

Test Execution Report

TABLE OF CONTENTS

1.0 Introduction

2.0 Testing

2.1 Test cases for Functionality Testing:

2.2 Test cases for Regular Expressions

2.3 Test cases for Database Testing

2.4 Test cases for Security Testing

2.5 Test cases for Integration Testing

3.0 Tools Used for Testing

1.0 INTRODUCTION

In the ever-evolving landscape of software development, ensuring the quality and reliability of applications is paramount. Software Testing and Quality Assurance (QA) are indispensable aspects of the software development process. To demonstrate and apply these critical principles, we have embarked on a mini project aimed at creating a dynamic website dedicated to providing real-time information about the COVID-19 pandemic. This project employs a stack of web technologies, including HTML, CSS, JavaScript, PHP, and a MySQL database.

Our mini project revolves around the development of a dynamic website that serves as a comprehensive platform for disseminating COVID-19 information, enabling user interactions through account creation, commenting, and registration. By leveraging various web technologies, we have aimed to create an intuitive, informative, and user-friendly website.

2.0 TESTING STRATEGY

The testing strategy for this project is designed to ensure the thorough evaluation of the dynamic COVID-19 information website, encompassing a range of testing approaches to meet our objectives. Our testing strategy includes the following key components:

Test Levels:

We will conduct testing at multiple levels, including unit testing, integration testing, system testing, and user acceptance testing (UAT). This ensures that every component, from individual code segments to the complete system, is rigorously evaluated.

Testing Types:

Functional testing, security testing, user experience testing, and regular expression testing will be the primary testing types employed. These types of testing will address critical aspects of the website's functionality, security, user-friendliness, and data validation.

Test Environment:

We will maintain a dedicated test environment that closely replicates the production environment to simulate real-world conditions accurately. This ensures that testing results are applicable to the live deployment of the website.

Testing Tools:

Various testing tools, such as automated testing frameworks, security scanning tools, performance testing tools, and user feedback collection mechanisms, will be used to facilitate efficient and thorough testing.

Regular Expression Testing:

An integral part of our strategy involves regular expression testing. Regular expressions will be applied to validate user input, including email addresses, passwords, and other user-generated data. These tests will ensure data integrity and security.

Major Activities, Techniques, and Tools:

Test Case Design:

- **Activity:** Develop detailed test cases for each functional requirement.
- **Technique:** Equivalence partitioning, boundary value analysis.
- **Tools:** Test case management tools for documentation.

Test Automation:

- **Activity:** Automate repetitive and regression tests.
- **Technique:** Scripted testing using Selenium for web-based functionalities.
- **Tools:** Selenium WebDriver

Integration Testing:

- **Activity:** Integrating various modules together and testing the working.
- **Technique:** Top-Down Testing, Bottom-Up Testing.
- **Tools:** Pytest, JUnit

Compatibility Testing:

- **Activity:** Validate cross-browser and device compatibility.
- **Technique:** Manual testing, responsive design checks.
- **Tools:** Selenium WebDriver

Performance Monitoring:

- **Activity:** Assess system behavior under varying loads.
- **Technique:** Real-time monitoring of system metrics.
- **Tools:** Selenium WebDriver

Unit Testing:

- **Activity:** Assess each module of the system separately.
- **Technique:** Test cases, Mocking, Isolation.
- **Tools:** Selenium WebDriver, JUnit, Pytest

This testing strategy is designed to comprehensively evaluate the COVID-19 information website, ensuring functionality, security, user-friendliness, data integrity, and scalability. The incorporation of regular expression testing will be instrumental in delivering a high-quality, reliable, and user-centric website.

Approach	Type of Testing	Manual Testing	Automated Testing on Device	Tools/APIs/Libraries
Functional Testing	Feature based testing	Detailed test case execution	Automated regression testing using Selenium	Selenium WebDriver
Unit Testing	Functional Testing	Manual Testing of each unit working correctly	Automated working of each unit	Selenium WebDriver,JUnit ,Pytest
Compatibility Testing	Cross Browser Testing	Manual testing across major browsers	Automated testing on multiple browsers using	Selenium WebDriver
Integration Testing	Top Down Integration Testing	Manual testing all the components from top to down	Automated testing the components together	JUnit, Pytest
Performance Testing	Load Testing	Manual execution of load tests	Automated load testing using performance	Selenium WebDriver
	Stress Testing	Manual execution of stress tests	Automated stress testing using performance	Selenium WebDriver

2.1 Test cases for Functionality Testing:

Execution Status: Completed

MODULE/FUNCTIONALITY NAME:	Functionality Testing
CREATED BY:	Prassana Munde
DATE OF CREATION:	29/09/23
DATE OF REVIEW:	29/09/23

TEST CASE ID	TEST SCENARIO	TEST CASE	TEST STEPS	TEST DATA	EXPECTED RESULT	ACTUAL RESULT	STATUS (PASS/ FAIL)
TC_01	User Registration	Register with Valid Data	1. Access the registration page. 2. Enter valid registration details. 3. Click "Register."	1.Email address 2.Phone Number 3.Password	The user has successfully registered.	User has entered the appropriate details and has successfully registered	Pass
TC_02	Admin Login	Login with Valid Credentials	1. Access the login page. 2. Enter valid login credentials. 3. Click "Login."	1.Email address / Phone Number 2.Password	The user has successfully login.	User has entered the correct credentials and has logged in to the website	Pass
TC_03	Admin Functionality	Verifying that admin is able to add new patient data.	1.Access the Admin page. 2.Login into it. 3.Add the data.	1.Admin login data 2.Patient data	The admin is able to add the patient data.	The admin was able to successfully add covid patient data to database.	Pass
TC_04	Data Display	Verify is users are able to access Covid-19 data	1. Navigate to the COVID-19 information page.	N/A	Covid-19 data is displayed correctly on the website.	The COVID-19 website correctly displays the data	Pass
TC_05	Data Display	Verify User can create their test result separately	1.Register on Website 2.Check for test Result	Name of Patient	Test result should be displayed to user separately	Test results of all patients are Displayed in one list	Fail

2.2 Test cases for Regular Expressions:

Execution Status: Completed

MODULE/FUNCTIONALITY NAME:	Regular Expression Test Cases
CREATED BY:	Shreyas Patil
DATE OF CREATION:	29/09/23
DATE OF REVIEW:	29/09/23

TEST CASE ID	TEST SCENARIO	TEST CASE	TEST STEPS	TEST DATA	EXPECTED RESULT	ACTUAL RESULT	STATUS (PASS/ FAIL)
TC_01	Verify if Registration works correctly	Validate Email Address	Enter the email.	harshal@gmail.com	The Email entered is valid	User has successfully entered the valid email address	Pass
TC_02	Verify if Registration works correctly	Validate Phone Number	Enter the phone number	9988998899	The Phone number entered is valid	User has successfully entered the valid phone number	Pass
TC_03	Verify if Registration works correctly	Verify name length should lie in between A particular range	1.Enter valid credentials 2.Click “Register”	N	The name entered is too short and should not be accepted	The name is accepted and is registered into the database	Fail
TC_04	Date Validation	Validate Date Format	Enter a date	30/9/2023	Date format entry is valid	The entered date follows the valid Date format	Pass

2.3 Test cases for Database Testing:

Execution Status: Completed

MODULE/FUNCTIONALITY NAME:	Database Testing
CREATED BY:	Pratik Pawar
DATE OF CREATION:	30/09/23
DATE OF REVIEW:	30/09/23

TEST CASE ID	TEST SCENARIO	TEST CASE	TEST STEPS	TEST DATA	EXPECTED RESULT	ACTUAL RESULT	STATUS (PASS/FAIL)
TC_01	Verify if Database works correctly	Validate Database Connectivity	Test the database connection	N/A	The Connection is opened	The connection has been established between the website and database.	Pass
TC_02	Verify if Database works correctly	User Account Data Retrieval	Retrieve user data from Database	User Data	Data is retrieved from the database	User is able to successfully retrieve data from the database	Pass
TC_03	Verify if Database works correctly	Adding new data	Enter the name and conditions of patients	Patient Information	Data is Stored in the database	User is able to successfully add data into the database	Pass
TC_04	Verify if Database works correctly	Validate Data Deletion	Delete a user's record from the database	User's record to be deleted	Record is removed from the database.	The user data to be deleted is removed.	Pass
TC_05	Verify if Database works correctly	Validate Backup and Restore	Create a database backup	Backup file	Backup is successful	Backup file is created	Pass

2.4 Test cases for Security Testing:

Execution Status: Completed

MODULE/FUNCTIONALITY:	Security
CREATED BY:	Prassana Munde
DATE OF CREATION:	03/10/23
DATE OF REVIEW:	03/10/23

TEST CASE ID	TEST SCENARIO	TEST CASE	TEST STEPS	TEST DATA	EXPECTED RESULT	ACTUAL RESULT	STATUS (PASS/FAIL)
TC_01	Verify the security of the Covid-19 Website	SQL Injection Prevention	Inject SQL query into input	Malicious SQL	SQL injection Query is blocked	The Query acting as an SQL Injection is blocked.	Pass
TC_02	Verify the security of the Covid-19 Website	XSS (Cross-Site Scripting)	Inject JavaScript code	Malicious code	The Code is sanitized of scripts.	The Code is not sanitized and is a victim to malicious scripts	Fail
TC_03	Verify the security of the Covid-19 Website	CSRF (Cross-Site Request Forgery)	Perform a CSRF attack	N/A	Attack on the website is prevented	Attack is not prevented and may lead to user impersonation	Fail
TC_04	Verify the security of the Covid-19 Website	Error Handling	Trigger an error with incorrect input	Invalid input	User receives a user-friendly error message	User receives a user-friendly error message	Pass

2.5 Test cases for Integration Testing:

Execution Status: Completed

MODULE/FUNCTIONALITY NAME:	Integration testing
CREATED BY:	Harshal Patil
DATE OF CREATION:	03/10/23
DATE OF REVIEW:	03/10/23

TEST CASE ID	TEST SCENARIO	TEST CASE	TEST STEPS	TEST DATA	EXPECTED RESULT	ACTUAL RESULT	STATUS (PASS/FAIL)
TC_01	User Registration & Database	User Registration Data Flow	A user successfully registers on the website	User registration details	User data is stored in the database	User data is successfully stored in the database	Pass
TC_02	User Account Management & Database	Verify if User Profile is Updatable	A user updates their profile information	Updated profile data	User's profile is updated in the database	User profile was updated and stored in the database	Pass
TC_03	User Registration & User Login	Test User Registration Flow	1. User registers. 2. User Tries to login	User registration details	User is registered and logged in	User is registered and is able to successfully login.	Pass

3.0 TOOLS:

- Selenium-0.4.13
 - Type: Open-source
 - Purpose: Web application testing (Java 8)
 - Features Cross-browser testing, supports multiple programming languages (Java, Python, C#, etc.), robust community support.
- Eclipse IDE 3
- Testing extension in Eclipse
- Covid information site on localhost / website