Selenium Java Project Instructions for StylemateTesting Website

Objective- E2e Testing

To automate the testing of the Stylemate using Selenium with the Cucumber framework. Incorporate assertions and reporting using TestNG, JUnit, and Extent Reports. Document the testing process with test cases, scenarios, a mind map, bug reports, summary reports, and a comprehensive test plan.

Project Scope

1. **Framework and Tool Integration**
   - Selenium for browser automation.
   - Cucumber for behavior-driven development (BDD) and Page Object Model (POM)
   - TestNG OR JUnit for test execution and management.
   - Extent Reports for advanced reporting.
2. **Key Functional Areas to Test**
   - Navigation and page consistency across the website.
   - Validation of UI components like buttons, images, and forms.
   - Testing dynamic elements such as dropdowns, popups, and modals.
   - Ensuring cross-browser and responsive compatibility.
   - Cover core Functionalities.
3. **Testing Goals**
   - Validate the functional and non-functional requirements.
   - Identify and report bugs effectively.
   - Generate detailed test reports for stakeholders.

Project Setup and Workflow

1. Framework Setup

- **Environment Preparation:**
  - Install Java, Selenium WebDriver, Maven/Gradle, and a preferred IDE (IntelliJ IDEA/STS/Eclipse).
  - Add dependencies

2. Documentation Tasks

- **Test Plan:**

- **Test Cases and Scenarios:**.
- **Mind Map:**
  - Visualize application testing areas using tools like XMind or Lucidchart.
- **Bug Reports:**
  - Use a standard template to log and categorize issues.
- **Summary Report:**
  - Summarize the execution status, highlighting issues, resolutions, and overall test metrics.

3. Implementation Tasks

- **Feature Files:**
- **Step Definitions:**
- **Assertions:**
- **Extent Reports:**
  - Configure Extent Reports for capturing logs, screenshots, and execution metrics.

4. Execution

- Execute tests on multiple browsers for cross-browser validation.
- Utilize TestNG and JUnit to manage and parameterize test runs.

Documentation Deliverables

1. Test Plan

2. Test Cases and Scenarios

3. Mind Map

4. Bug Reports

5. Summary Report

Execution and Reporting

1. **Execution with TestNG and JUnit:**
   - Run tests in parallel using TestNG suites.
   - Validate tests using JUnit assertions where applicable.
2. **Extent Report Generation:**
   - Capture logs, screenshots, and execution details for each test.
   - Customize reports with descriptive summaries and categorization.

Evaluation Metrics

1. Coverage of test cases and scenarios.
2. Proper implementation of the Selenium Cucumber framework.
3. Quality and completeness of documentation.
4. Effective reporting with Extent Reports and assertions.
5. Successful cross-browser and responsive testing.