**CURA Healthcare Service**

**Book an Appointment**

**Test Plan: Automation Testing using Java, Selenium, and TestNG**

1. **Introduction**

**•**Purpose of the test plan: To automate the testing of a web application using Java, Selenium, and TestNG.

•Scope of the automation testing: The test plan covers functional testing of the web application across multiple browsers.

1. **Test Objective**

Objective 1: Validate the login functionality.

Objective 2: Test the registration process.

Objective 3: Verify the functionality of the search feature.

1. **Test Environment**

* Software requirements:

Java Development Kit (JDK) 8 or higher.

Selenium WebDriver.

TestNG testing framework.

Apache POI library for reading data from Excel.

* Hardware requirements:

Minimum 2GB RAM.

Dual-core processor.

* Browsers and versions to be tested:

Google Chrome (latest version).

Mozilla Firefox (latest version).

Microsoft Edge (latest version).

**4. Test Framework and Tools**

* Test Automation Framework: Selenium WebDriver.
* Programming Language: Java.
* Libraries and Dependencies:

TestNG (version 7.0.0 or higher).

Apache POI (version 5.0.0 or higher) for reading data from Excel.

1. **Test Scenarios**

* Scenario 1: Verify successful login with valid credentials.
* Scenario 2: Validate error message for incorrect login credentials.
* Scenario 3: Test successful user registration.
* Scenario 4: Verify search functionality with different search queries.

1. **Test Data**

* Test data will be provided in an Excel sheet (testdata.xlsx).
* The sheet "LoginTestData" contains columns for username and password.
* The sheet "RegistrationTestData" contains columns for name, email, and password.
* The sheet "SearchTestData" contains columns for search query and expected results.

1. **Test Execution Flow**

* Set up the test environment by installing the required software and dependencies.
* Read test data from the Excel sheet using Apache POI library.
* Initialize the WebDriver instance and open the application URL.
* Execute test cases in the defined sequence.

1. **Test Suite**

* Create a TestNG XML suite file (testng.xml) to define test suites and configurations.
* Specify the test classes or packages to include in the suite.

1. **Test Cases**

* Implement test cases using Selenium WebDriver and TestNG annotations.
* Read test data from the Excel sheet and use it in the test cases.
* Include assertions and verifications for each test case.

1. **Test Reporting**

- Generate Extend Reports for comprehensive test reporting.

- Include information such as test case status, execution time, and failure details.

**11. Test Execution**

- Execute the automated tests using a build tool like Maven or Gradle.

- Set up video recording during the test execution process to capture the test runs.

**12. Test Maintenance** - Maintain the test scripts, including the Excel sheets used for test data. - Use a version control system (e.g., Git) to manage the test scripts and related files.

**13. Test Schedule** - Define the timeline for executing the automation tests. - Allocate time for test creation, execution, and maintenance.

**14. Risks and Mitigation** - Risk: Excel sheet format compatibility issues. - Mitigation: Validate and handle different Excel sheet formats