**INDIAN INSTITUTE OF INFORMATION TECHNOLOGY, ALLAHABAD**

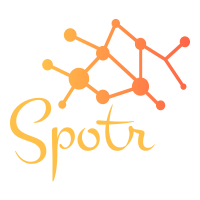
**Software Engineering**

Instructor: Abhishek Sir

TEST PLAN DOCUMENTATION

For

**SPOTR**



A PROFILE GENERATION

WEB APPLICATION

Group Members:

IIT2019127 MRIDUL MITTAL [iit2019127@iiita.ac.in](mailto:iit2019127@iiita.ac.in)

IIT2019145 PARAS AGRAWAL [iit2019145@iiita.ac.in](mailto:iit2019145@iiita.ac.in)

IIT2019146 KSHITIJ KUMAR [iit2019146@iiita.ac.in](mailto:iit2019146@iiita.ac.in)

IIT2019148 PIYUSH GURJAR [iit2019148@iiita.ac.in](mailto:iit2019148@iiita.ac.in)

1. **Introduction:**

This is the test plan for the Web Application Spotr , which is a profile generation application which searches the profile of the person with the help of username provided. A report of the results of the test performed on the application is displayed in this document.

**1.1 Test were run on following environment:**

● RAM: 8GB

● Processor: Intel i5 9th Gen

● OS: Windows 10 Home

● Tools: VS Code

1. **UNIT TESTING:**

Spotr is a web application and has two major components, the Frontend tests and backend tests.

For the Backend, we are checking each of the functions and the use cases, whether or not they are working properly as expected or not.

Also, we are checking for the edge cases (i.e. handling exceptional cases).

Similarly, in the Frontend we are doing the same for the components and the buttons, whether they are working fine as in ideal scenarios. Also, in case of invalid data it should respond accordingly.

|  |  |  |  |
| --- | --- | --- | --- |
| **Test No.** | **Test Name** | **Test file** | **Test Case** |
| 1. | url | 1.1 test\_homepage | Checks whether the url of the homepage is accessed by that function |
|  |  | 1.2 test\_login | Checks whether the url of the login is accessed by that function |
|  |  | 1.3 test\_register | Checks whether the url of the register is accessed by that function |
|  |  | 1.4 test\_logout | Checks whether the url of the logout is accessed by that function |
|  |  | 1.5 test\_github | Checks whether the url of the github is accessed by that function |
|  |  | 1.6 test\_codeforces | Checks whether the url of the codeforces is accessed by that function |
|  |  | 1.7 test\_instagram | Checks whether the url of the instagram is accessed by that function |
|  |  | 1.8 test\_twitter | Checks whether the url of the twitter is accessed by that function |
|  |  | 1.9 test\_reddit | Checks whether the url of the reddit is accessed by that function |
|  |  | 1.10 test\_sentiment | Checks whether the url of the Sentiment Analysis is accessed by that function |
| 2. | views | 2.1 homepage | Checks whether the html file is correctly accessed |
|  |  | 2.2 login | Checks whether the html file is correctly accessed |
|  |  | 2.3 test\_register | Checks whether the html file is correctly accessed |
|  |  | 2.4 test\_logout | Checks whether the html file is correctly accessed |
|  |  | 2.5 test\_github | Checks whether the html file is correctly accessed |
|  |  | 2.6 test\_codeforces | Checks whether the html file is correctly accessed |
|  |  | 2.7 test\_instagram | Checks whether the html file is correctly accessed |
|  |  | 2.8 test\_twitter | Checks whether the html file is correctly accessed |
|  |  | 2.9 test\_reddit | Checks whether the html file is correctly accessed |
|  |  | 2.10 test\_sentiment | Checks whether the html file is correctly accessed |
| 3. | database | 3.1 Codeforces | Checks whether the data in that database exists |
|  |  | 3.2 Instagram | Checks whether the data in that database exists |
|  |  | 3.3 github | Checks whether the data in that database exists |
|  |  | 3.4 Reddit | Checks whether the data in that database exists |
|  |  | 3.5 Twitter | Checks whether the data in that database exists |
| 4. | forms | 4.1 Valid Details | Checks whether the details are true |
|  |  | 4.2 Similar Password | Checks whether the password entered is similar to username. |
|  |  | 4.3 Mismatch | Checks if is password is written correctly or not |
|  |  | 4.4 Too Small | Checks for the password if it is too small |
|  |  | 4.5 Gmail invalid | Checks correctness of email entered |
|  |  | 4.6 Empty Fields | Checks for the empty fields in the forms |

**TEST NO. 1**

**TEST NAME: url**

This test checks whether the url defined for the webpages in “***url.py***” file and the functions defined for the webpages in “***views.py***” file is correctly matched or not. This means that a certain function should open that particular url only.

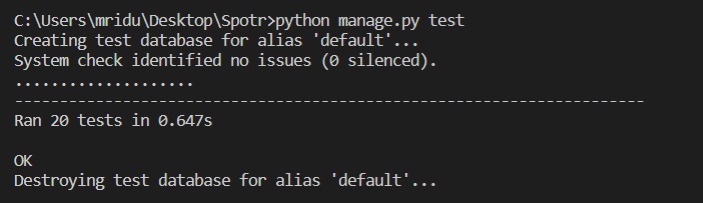
**TEST NO. 2**

**TEST NAME: views**

This test checks two things on the webpage.

Firstly, it checks whether the html webpage is opened or not. For that we have used “***response.status\_code,200***” which gives the output 200 if that html page is successfully opened on the web browser or not.

Secondly, to check whether the correct html page is opened or not we have checked it with the html file of that webpage.



**TEST NO. 3**

**TEST NAME: database**

This test checks whether the data searched by a user is stored in the database or not.

**TEST NO. 4**

**TEST NAME: forms**

1. Valid Details: It checks whether the information entered by the user is correct or not.
2. Similar password: It checks the username and password and makes sure that both are different.
3. Mismatch: It checks whether the password entered is similar to the registered password or not.
4. Too small: It checks for the length of the password. If it is too small then it shows error.
5. Gmail Invalid: It checks for the proper input of emails that have been entered or not. Like in case of missing ‘@’ it will show error.
6. Empty Fields: It checks for empty fields in the forms. For proper functioning all the fields must be properly filled.

