Plagiarism Detector Application

TEAM 101:

- AYUSH SHUKLA
- MONICA MALKANI
- PRACHI VED
- PREYANK JAIN

Functionalities achieved

- Source code of python files is compared for Plagiarism.
- Detection is done using four strategies:
    - Code Move Over
    - Variable Check
    - Modularity Check
    - Comment similarity
- The results above are aggregated using a weighted polynomial based on MOSS.

Additional Functionalities

- Results can be sorted based on Plagiarized/ Non-Plagiarized.
- Detailed report of the analysis can be downloaded.
- Report highlights the similar lines between files , gives detailed analysis for the same.
- The check can be run on individual submissions or class submission.

What we achieved ☺ or did not achieve ☹

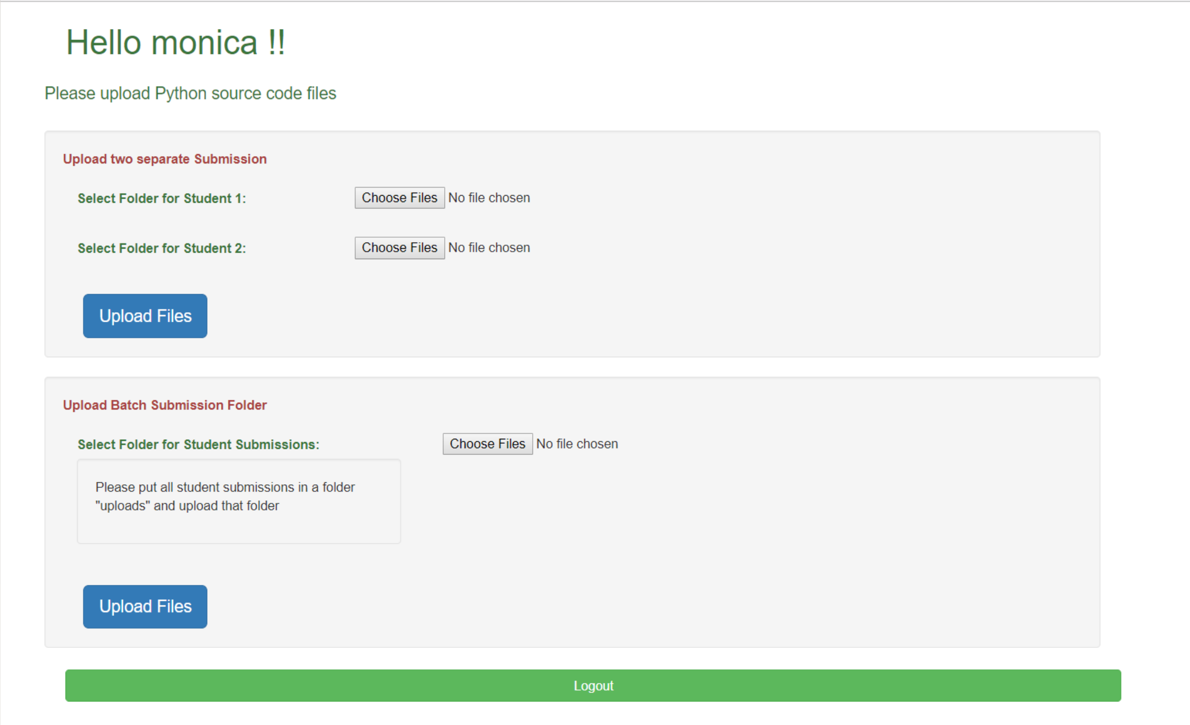
- Achieved:
  - ❖ All the functionalities mentioned in the previous slides were achieved.

- What is yet to come?
  - ❖ Along with the loader we plan to give number of files compared.
  - ❖ We plan on alerting the user when the run completes.
  - ❖ We plan to give user an option to display top 10 plagiarized files .

Use of our application

- Client can use this application to check plagiarism amongst python files.
- It would be useful as the results are approx. to MOSS standards.
- It even lets the user download the report which makes it easy for the user to comprehend the similarity in files.

Screenshots from application



Hello monica !!

Please upload Python source code files

Upload two separate Submission

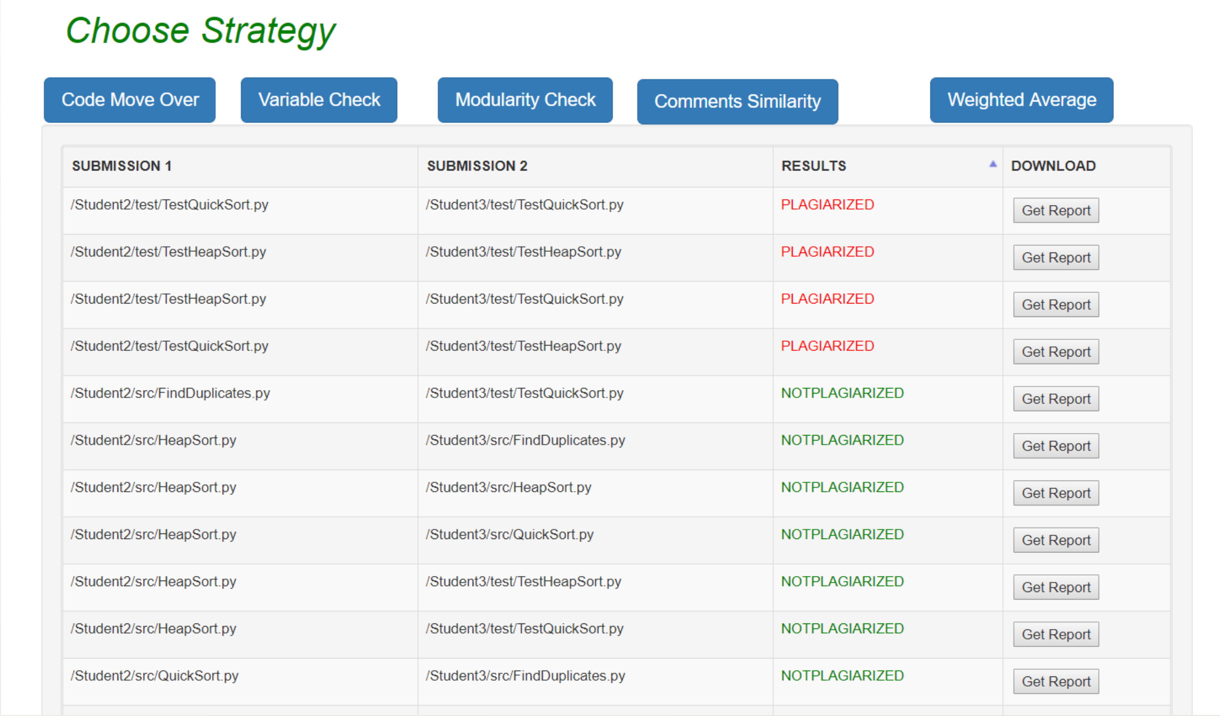
Select Folder for Student 1: Choose Files No file chosen

Select Folder for Student 2: Choose Files No file chosen

Upload Batch Submission Folder

Select Folder for Student Submissions: Choose Files No file chosen

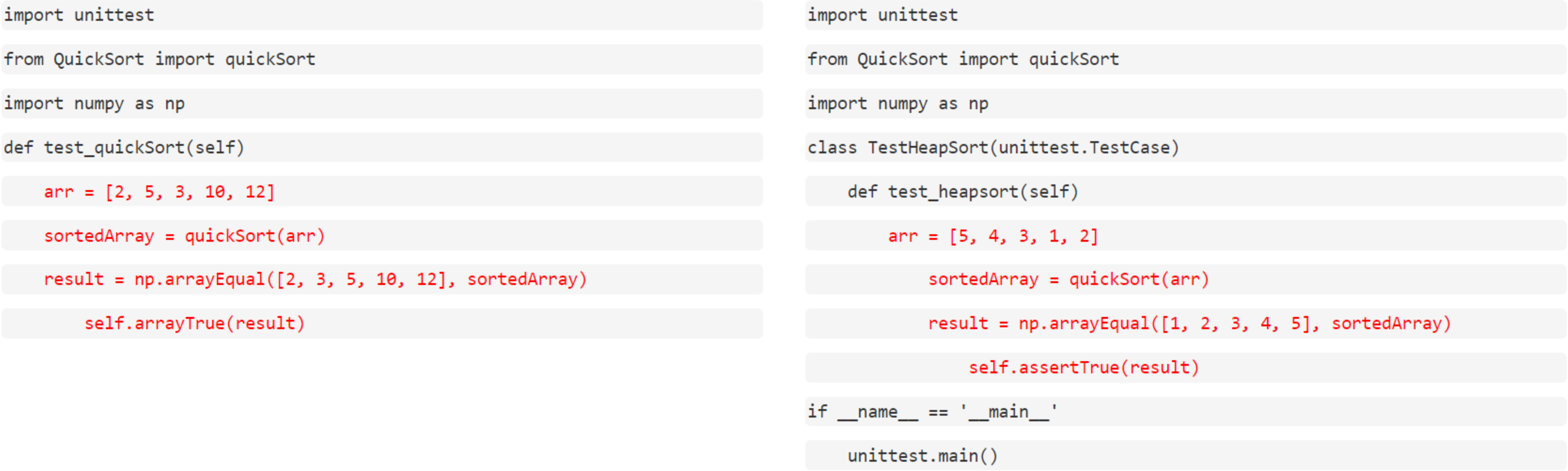
Please put all student submissions in a folder "uploads" and upload that folder



Choose Strategy

SUBMISSION 1	SUBMISSION 2	RESULTS	DOWNLOAD
/Student2/test/TestQuickSort.py	/Student3/test/TestQuickSort.py	PLAGIARIZED	Get Report
/Student2/test/TestHeapSort.py	/Student3/test/TestHeapSort.py	PLAGIARIZED	Get Report
/Student2/test/TestHeapSort.py	/Student3/test/TestQuickSort.py	PLAGIARIZED	Get Report
/Student2/test/TestQuickSort.py	/Student3/test/TestHeapSort.py	PLAGIARIZED	Get Report
/Student2/src/FindDuplicates.py	/Student3/test/TestQuickSort.py	NOTPLAGIARIZED	Get Report
/Student2/src/HeapSort.py	/Student3/src/FindDuplicates.py	NOTPLAGIARIZED	Get Report
/Student2/src/HeapSort.py	/Student3/src/HeapSort.py	NOTPLAGIARIZED	Get Report
/Student2/src/HeapSort.py	/Student3/src/QuickSort.py	NOTPLAGIARIZED	Get Report
/Student2/src/HeapSort.py	/Student3/test/TestHeapSort.py	NOTPLAGIARIZED	Get Report
/Student2/src/HeapSort.py	/Student3/test/TestQuickSort.py	NOTPLAGIARIZED	Get Report
/Student2/src/QuickSort.py	/Student3/src/FindDuplicates.py	NOTPLAGIARIZED	Get Report

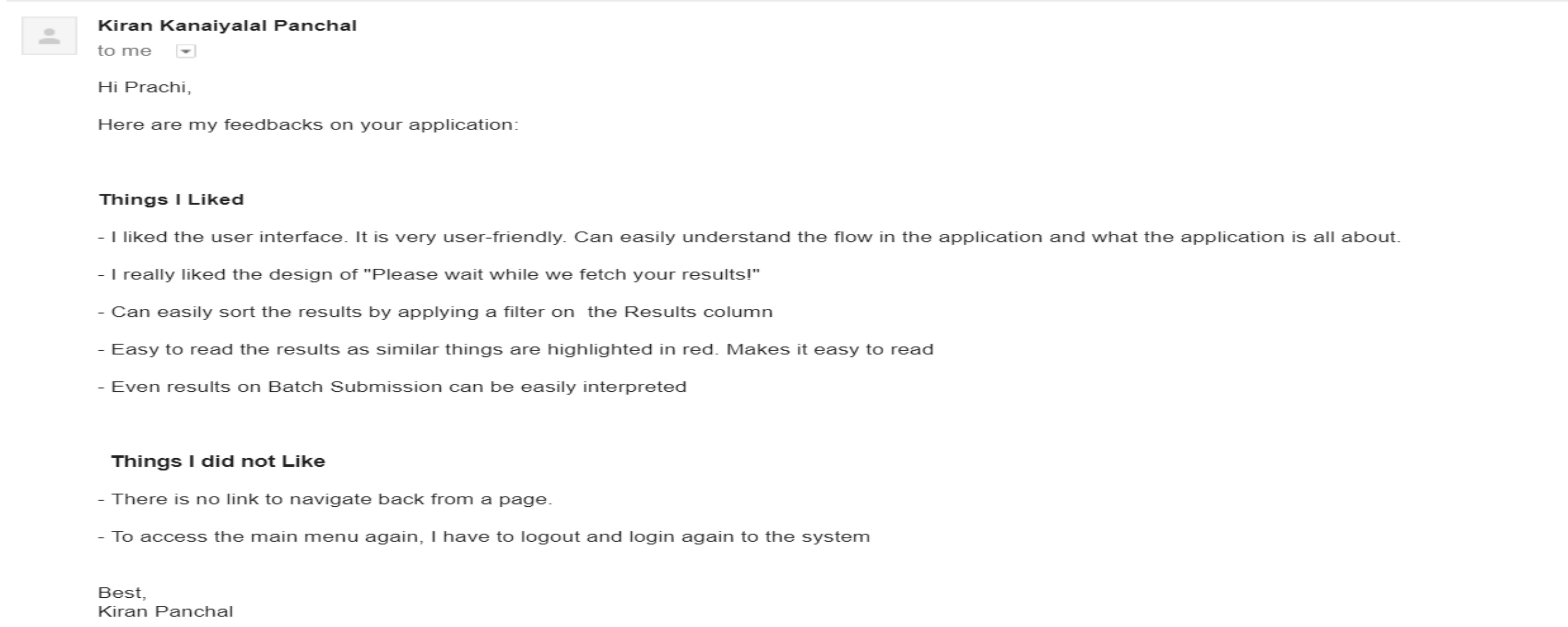
Plagiarism Detection Application



```
import unittest
from QuickSort import quickSort
import numpy as np
def test_quickSort(self)
    arr = [2, 5, 3, 10, 12]
    sortedArray = quickSort(arr)
    result = np.arrayEqual([2, 3, 5, 10, 12], sortedArray)
        self.arrayTrue(result)
```

```
import unittest
from QuickSort import quickSort
import numpy as np
class TestHeapSort(unittest.TestCase)
    def test_heapsort(self)
        arr = [5, 4, 3, 1, 2]
            sortedArray = quickSort(arr)
            result = np.arrayEqual([1, 2, 3, 4, 5], sortedArray)
                self.assertTrue(result)
if __name__ == '__main__'
    unittest.main()
```

User feedback-1



Kiran Kanaiyalal Panchal
to me

Hi Prachi,

Here are my feedbacks on your application:

Things I Liked

- I liked the user interface. It is very user-friendly. Can easily understand the flow in the application and what the application is all about.

- I really liked the design of "Please wait while we fetch your results!"

- Can easily sort the results by applying a filter on the Results column

- Easy to read the results as similar things are highlighted in red. Makes it easy to read

- Even results on Batch Submission can be easily interpreted

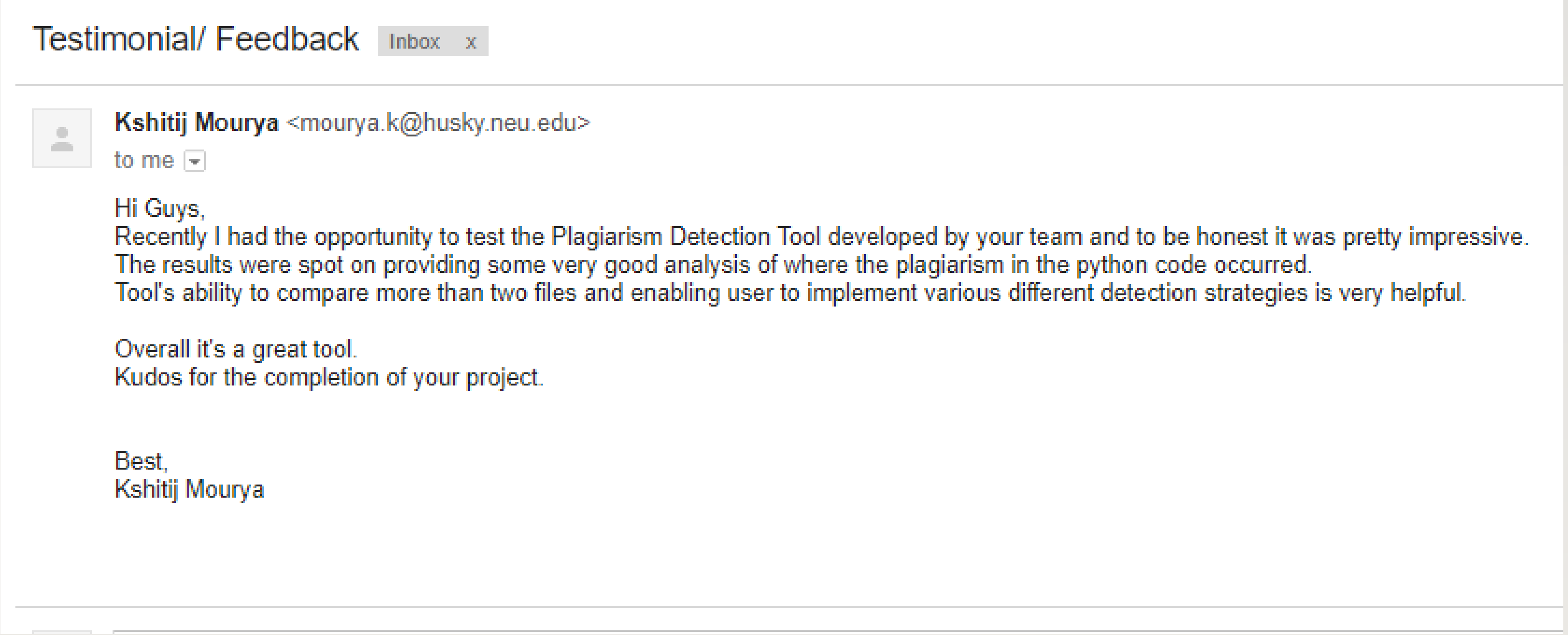
Things I did not Like

- There is no link to navigate back from a page.

- To access the main menu again, I have to logout and login again to the system

Best,
Kiran Panchal

User Feedback-2



Testimonial/ Feedback

Kshitij Mourya <mourya.k@husky.neu.edu>

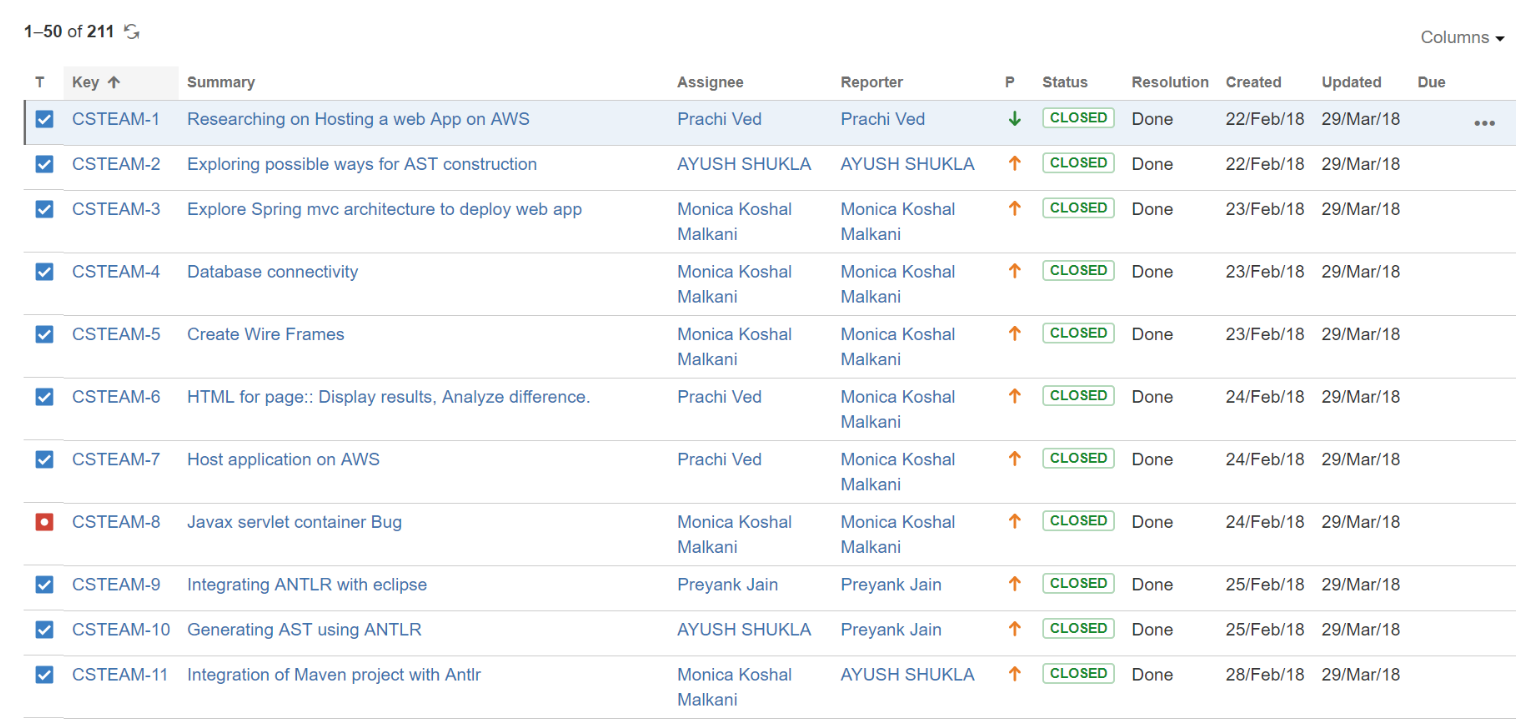
to me

Hi Guys,
Recently I had the opportunity to test the Plagiarism Detection Tool developed by your team and to be honest it was pretty impressive.
The results were spot on providing some very good analysis of where the plagiarism in the python code occurred.
Tool's ability to compare more than two files and enabling user to implement various different detection strategies is very helpful.

Overall it's a great tool.
Kudos for the completion of your project.

Best,
Kshitij Mourya

Jira



1–50 of 211

T	Key ↑	Summary	Assignee	Reporter	P	Status	Resolution	Created	Updated	Due
☑	CSTEAM-1	Researching on Hosting a web App on AWS	Prachi Ved	Prachi Ved	↓	CLOSED	Done	22/Feb/18	29/Mar/18	
☑	CSTEAM-2	Exploring possible ways for AST construction	AYUSH SHUKLA	AYUSH SHUKLA	↑	CLOSED	Done	22/Feb/18	29/Mar/18	
☑	CSTEAM-3	Explore Spring mvc architecture to deploy web app	Monica Koshal Malkani	Monica Koshal Malkani	↑	CLOSED	Done	23/Feb/18	29/Mar/18	
☑	CSTEAM-4	Database connectivity	Monica Koshal Malkani	Monica Koshal Malkani	↑	CLOSED	Done	23/Feb/18	29/Mar/18	
☑	CSTEAM-5	Create Wire Frames	Monica Koshal Malkani	Monica Koshal Malkani	↑	CLOSED	Done	23/Feb/18	29/Mar/18	
☑	CSTEAM-6	HTML for page:: Display results, Analyze difference.	Prachi Ved	Monica Koshal Malkani	↑	CLOSED	Done	24/Feb/18	29/Mar/18	
☑	CSTEAM-7	Host application on AWS	Prachi Ved	Monica Koshal Malkani	↑	CLOSED	Done	24/Feb/18	29/Mar/18	
🟥	CSTEAM-8	Javax servlet container Bug	Monica Koshal Malkani	Monica Koshal Malkani	↑	CLOSED	Done	24/Feb/18	29/Mar/18	
☑	CSTEAM-9	Integrating ANTLR with eclipse	Preyank Jain	Preyank Jain	↑	CLOSED	Done	25/Feb/18	29/Mar/18	
☑	CSTEAM-10	Generating AST using ANTLR	AYUSH SHUKLA	Preyank Jain	↑	CLOSED	Done	25/Feb/18	29/Mar/18	
☑	CSTEAM-11	Integration of Maven project with Antlr	Monica Koshal Malkani	AYUSH SHUKLA	↑	CLOSED	Done	28/Feb/18	29/Mar/18	

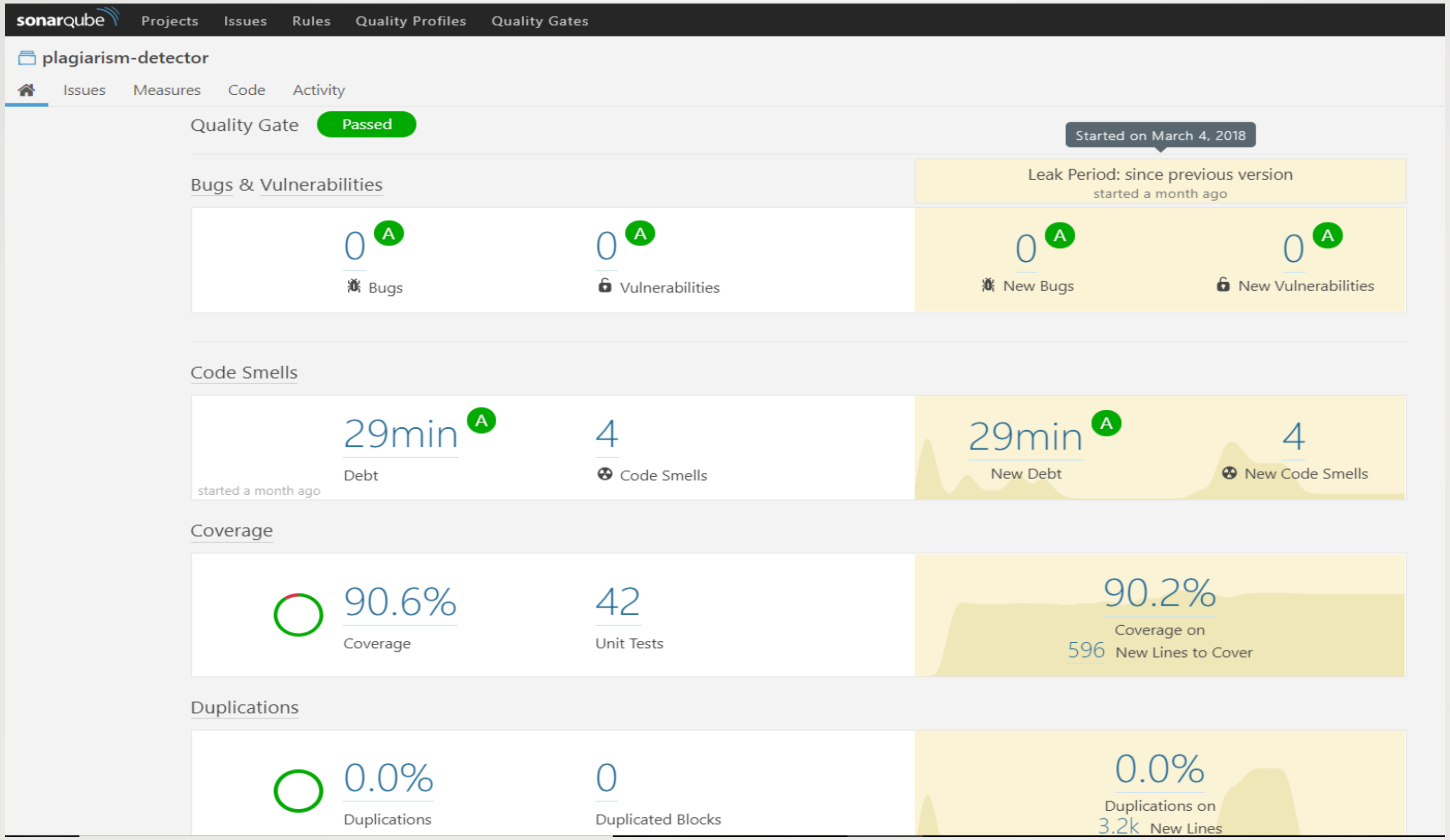
Quality of Application

- Quality was monitored using Sonar Qube.
- We achieved 90.6% code coverage.
- 42 unit tests were written.
- Maintainability and Security graded A
- 0% code duplications
- Very few ~4 code smells.

Team Performance.

- The team performance improved over time.
- Initially some of us were novices in Java but as the project progressed, familiarity with Java increased and therefore the productivity of the overall team increased.
- The work done in fourth week was more than the first week quantity and quality wise.

SonarQube screenshot



plagiarism-detector

Quality Gate Passed

Started on March 4, 2018

Leak Period: since previous version
started a month ago

Bugs & Vulnerabilities

0 A Bugs

0 A Vulnerabilities

0 A New Bugs

0 A New Vulnerabilities

Code Smells

29min A Debt

4 Code Smells

started a month ago

29min A New Debt

4 New Code Smells

Coverage

90.6% Coverage

42 Unit Tests

90.2%
Coverage on
596 New Lines to Cover

Duplications

0.0% Duplications

0 Duplicated Blocks

0.0%
Duplications on
3.2k New Lines

Teamwork

- Each member of the team took up individual task in each sprint.
- Other team members were updated about each other's progress via scrum meetings.
- Agile methodology was used.
- Progress was tracked via JIRA.
- GIT was used for version control.

- Automation was achieved with the help of Jenkins.
- After every pull request 3 builds were triggered:
    - Merge
    - Head
    - Branch
- Once the builds are successful the pull request is automatically merge to master and new deploy build is fired which deploys the code on AWS.
- Team members are notified about any new pull request that is created and about updates if any on their pull requests.

Shortcomings

- Learning the technicalities of spring boot , AWS , Jenkins.
- Internet resources were helpful.
- A lot was there to achieve.
- We prioritized our use cases to fit it in the desired time frame.
- Regular meeting and update sessions helped to achieve the end goal.

Technology Transfer

- We are yet to implement session management.
- Remaining functionalities work fine.
- A single user can satisfactorily use the system.

The future

- Implement sessions.
- Extend the system to support multiple programming languages.
- Alert pop ups in the system.
- Give real-time updates to the user on the screen.
- Have the system hosted on a proper domain.

Thank You! (Dhanyavaad) ☺

Cheers,
Team 101