12. TESTING REPORT

The following section contains the report of the testing phase of the software.

Project Name UNIPREDICT

Project Type Web Application

Developer Aanchal Thakur

Language Python, javascript, css, html

Total Number of test cases 51

Number of test cases executed 51

Number of test cases Passed 50

Number of test cases Failed 1 – System not useful for Blind People

Positive Test Cases 39

Negative Test Cases 12

The code was tested both while developing as well as after development the following are some classes of tests that were executed-

12.1. Unit Testing

Testing of an individual software component or module is termed as Unit Testing. It is typically done by the programmer and not by testers, as it requires detailed knowledge of the internal program design and code.

The Code was developed in 3 separate parts-

1. AI Model developed using Jupyter Notebook
2. Web Front end was developed using VS Code
3. Backend Database was developed using MongoDB

Each of these parts underwent Unit Testing as follows:

12.1.1. AI Model

The AI Model was built and tested in Jupyter Notebook environment.

Boundary Value Testing

This type of testing checks the behaviour of the application at the boundary level.

Boundary Value Testing is performed for checking if defects exist at boundary values.

Boundary Value Testing is used for testing a different range of numbers. There is an upper

and lower boundary for each range and testing is performed on these boundary values.

12.1.2. Web Front End

The objective of this GUI Testing is to validate the GUI as per the business requirement. The expected GUI of the application is mentioned in the Detailed Design Document and GUI mockup screens.

12.1.3. Backend- Database

Incremental Integration Testing

Testing of all integrated modules to verify the combined functionality after integration is termed as Integration Testing.

Modules are typically code modules, individual applications, client and server applications on a network, etc.

After all the parts were created, they were then integrated as follows-

1. AI Component was integrated with the front end
2. Backend was integrated with the entire model

12.2.1. AI component + Front End

12.2.2. Data Base + Application

Negative Testing

Testers having the mindset of “attitude to break” and using Negative Testing they validate that if system or application breaks. A Negative Testing technique is performed using incorrect data, invalid data or input. It validates that if the system throws an error of invalid input and behaves as expected.

12.4. Functionality Testing

This type of testing ignores the internal parts and focuses only on the output to check if it is as per the requirement or not. It is a Black-box type testing geared to the functional

Non-Functional Testing

Non-Functional Testing involves testing of non-functional requirements such as Load Testing, Stress Testing, Security, Volume, Recovery Testing, etc. The objective of NFT testing is to ensure whether the response time of software or application is quick enough as per the business requirement.

12.6. Security Testing

Security Testing is done to check how the software or application or website is secure from internal and external threats. This testing includes how much software is secure from the malicious program, viruses and how secure and strong the authorization and authentication