Test Strategy Document

Team number	***24***
Project Title	**Fake News Detection**
Document	**Testing Stategy**
Created By	**Parv Joshi** **Siddharth Jain** **Avani Gupta** **Apoorva Tirupathi**
Client	**Mounikesh Thota (Scrapshut)**
Mentor	**Mohit Chandra**

Scope

Testing activities:

- Testing functionality of links leading to signup, login and various dashboard components, check website, otp authentication, rate website.
- Testing working of email verification process and whether signup is possible without it.
- Testing encryption of passwords.
- Testing of database updates.

Test Approach

Process of testing:

- Setting up the test for a particular piece of an application (called the system under test)
- Performing the actual testing (interacting with the system under test)
- Observing the resulting behavior and checking whether expectations were met.

Testing Approach and Types of testing:

1. We performed load testing to check site performance under normal & peak conditions
2. We did unit testing of all frontend & backend components, including our deployed ML models.
3. We did security testing for user authentication & verification
4. We did localization testing by trying out the site on various browsers & platforms
5. We did interface testing of the entire UI and have tried to keep things as easy to understand as possible
6. We did integration testing after connecting the 3 different components of our project - the UI, database & the on-the- fly ML models

Testing Tools and Environment

- Automation and Test management tools needed for test execution.

- Figure out the number of open-source as well as commercial tools required, and determine how many users are supported on it and plan accordingly testing Tool.

Testing Tools:

1. **Postman**is a powerful tool for performing integration testing with your API. It allows for repeatable, reliable tests that can be automated and **used**in a variety of environments and includes useful tools for persisting data and simulating how a user might actually be interacting with the system.Postman was used to test and check the validity of the various get and post requests.
2. Jasmine (Testing Server For Angular)
3. Selenium (For Django)
4. Manual Checking

Testing Environment:

1. **For the test environment, a key area to set up includes:**
    - System and applications - Gitlab , Text editors - VSCode ,npm
    - Database server - SQLite
    - Front-end running environment – AngularJS
    - Backe-end running environment - DJANGO
    - Client operating system - Windows/Linux
    - Browser - Google Chrome/Mozilla Firefox
    - Hardware includes Server Operating system
    - Network - Localhost (frontend running on port 4200, backend running on port 8000)
2. **Network**

    Network set up as per the test requirement. It includes:

    - Internet setup
    - LAN Wifi setup
    - Private network setup

Restore strategy:

We use gitlab to incorporate the work of all team-members and this version control tool helps us keep track of the project commit by commit. In case of restoring requirements, we can do so through this exact tool.

Use Cases

1. Sign Up portal for users
2. Login Portal for users
3. Homepage
4. StartPage (Page After Log In)
5. Database Management
6. Check Website
7. Rate Website
8. Logout

Test Cases

1. Login portal for users
    a. Test by trying all variations of invalid/valid password and username.
    b. Working as of R1.
    c. Cannot Return To Logged In Page OnceLeft
2. Sign Up portal for user
    a. Test by emptying fields one by one
    b. Test by using not strong passwords
    c. Test by giving invalid Email Address
    d. Check If OTP is sent to email address
    e. Check If Correct OTP Is Used
    f. Check by giving previously used Email ID and Usernames
    g. Test by giving the wrong different password in the confirm password.
3. Homepage
    a. Test If All Hyperlinks Are Working
    b. Test Check Website Is Going and ML and scraping model is working.

c. Check If popup and correct message is seen after check website
d. Check if admin recieves maill after contact form is filledWorking as of R2.

4. StartPage
    a. Check If All Hyperlinks And Buttons Are Working
    b. Check If Check Website And Rate Website Are Working
    c. Check If Contact Form Is Working
5. Database Management
    a. After signup is complete, we need to check whether user information is uploaded to the server properly.
    b. To check if the Rated Websites rated by the user are stored correctly in the database.
    c. To check if the new details added to an incident during the progress of its solving, the details are appropriately added.
6. Check Website
    a. Check If Website is sent to backend and checked for its existence
    b. The link once verified should call upon Scraping And ML Model

7. Rate Website
    a. Check If Button Is Working
    b. Check If Form Details Are correctly filled In database
8. Logout
    a. Tested if the linking of logout with sign in page is correct or not.
    b. Working as of R1.

Summary

1. We performed load testing to check site performance under normal & peak conditions
2. We did unit testing of all frontend & backend components, including our deployed ML models.
3. We did security testing for user authentication & verification
4. We did localization testing by trying out the site on various browsers & platforms
5. We did interface testing of the entire UI and have tried to keep things as easy to understand as possible
6. We did integration testing after connecting the 3 different

components of our project - the UI, database & the on-the- fly

ML models