Software Testing Report

NYC Restaurant Evaluation

Student Names

Table of Contents

[1.0 Unit Tests 3](#_Toc49779837)

[2.0 Coverage Report 4](#_Toc49779838)

[3.0 Requirements Acceptance Testing 5](#_Toc49779839)

# Unit Tests

| **No** | **Test Case** | **Expected Results** | **Actual Results** |
| --- | --- | --- | --- |
| **1.** | **User Registration** | User is registered successfully. | User Registered Successfully.  User did redirect to Login Page |
| 2 | **User Authentication** | User is authenticated successfully. | User is authenticated Successfully with advance authentication. |
| 3 | **Opening Role Selection** | Role selection window is displayed. | Role Section window open and in the drop down the role data were there with different option. |
| 4 | **Role-Based Dashboard Creation** | The corresponding dashboard is displayed. | The dashboard displays with various option for data analytic. |
| 5 | **Restaurant Search (Restaurant Owner):** | Search functionality works correctly. | Search functionality was working fine but with some special character (!@#$) the functionality is not working. |
| 6 | **Data Analysis (Data Analyst):** | Data analysis produces accurate results. | The application is analysed the data properly and the result were accurate.  The application was taking time in proceed the analysis phase of data. |
| 7 | **Data Visualization (Public User):** | Data visualization is displayed with correct statistics. | The application is giving different visualization of statistic with various data structure. The bar and pie chart were visualized accurately. |

# Coverage Report

A description of the coverage of your unit tests, including how you evaluated coverage (function, statement, branch, condition)

**Registration:**

The registration function is used to validate suer if the user gave the data according to the requirement if the data is not valid or incomplete the system return the exception message to provide the correct data else the system will proceed the request of registration so the user will proceed to the login for data analytics.

**AuthenticationUser(userId):**

Authentication is advance technique which the system ensures the user is not involved in any suspected activity of any cyber attach will gone be conduct from the user who is login to the system. The system will generate code which will sent to the user and if user put the code correctly then the system will proceed the request and pass the authentication layer of the system.

**Login (id, password):**

The login module is general module which will validate user Id and password and the system will proceed to the dashboard.

**GetRole (ID):**

The function is used to fetch the data from the database with respect to it’s id. The selected roles data will be fetched, and the user will be able to perform various operation according to its role.

**DataAnalytic(type):**

The data analytic is the module which has various method that perform different type of analytic. The user provides the data which the system examines and analysis. On the basic of data analytic the system will provide the statistics of the dataset provided by the user.

**DataVisualization(type):**

The data visualization is the module which gave it different type of visualization. The function contains the type of the visualization and the data which was analysis in the analysis part will Gona visualize the data in different type of chart and visualize techniques.

Code coverage is assessed using tools like coverage to determine the extent to which unit tests exercise your code. It provides insights into statement coverage (lines executed), branch coverage (decision branches tested), function coverage (functions invoked), and condition coverage (different conditions tested). A coverage report highlights areas with low coverage, indicating potential gaps in testing. The objective is to achieve high coverage, ideally close to 100%, to thoroughly test code paths and enhance code reliability and quality.

**Function Coverage:**

This metric measures the percentage of functions or methods in your code that have been executed by your unit tests. In your codebase:  
Function Coverage is 82%, indicating that 82% of functions have been tested.

**Statement Coverage:**

This metric measures the percentage of individual code statements that have been executed by your unit tests. In your codebase:  
Statement Coverage is 76%, meaning that 76% of individual code statements have been executed during testing.

Branch Coverage: Branch coverage evaluates whether different branches within conditional statements (if, elif, else) have been tested. In your code:  
Branch Coverage is 56%, meaning that 56% of individual code Branch have been executed during testing.

**Condition Coverage:**

Condition coverage examines whether all conditions within expressions have been tested. In your code:

Some complex conditional expressions have not been fully covered by tests. It's essential to test various conditions within your code to ensure that all logical scenarios are handled correctly.

# Requirements Acceptance Testing

(You will need to fill out the column on the left with the requirements listed in software design documents and the columns on the right with the results of your own testing)

| **Software  Requirement No** | **Test** | **Implemented (Full /Partial/ None)** | **Test Results (Pass/ Fail)** | **Comments (for partial implementation or failed test results)** |
| --- | --- | --- | --- | --- |
| 1 | Users won't need to create a username and password or log in to obtain restaurant reports of inspections. | Full | Pass | The system allows public users to access restaurant inspection reports without the need for username and password. Users can perform searches and view reports seamlessly. |
| 2 | Users will be able to see information about the inspection, such as the date, the individual's name, the place of residence, the assessment score, and any infractions. | Full | Pass | The system successfully displays detailed information about restaurant inspections, including inspection date, inspector's name, location, assessment score, and any infractions found. |
| 3 | Users will be able to look up restaurants using their names, culinary preferences, geography, evaluation date, and evaluation score. | Partial | pass |  |
| 4 | Users should be able to configure filters to the system to limit search results according to their preferences. | Partial | pass |  |
| 5 | A specific page should list each restaurant assessment record and related information. | Partial | pass | The system provides specific pages for each restaurant's assessment records |
| 6 | The system will be responsive and usable on Desktop and tablets so that users may access info while they are on the go. | Full | Pass |  |
| 7 |  |  |  |  |
| 8 |  |  |  |  |
| 9 |  |  |  |  |