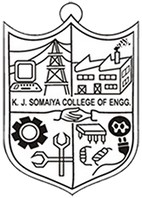
**K. J. SOMAIYA COLLEGE OF ENGINEERING**

**SOMAIYA VIDYAVIHAR**



**Intelligent System for detecting disinformation in Fake News**

Software Test Document

**PROJECT ID: 19**

**Project By:**

**Hritik Jaiswal (1714086)**

**Vedang Parasnis (1714100)**

**Heet Sakaria (1714110)**

*Under the guidance of*

**Dr Irfan Siddavatam**

**DEPARTMENT OF INFORMATION TECHNOLOGY A. Y. 2020-2021**

**Contents**

### 1 Introduction 4

|  |  |  |
| --- | --- | --- |
| 1.1 | System Overview . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 4 |
| 1.2 | Test Approach . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 4 |
|  | 1.2.1 Testing Methods . . . . . . . . . . . . . . . . . . . . . . . . . . | 4 |
|  | 1.2.2 Testing Strategies . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 5 |
| **2 Test** | **Plan** | **6** |
| 2.1 | Features to be Tested . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 6 |
| 2.2 | Features not to be Tested . . . . . . . . . . . . . . . . . . . . . . . . . . . | 6 |
| 2.3 | Testing Tools and Environment . . . . . . . . . . . . . . . . . . . . . . . . | 6 |
| **3 Test** | **Cases** | **7** |
| 3.1 | Relevancy of Scrapped Articles: TC-1 . . . . . . . . . . . . . . . . . . . . . . . . . . | 7 |
|  | 3.1.1 Purpose . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 7 |
|  | 3.1.2 Inputs . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 7 |
|  | 3.1.3 Expected Outputs & Pass/Fail criteria . . . . . . . . . . . . . . . . | 7 |
|  | 3.1.4 Test Procedure . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 7 |
| 3.2 | High Website traffic Scenarios: TC-2 . . . . . . . . . . . . . . . . . . . | 7 |
|  | 3.2.1 Purpose . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 7 |
|  | 3.2.2 Inputs . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 7 |
|  | 3.2.3 Expected Outputs & Pass/Fail criteria . . . . . . . . . . . . . . . . | 8 |
|  | 3.2.4 Test Procedure . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 8 |
| 3.3 | Classification report generated by the model: TC-3 . . . . . . . . . . . . . . . . . . . | 8 |
|  | 3.3.1 Purpose . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 8 |
|  | 3.3.2 Inputs . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 8 |
|  | 3.3.3 Expected Outputs & Pass/Fail criteria . . . . . . . . . . . . . . . . | 8 |
|  | 3.3.4 Test Procedure . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 8 |
| 3.4 | Lower Api Request Latency based on location: TC-4 . . . . . . . . . . . . . . . . . . | 8 |
|  | 3.4.1 Purpose . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 8 |
|  | 3.4.2 Inputs . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 8 |
|  | 3.4.3 Expected Outputs & Pass/Fail criteria . . . . . . . . . . . . . . . . | 9 |
|  | 3.4.4 Test Procedure . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 9 |
| 3.5 | Display of Analytical Reports:: TC-5 . . . . . . . . . . . . . . . . . . . . . . . . | 9 |
|  | 3.5.1 Purpose . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | 9 |

|  |  |  |  |
| --- | --- | --- | --- |
|  | 3.5.2 Inputs . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | | 9 |
| 3.5.3 Expected Outputs & Pass/Fail criteria . . . . . . . . . . . . . . . . | | 9 |
| 3.5.4 Test Procedure . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | | 9 |
| 3.6 | Working again anti crawling Websites: TC-6 . . . . . . . . . . . . . . . . . . . . . . . . . | | 9 |
|  | 3.6.1 Purpose . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | | 9 |
|  | 3.6.2 Inputs . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | | 9 |
|  | 3.6.3 Expected Outputs & Pass/Fail criteria . . . . . . . . . . . . . . . . | | 10 |
|  | 3.6.4 Test Procedure . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | | 10 |
| 3.7 | Client Application Interfaces: TC-7 . . . . . . . . . . . . . . . . . . | | 10 |
|  | 3.7.1 Purpose . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | | 10 |
|  | 3.7.2 Inputs . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | | 10 |
|  | 3.7.3 Expected Outputs & Pass/Fail criteria . . . . . . . . . . . . . . . . | | 10 |
|  | 3.7.4 Test Procedure . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | | 10 |
| 3.8 | Continuous Integration for new beta releases: TC-8 . . . . . . . . . . . | | 10 |
|  | 3.8.1 Purpose . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | | 10 |
|  | 3.8.2 Inputs . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | | 10 |
|  | 3.8.3 Expected Outputs & Pass/Fail criteria . . . . . . . . . . . . . . . . | | 11 |
|  | 3.8.4 Test Procedure . . . . . . . . . . . . . . . . . . . . . . . . . . . . . | | 11 |
| **4 Test Logs** | |  | **12** |

**5 Test Results 13**

1. **Introduction**
   1. **System Overview**

Disinformation detection system is a system for detecting disinformation in fake news. It is s a web based application that enables the users from field of information security, information law and also normal users to determine the disinformation in the news articles and provides awareness so as to prevent getting fallacious information about the given topic Disinformation identification System mainly works to identify the fabricated content in the news and provide adaptive based approach to adapt itself and thereby enabling itself to a state as to determine with confidence that the given information generated from this source is of high chance being fake using machine learning .We are following an hybrid based approach which is and ensemble and feedback based approach by analyzing transitive relations and two level ensemble approach so as the domain in determining fake news can work on variety of news as an input .We focus on providing accurate results as of proving scrapping data from various sources so as to provide wider category for classification. For Working in real time we are scraping articles from various renowned sources and thereby passing this raw data to our machine Learning models based on source generated by data and prediction provided as a analytical report to End User.

The Software Application is divided into 3 modules which need to be tested individually including the following:

* + - Articles Scrapping
      * Scrapped News Articles
      * Social Media tweets
    - End Users
      * Client End Articles validation
      * Analytical Reports
      * Non-Bot Validation

• Browser

* Extension Integration
  + - Database Instances

|  |  |  |
| --- | --- | --- |
|  | **–**  Parity of news articles |  |
| **–**  Partially or Jargon Entries |  |
|  |  |
| **1.2** | **Test Approach** |  |
| **1.2.1** | **Testing Methods** |  |

Black-box testing is a method of software testing that examines the functionality of an application without peering into its internal structures or workings. This method of test can be applied virtually to every level of software testing: unit, integration, system and acceptance. We are mainly using this type of testing to make sure that the application created is usable for the end client without any technical glitches.

Other Types of Testing performed by us include:

* + - **Unit Testing:** Individual Modules are tested to check their functionality is being performed appropriately. This is done in parallel with the development phase.
    - **Integration Testing:** Integration Testing is performed to expose defects in the interfaces and in the interactions between integrated components or systems.
    - **System Testing:** System testing confirms all system elements and performance are tested entirely.
    - **Compatibility Testing:** Compatibility testing to check whether your software is capable of running on different hardware, operating systems, applications, network environments or Mobile devices.
    - **Usability Testing:** The Data must be correct. The user interface should be easily accessible and content must be correct.

### 1.2.2 Testing Strategies

Alpha testing is a type of acceptance testing that is performed to identify all possible issues/bugs before releasing the software to clients or the end users. The focus of this testing is to simulate real users by using a black box and white box techniques. The aim is to carry out the tasks that a typical user might perform. Alpha testing is carried out in a lab environment and usually, the testers are internal testers of the organization. It is done early on, near the end of the development of the software, and before beta testing.

Beta Testing of the software is performed by ”real users” of the software application in a ”real environment” and can be considered as a form of external User Acceptance Testing. Beta version of the software is released to a limited number of end-users of the software to obtain feedback on the quality. Beta testing reduces failure risks and provides increased quality of the product through customer validation. Direct feedback from customers is a major advantage of Beta Testing. This testing helps to tests the product in customer’s environment.

# Test Plan

The application will be tested as per the completion of the given modules.

|  |  |  |  |
| --- | --- | --- | --- |
| Sr No. | Deliverable Name | Author | Reviewer |
| 1 | Software Testing Document | Hritik Jaiswal | Project Manager |
| 2 | Test Plan | Vedang Parasnis | Project Manager |
| 3 | Functional Test Cases | Hritik Jaiswal | Developer |
| 4 | Logging Defects | Test Team | Test Lead |
| 5 | Daily/Weekly Status | Test Team | Project Manager/Test Lead |
| 6 | Test Closure Report | Heet Sakaria | Project Manager |

## Features to be Tested

Software modules that need to be tested are as follows:

* + - Relevancy of Scrapped Articles: TC-1
    - High Website traffic Scenarios TC-2
    - Classification report generated by the model: TC-3
    - Lower Api Request Latency based on location TC-4
    - Display of Analytical Reports: TC-5
    - Working against anti crawling Websites: TC-6
    - Client Application Interfaces: TC-7
    - Continuous Integration for new beta releases: TC-8

## Features not to be Tested

Features that need not be tested are as follows:

* + - Cross platform support
    - Security

## Testing Tools and Environment

As soon as the development is completed for individual modules, the testers are to be ready with the test cases and start with its execution. Manual approach for creating

the software will be applied. Each Module is developed separately using HTML, CSS,

Typescript, Python ,Postman, Curl, RedisCLI or MongoCLI Aws SDK as its developing language and framework along with additional APIs and requirements as and when required. This developed application must be integrated together using version control software. For manual testing, Tester creates different environments and solves the issues using various manual as well as automated tools available for Python and Flask.

# Test Cases

## Relevancy of Scrapped Articles: TC-1

### Purpose

To identify and generate most relevant articles or tweets based on user area of interest and pass the corresponding input to the machine learning model and generate corresponding analytical report.

### Inputs

News Article or relevant News Headline Tags provided by End User.

### Expected Outputs & Pass/Fail criteria

Input by the user and corresponding news articles generated will be validated to the corresponding user news or headlines tags and then be processed in the machine learning model to generate corresponding analytical reports else error message will be displayed on screen

### Test Procedure

This module needs to be tested manually.

**Constraints:** A user can provide news tags or articles only in English Language.

## High Website traffic Scenarios: TC-2

### Purpose

To maintain faster and less delay to the end user once news tag requested by the user and to analyze how the application perform after scrapping of copious number of articles and then providing result to end User in analytical format without request timeout even in high traffic by using proper load balancing

### Inputs

News Article or relevant News Headline Tags provided by End User.

### Expected Outputs & Pass/Fail criteria

Appropriate action will be performed for request response timeouts and if timeout display the corresponding server timeout issues else provide the result which is intended to user

### Test Procedure

This module needs to be tested manually or through automated record and play tools using Postman Api testing tool.

**Constraints:** The user must not be a Bot and have verified captcha for application domain.

## Classification report generated by the model: TC-3

### Purpose

To test the same model with hyper parameter tuning and generate a precision recall and F1 score values.

### Inputs

Hyper parameters

### Expected Outputs & Pass/Fail criteria

If current model generate better accuracy than the preciously trained model on different parameters then current parameters are saved else discarded

### Test Procedure

This module needs to be tested manually or through automated record.

**Constraints:** None

## Lower Api Request Latency based on location: TC-4

### Purpose

To manage and provide best possible AWS EC2 Instance having proximity to the client so as to reduce latency in response and faster response to the client.

### Inputs

None.

### Expected Outputs & Pass/Fail criteria

Appropriate action will be performed for lower latency regions if lower latency then we would provide a lower latency option so as to provide a faster response else try to select the closest proximity instance to the client

### Test Procedure

This module needs to be tested manually or through automated record and play tools.

**Constraints:** The Admin must have multiple parallel EC2 Instances available.

## Display of Analytical Reports: TC-5

### Purpose

To provide best possible and relevant output to the end user in an user friendly analytical form Also, the generated reports provide correct and specific results based on user query.

### Inputs

Corresponding news articles tags and headlines

### Expected Outputs & Pass/Fail criteria

If proper response is generated during the process corresponding analytical report will be displayed else error report will be displayed to the end user.

### Test Procedure

This module needs to be tested manually or through automated testing tools.

**Constraints:** The client must have provided proper search query to the server

## Working against anti crawling Websites: TC-6

### Purpose

How crawler performs on anti-crawling website sources and which action is taken for such websites

### Inputs

None

### Expected Outputs & Pass/Fail criteria

None

### Test Procedure

This module needs to be tested manually or through automated robot tools

**Constraints:** None.

## Client Application Interfaces: TC-7

### Purpose

To provide error free and user friendly application environment so as the user can be able to get all the necessary reports without any point of failure.

### Inputs

None

### Expected Outputs & Pass/Fail criteria

The analytical reports will be displayed on success else corresponding error actions will be shown to the end user

### Test Procedure

This module needs to be tested manually or through automated testing tools.

**Constraints:** The End-user must have latest browser version installed in the system.

## Continuous Integration for new beta releases: TC-8

### Purpose

To manage future releases of application so as to maintain contiguous integration for upcoming release and provide bug free and better application support to the newer version as compared to earlier version.

### Inputs

None

### Expected Outputs & Pass/Fail criteria

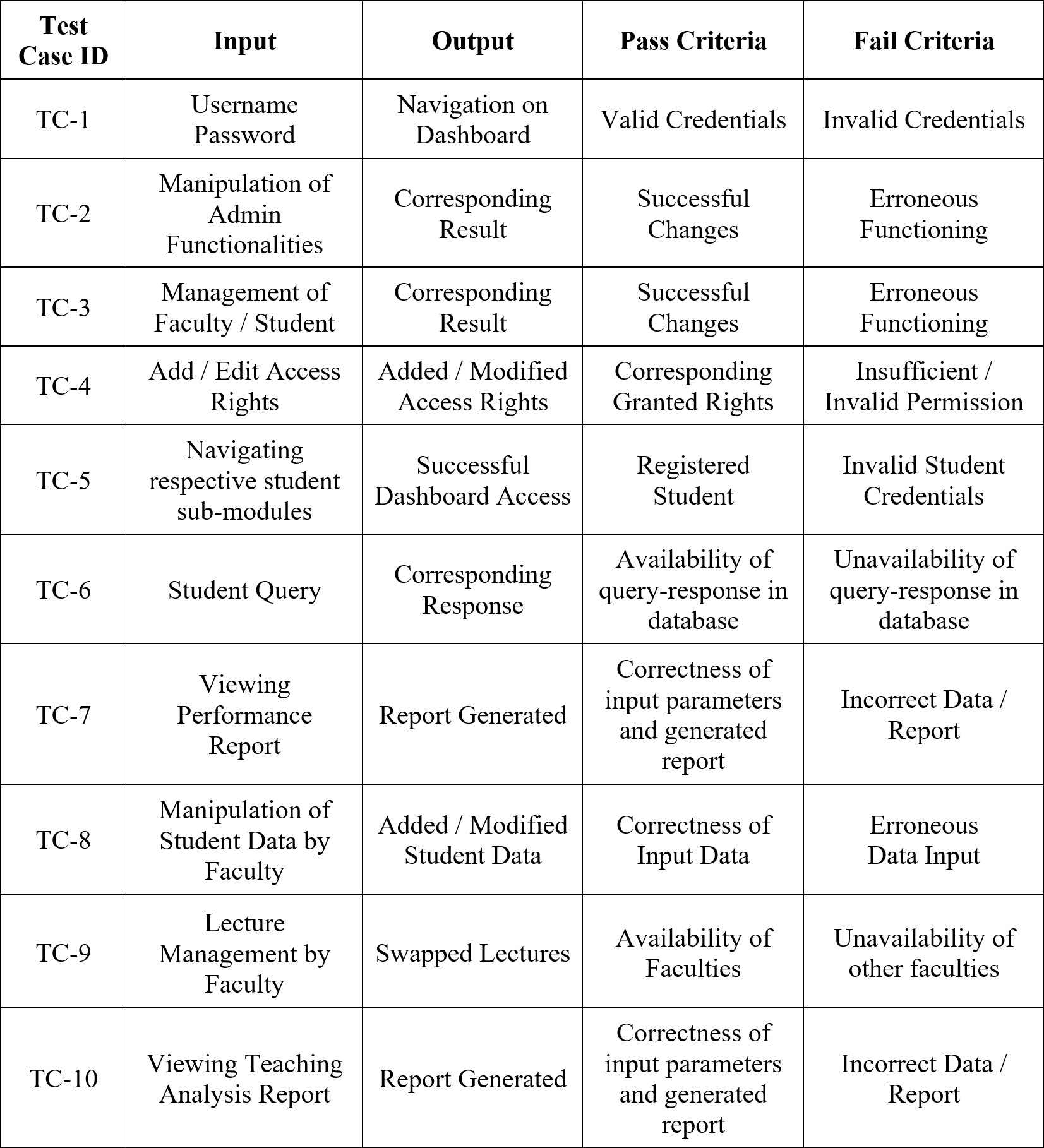
Further application releases are well maintained bug free and also backward comaptable

### Test Procedure

This module needs to be tested manually or through automated tools. We utilize CI/CD tools like TravisCI or CircleCI using version controller like GitHub.

**Constraints:** None

# Test Logs



1. **Test Results**

