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: Book Recommendation System

Section: 6F

Team Members:

Harish P B PES1201701435

Prince Jha PES1201701608

Anagha Molkalmur PES1201701102

UNIT TESTING

By: Harish P B

1. Introduction

   We had designed a book recommendation system in which the user can add a book, check the list of books, get recommendations similar to the selected book. The unit testing is a piece of code that tests the logical unit of code in a tested system. unittest built-in testing module is used to implement unit tests in Python

2. Objective

   To verify if the backend API's functions as per the requirements and return appropriate status codes

3. Test Report

Test case No	Test case Description	Expected Output	Actual Output	Test Result (Pass/ Fail)
1	to test if the /recommend/{title} GET returns as per required and the len of the json object is size 10	status code 200 and no of books returned is size 10	status code 200 and no of books returned is size 10	Pass
2	to test /recommend/{title} on invalid delete method instead of get	status code 405 method not allowed	status code 405 method not allowed	Pass
3	to test /recommend/{title}	return	key error	Fail

	in case of title not encoded using URI component	status code 400 bad input	that results in status code 500 internal server error	
4	to test /books request on invalid put method instead of post	status code 405 method not allowed	status code 405 method not allowed	Pass
5	to test /books POST request taking index 0 and size 15 return OK	status code 200 and books from index 0 to 14 returned	status code 200 and size number of books returned	Pass
6	to test /books POST request if there are less than size no of books remaining in case of multistage download	no of remaining books appended to the previous list returned	no of remaining books appended to the previous list returned	Pass
7	to test /books POST request in case of invalid index	no books returned	no books returned	Pass
8	to test /add_book POST request in case of empty title	no book must be added into database	returns empty string and no book added into the database	Pass
9	to test /add_book POST request in case of empty	no book must be	returns empty	Pass

	authors	added into database	string and no books added into database	
10	to test /add_book POST request in case of correct title, authors and an average rating	status code 200, returns book details in form of json object and check the book added into the database	status code 200, returns the added book details in json format	Pass

4. Observation and Conclusion

```
hduser@bootcamp-VirtualBox:~/Documents/BookRe/Book-Recommendation-System/ml-reco
mmendation$ ls
app.py  mydb.db  __pycache__  unit_test.py
hduser@bootcamp-VirtualBox:~/Documents/BookRe/Book-Recommendation-System/ml-reco
mmendation$ python3 unit_test.py
.........F
======================================================================
FAIL: test_to_recommend_invalid_url (__main__.FlaskTestCase)
----------------------------------------------------------------------
Traceback (most recent call last):
  File "unit_test.py", line 28, in test_to_recommend_invalid_url
    self.assertEqual(response.status_code, 400)
AssertionError: 500 != 400

----------------------------------------------------------------------
Ran 10 tests in 4.967s

FAILED (failures=1)
hduser@bootcamp-VirtualBox:~/Documents/BookRe/Book-Recommendation-System/ml-reco
mmendation$
```

From the above unit tests, the application is working as per requirements except for a case of book recommendation in which the title sent is not URI encoded that results

in 500 internal server error status code that could be handled. I conclude that in addition to the React front end, the backend should also validate user input and return appropriate status codes.