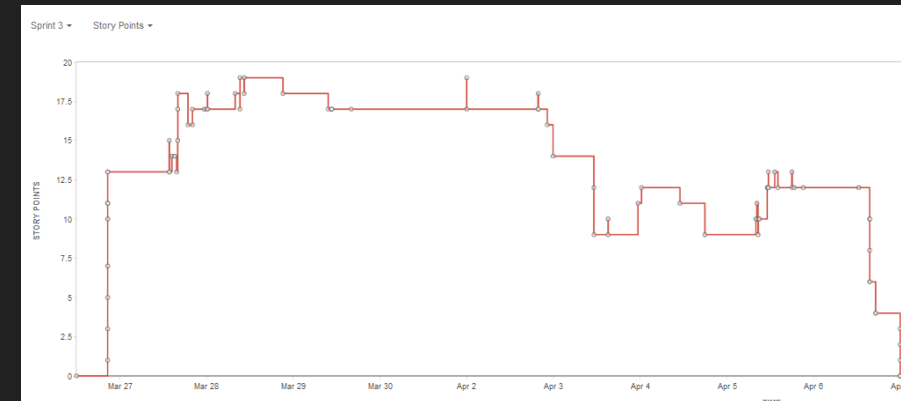
- We hope so!



# System Functionality

○ Is there evidence from the backlog/requirements to support these claims?

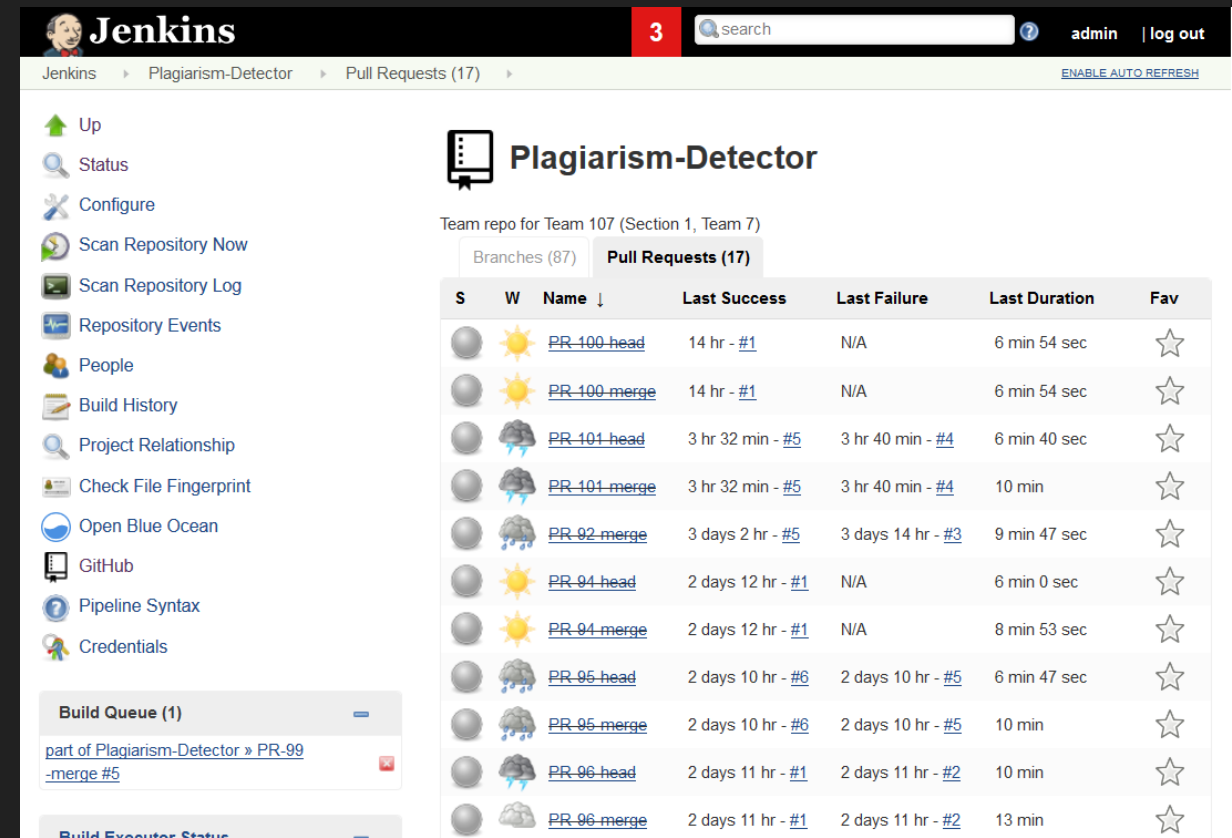
- Yes!



# Job Quality

○ Did the team present evidence that it did a quality job?

- Yes, Jenkins and SonarQube helped us maintain quality!



The screenshot shows the Jenkins web interface for the 'Plagiarism-Detector' job. The left sidebar contains navigation links such as 'Up', 'Status', 'Configure', 'Scan Repository Now', 'Scan Repository Log', 'Repository Events', 'People', 'Build History', 'Project Relationship', 'Check File Fingerprint', 'Open Blue Ocean', 'GitHub', 'Pipeline Syntax', and 'Credentials'. The main content area displays the job name 'Plagiarism-Detector' and a table of pull requests. The table has columns for 'S' (Status), 'W' (Weather icon), 'Name', 'Last Success', 'Last Failure', 'Last Duration', and 'Fav' (Favorite). The table lists 17 pull requests, with the first two being successful and the remaining 15 being failed. The 'Build Queue' section at the bottom shows a single build in progress: 'part of Plagiarism-Detector » PR-99-merge #5'.

S	W	Name ↓	Last Success	Last Failure	Last Duration	Fav
🟢	☀️	<a href="#">PR-100-head</a>	14 hr - <a href="#">#1</a>	N/A	6 min 54 sec	☆
🟢	☀️	<a href="#">PR-100-merge</a>	14 hr - <a href="#">#1</a>	N/A	6 min 54 sec	☆
🔴	☁️	<a href="#">PR-101-head</a>	3 hr 32 min - <a href="#">#5</a>	3 hr 40 min - <a href="#">#4</a>	6 min 40 sec	☆
🔴	☁️	<a href="#">PR-101-merge</a>	3 hr 32 min - <a href="#">#5</a>	3 hr 40 min - <a href="#">#4</a>	10 min	☆
🔴	☁️	<a href="#">PR-92-merge</a>	3 days 2 hr - <a href="#">#5</a>	3 days 14 hr - <a href="#">#3</a>	9 min 47 sec	☆
🟢	☀️	<a href="#">PR-94-head</a>	2 days 12 hr - <a href="#">#1</a>	N/A	6 min 0 sec	☆
🟢	☀️	<a href="#">PR-94-merge</a>	2 days 12 hr - <a href="#">#1</a>	N/A	8 min 53 sec	☆
🔴	☁️	<a href="#">PR-95-head</a>	2 days 10 hr - <a href="#">#6</a>	2 days 10 hr - <a href="#">#5</a>	6 min 47 sec	☆
🔴	☁️	<a href="#">PR-95-merge</a>	2 days 10 hr - <a href="#">#6</a>	2 days 10 hr - <a href="#">#5</a>	10 min	☆
🔴	☁️	<a href="#">PR-96-head</a>	2 days 11 hr - <a href="#">#1</a>	2 days 11 hr - <a href="#">#2</a>	10 min	☆
🔴	☁️	<a href="#">PR-96-merge</a>	2 days 11 hr - <a href="#">#1</a>	2 days 11 hr - <a href="#">#2</a>	13 min	☆

# Job Quality

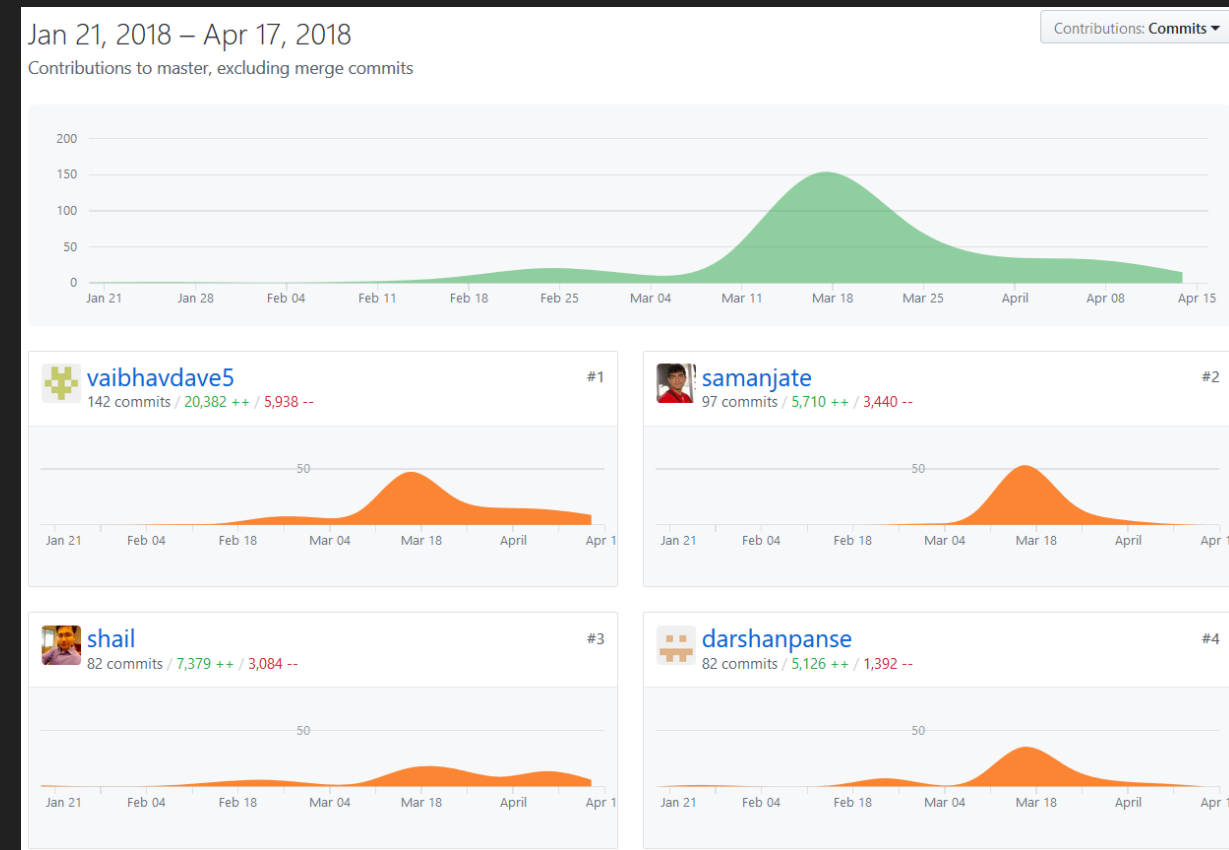
- Did the team improve performance with time?
  - Yes, we added more features and resolved bugs over time.
  - Listened to feedback given by others and made changes.



# Process and Teamwork

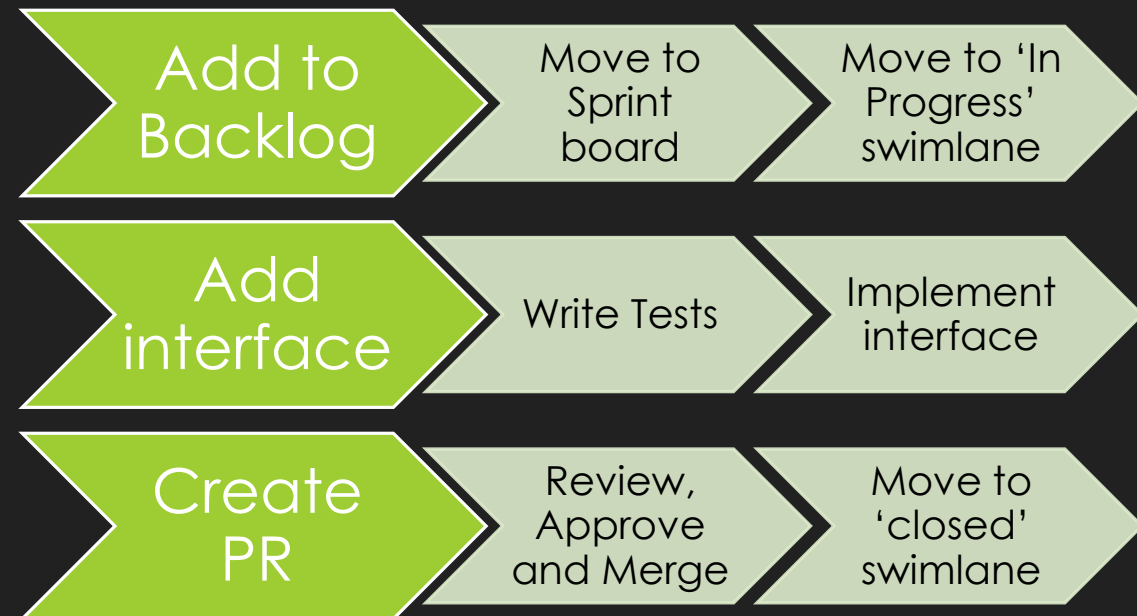
## ○ Did the team work as a team?

- Yes, we met in-person most of the times and helped each other with design and implementation



# Process and Teamwork

- Did the team use process well?
  - Yes, agile methodology was followed



# Process and Teamwork

## ○ Was the team able to automate the build, test, and promote processes?

- Yes, Jenkins was used for Continuous Integration.
- Merging was allowed only after build passed

The screenshot shows a GitHub pull request interface for the pull request titled "CS107-110: Adds filename and match% labels #94". The pull request is merged and shows a list of commits, a conversation tab, and a list of checks that passed. The commits include updates to similarity scores, backend support for new files, and the addition of filename and match percentage labels. The conversation tab shows a comment from "shail" stating "No description provided." and a review from "darshanpanse" approving the changes. The checks section shows three checks passed: "continuous-integration/jenkins/branch", "continuous-integration/jenkins/pr-head", and "continuous-integration/jenkins/pr-merge".

CS107-110: Adds filename and match% labels #94

Merged shail merged 6 commits into master from CS107-110 2 days ago

Conversation 0 Commits 6 Files changed 11 +127 -84

shail commented 2 days ago

No description provided.

darshanpanse and others added some commits 8 days ago

- Accommodated changes for similarity scores
- Updated the backend to support new file
- CS107-110: Adds file match percentages in report
- CS107-110: Fixes result percentages
- CS107-110: Adds filename and percentage match labels
- Merge branch "master" into CS107-110

shail requested review from samanjate, darshanpanse, and vaibhavdave5 2 days ago

shail changed the title from CS107-110: Adds filename match% labels to CS107-110: Adds filename and match% labels 2 days ago

darshanpanse approved these changes 2 days ago

shail merged commit aaa5404 into master 2 days ago

3 checks passed

- continuous-integration/jenkins/branch This commit looks good
- continuous-integration/jenkins/pr-head This commit looks good
- continuous-integration/jenkins/pr-merge This commit looks good

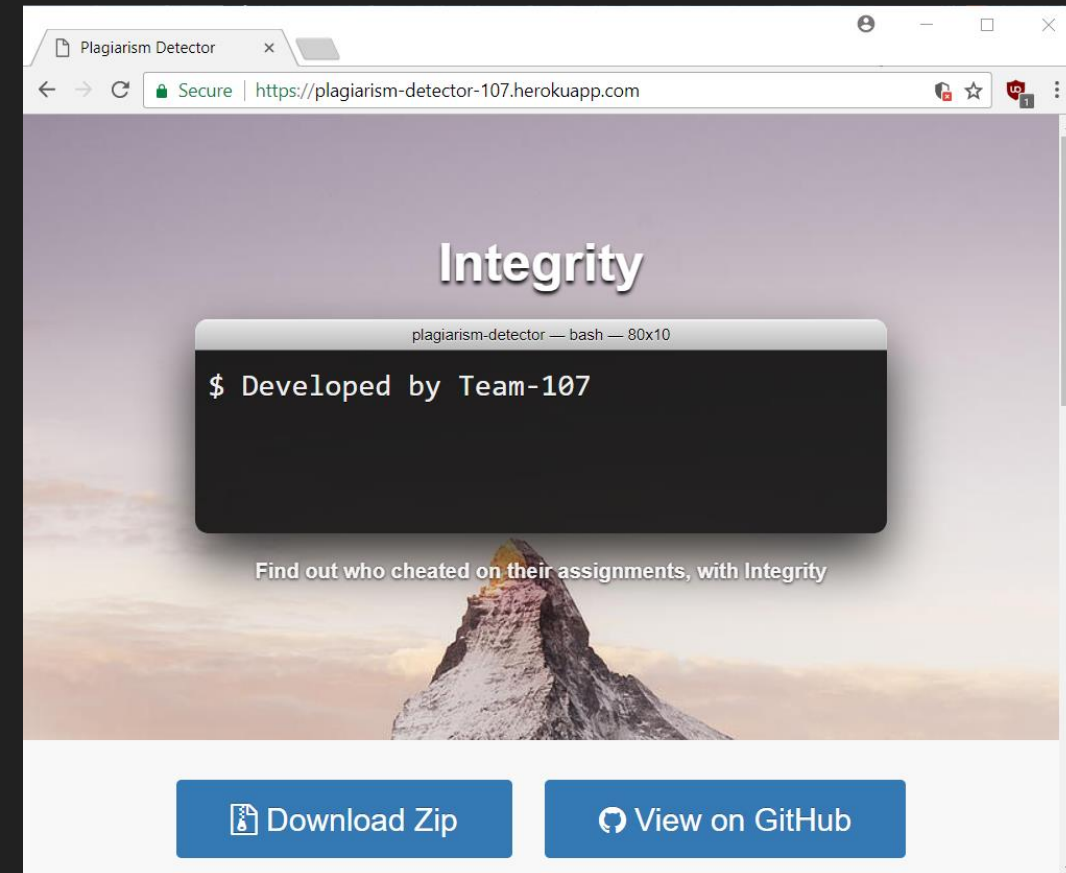
shail deleted the CS107-110 branch 2 days ago

# Process and Teamwork

- Did the team recognize short-comings? If so, how well did they work around the challenges?
  - Configuring Jenkins was a big hurdle
    - Faced a lot of downtimes with Jenkins
    - Alex helped us with the operations aspect of the project
  - Moss Integration was problematic
    - Did not get credentials for Moss for a long time
    - Finally got it on the 7<sup>th</sup> try!

# Technology Transfer

- Is the system in shape to be handed over to the client?
  - Yes, download the app today!  
<https://plagiarism-detector-107.herokuapp.com>



# Technology Transfer

## ○ Next Steps

- Continue refactoring the code
- Add support for other programming languages
- Provide support to the instructors and TAs

**Thank you!**