P.E.S. UNIVERSITY

Department of Computer Science and Engineering

Session: Jan-May 2020

UE17CS355 – Web Technologies-II Lab

Project Phase – II

Test Report

Project Title: Stock Trading System

Section: 6C

Team Members:

Shubha M PES1201701540

Kritika Kapoor PES1201701868

Shrutiya M PES1201700160

UNIT TESTING

By: Shrutiya M PES1201700160

1. INTRODUCTION

   UNIT TESTING is a level of software testing where individual units/ components of a software are tested. The purpose is to validate that each unit of the software performs as designed. The unit tests done are automated, independent and consistent. The xUnit framework, unittest, a built-in testing module for implementing unit tests in Python is used.

2. OBJECTIVE

   Finding errors in an integrated module is much more complicated than first isolating the units, testing each, then integrating them and testing the whole.

   The goal of unit testing is to segregate each part of the program and test that the individual Flask APIs are working correctly according to our requirements. Unit testing provides numerous benefits including finding software bugs early, facilitating change, simplifying integration, providing a source of documentation, and so on.

3. TEST REPORT

Test case No.	Test Case Description	Expected Output	Actual Output	Test Result (Pass/Fail)
1.	Test Flask Set up API: /stocks/all	Response status code 200	Response status code 200	Pass
2.	Test submission throttle API: /check	i. Response Status code 200 ii. Company symbol 'ABBV' is returned when 'A' is entered as the input.	i. Response Status code 200 ii. Company symbol 'ABBV' is returned when 'A' is entered as the input.	Pass

3.	Test text loading in multistage download API: /about/description	i. Response status code 200 ii. 'Stock Trading System' in the response returned	i. Response status code 200 ii. 'Stock Trading System' in the response returned	Pass
4.	Test video loading in multistage download API: /about/video	i. Response status code 200 ii. Content type of the response returned is video/mp4	i. Response status code 200 ii. Content type of the response returned is video/mp4	Pass
5.	Test portfolio loading of user when logged in API: /user/portfolio Prerequisite: /api/login	i. Response status code 200 ii.'Transactions' logs in the response returned	i. Response status code 200 ii.'Transactions' logs in the response returned	Pass
6.	Test user balance loading when logged in API: /user/balance Prerequisite: api/login	Response status code 200	Response status code 200	Pass
7.	Test the prediction of stocks of a company API: /stocks/detail	i.Response status code 200 ii.'actual' and 'predicted' values in the response json	i.Response status code 200 ii.'actual' and 'predicted' values in the response json	Pass
8.	Test buying of stocks when user has enough money to buy	i.Response status code 200	i.Response status code 200	Pass

	API: /stocks/buy Prerequisite: /api/login	ii.'Not enough money' is not in the response	ii.'Not enough money' is not in the response	
9.	Test buying of stocks when user does not have enough money in his account API: /stocks/buy Prerequisite: /api/login	i.Response status code 200 ii.'Not enough money' is the response	i.Response status code 200 ii.'Not enough money' is the response	Pass
10.	Test selling of stocks when user has the selected stock in his account API: /stocks/sell Prerequisite: /api/login	i.Response status code 200 ii.'Null' is not returned as the response	i.Response status code 200 ii.'Null' is not returned as the response	Pass
11.	Test selling of stocks when user does not have the selected stock in his account API: /stocks/sell Prerequisite: /api/login	i.Response status code 200 ii.'Null' is returned as the response	i.Response status code 200 ii.'Null' is returned as the response	Pass

4. OBSERVATION and CONCLUSION

   All the test cases are passed successfully. Code coverage was used as a measure to describe the degree to which the source code of a program is tested by the test program. Coverage module in python gave a text based summary of code coverage as 78%. The application works well according to the requirements.

SYSTEM TESTING

By: Shubha M PES1201701540

1. INTRODUCTION

   System Testing is a level of testing that validates the complete and fully integrated software product. The purpose of a system test is to evaluate the end-to-end system specifications. Usually, the software is only one element of a larger computer-based system.
   Selenium is a suite of tools to automate web browsers for a variety of purposes and is used for system testing. Selenium WebDriver in particular is useful when writing browser-based tests.

2. OBJECTIVE

   Specific feature/function of a program can be perfectly working on its own, but it might not when they are connected to each other. The aim of System Testing is to ensure that the System will function properly when all features are bundled as a whole and meets the functional requirements.

3. TEST REPORT

Test case No.	Test Case Description	Expected Output	Actual Output	Test Result (Pass/Fail)
1.	Ensure that there is no user logged in initially	Username is null	Username is null	Pass
2.	Ensure the ‘buy’ button is disabled since there is no user	Return ‘False’ when ‘buy’ button is checked whether it is enabled	Returns ‘False’ when ‘buy’ button is checked whether it is enabled	Pass
3.	Ensure the ‘sell’ button is disabled since there is no user	Return ‘False’ when ‘sell’ button is checked	Return ‘False’ when ‘sell’ button is checked	Pass

		whether it is enabled	whether it is enabled	
4.	Ensure when the username appears when logged in	Returns 'True' for assertion that username is equal to logged in username.	Returns 'True' for assertion that username is equal to logged in username.	Pass
5.	Ensure buying stocks of companies works for the logged in user	The balance amount in the user account is reduced	The balance amount in the user account is reduced	Pass
6.	Ensure selling stocks of companies works for the logged in user	The balance amount in the user account is increased	The balance amount in the user account is increased	Pass
7.	Ensure the graph for the stock prices of last three months and future predicted prices is displayed	The innerHTML of the graph div element is present	The innerHTML of the graph div element is present	Pass
8.	Ensure the logged in user can view their previous transactions	'Transaction logs' present in the response	'Transaction logs' present in the response	Pass
9.	Ensure video is loaded from the server during multistage download	The innerHTML of the video element is present	The innerHTML of the video element is present	Pass
10.	Ensure submission throttle for suggesting in the search functionality works	The first letter of the returned response is equal to the searched query	The first letter of the returned response is equal to the searched query	Pass

4. OBSERVATION and CONCLUSION

   All the test cases passed successfully. The application works as per our requirements. The System functioned properly when all features are bundled as a whole and it met the functional requirements.

PERFORMANCE TESTING

By: Kritika Kapoor PES1201701868

1. INTRODUCTION

   Load testing is a type of non-functional testing. A load test is a type of software testing which is conducted to understand the behavior of the application under a specific expected load. Load testing is performed to determine a system's behavior under both normal and at peak conditions, which is achieved using Postman.

2. OBJECTIVE

   Postman is an open source API testing tool. It can work with any given endpoint even if it's Restful or XML based. It has an easy to use interface to make the request to the given endpoints. It offers a rich variety of features. Creating a collection from requests, extracting data from any response and storing in a variable are the least you can do with Postman.

3. TEST REPORT

Test case No.	Test Case Description	Expected Output	Actual Output	Test Result (Pass/Fail)
1.	Test load on Flask Set up with response time < 90s (10 iterations) API: /stocks/all	i. Response status code 200 ii. Response Time < 90s for all the 10 test iterations	i. Response status code 200 ii. Response Time < 90s for all the most iterations	Most Pass
2.	Test load on Flask Set up with response time < 30s (10 iterations) API: /stocks/all	i. Response status code 200 ii. Response Time < 30s for all the 10 test iterations	i. Response status code 200 ii. Response Time < 30s for less number of iterations	Most Fail

3.	Test submission throttle with 30 iterations of input fields. API: /check	i. Response Status code 200 ii. Company symbol 'ABBV' is returned when 'A' is entered as the input. for all the 10 test iterations	i. Response Status code 200 ii. Company symbol 'ABBV' is returned when 'A' is entered as the input. iv.Content-type header verified	All Pass
4.	Test text loading in multistage download for 10 iterations API: /about/description	i. Response status code 200 ii. 'Most new traders fail and lose money' in the response returned iii. Response time < 1 second iv.Content-type header verified	i. Response status code 200 ii. 'Most new traders fail and lose money' in the response returned iii. Response time < 1 second iv.Content-type header verified	All Pass
5.	Test text loading in multistage download for 30 iterations API: /about/description	i. Response status code 200 ii. 'Most new traders fail and lose money' in the response returned iii. Response time < 1 second	i. Response status code 200 ii. 'Most new traders fail and lose money' in the response returned iii. Response time < 1 second	All Pass

		iv.Content-type header verified	iv.Content-type header verified	
6.	Test video loading in multistage download for 10 iterations API: /about/video	i. Response status code 200 ii. Content type of the response returned is video/mp4 iii. Response Time < 1s for all the test iterations	i. Response status code 200 ii. Content type of the response returned is video/mp4 iii. Response Time < 1s for all the test iterations	All Pass
7.	Test video loading in multistage download for 30 iterations API: /about/video	i. Response status code 200 ii. Content type of the response returned is video/mp4 iii. Response Time < 1s for all the test iterations	i. Response status code 200 ii. Content type of the response returned is video/mp4 iii. Response Time < 1s for all the test iterations	All Pass
8.	Test portfolio loading of user when logged in for 10 iterations API: /user/portfolio Prerequisite: /api/login	i. Response status code 200 ii.'Transactions' logs in the response returned iii. Response Time < 1 second iv. Content-type header verified for	i. Response status code 200 ii.'Transactions' logs in the response returned iii. Response time < 1 second iv. Content-type header verified	All Pass

		all the test iterations	for all the test iterations	
9.	Test load on the prediction of stocks of a company for 10 iterations API: /stocks/detail	i.Response status code 200 ii.'actual' and predicted values in the response json iii. Response Time < 3s for all the test iterations	i.Response status code 200 ii.'actual' and predicted values in the response json iii. Response Time < 3s for all the test iterations	All Pass
10.	Test load on the prediction of stocks of a company for 30 iterations API: /stocks/detail	i.Response status code 200 ii.'actual' and predicted values in the response json iii. Response Time < 3s for all the test iterations	i.Response status code 200 ii.'actual' and predicted values in the response json iii. Response Time < 3s for all the test iterations	All Pass
11.	Test buying of stocks API: /stocks/buy giving appropriate body for 10 iterations	i.Response status code 200 ii.Response time < 2 seconds iii.Content-type is	i.Response status code 200 ii.Response time < 2 seconds iii.Content-type is	All Pass

		verified for all test cases	verified for all test cases	
12.	Test load on selling of stocks for 10 iterations API: /stocks/sell Prerequisite: /api/login	i.Response status code 200 ii.Response time < 2 seconds iii.Content-type is verified for all test cases	i.Response status code 200 ii.Response time < 2 seconds iii.Content-type is verified for all test cases	All Pass
13.	Test load on registering of 10 different users. API: /api/register	i.Response status code 200 ii.Response time < 1 second iii.Result is verified to be success for all test cases	i.Response status code 200 ii.Response time < 1 second iii.Result is verified to be success for all test cases	All Pass

4. OBSERVATION and CONCLUSION

   All the test cases passed successfully. The application works as per our requirements. The application handles load efficiently.