**Test Strategy Document**

****

**Flow Chart:**

**Test Strategy**

Purpose

Scope of Testing

Types of Testing

Tools Used

Test Environment

Roles and Responsibilities

Risks

Entry Criteria

Exit Criteria

Test Deliverables

**Approach:**

**1.Purpose**

The purpose of this document is to define the overall testing strategy for the Demo Web Shop website. It outlines the approach, scope, and types of testing that will be performed to ensure the quality, reliability, and performance of the system under test.

**2. Scope of Testing**

**In Scope:**

* User Registration and Login functionality
* Product listing and search features
* Product detail pages
* Add to cart and update quantity
* Checkout process (billing, payment method, order confirmation)
* Navigation, buttons, links, and basic UI validations
* Contact us and newsletter subscription features

**Out of Scope:**

* Admin functionalities (not accessible from the front end)
* Backend database validations
* Integration with live payment gateways (assumed to be mocked)

**3. Types of Testing to be Performed**

* **Functional Testing:**  
  To ensure that each function of the application behaves as expected.
* **Smoke Testing:**  
  To verify critical functionalities are working after each build.
* **Regression Testing:**  
  To validate that recent code changes have not affected existing functionality.
* **UI/UX Testing:**  
  To check layout, images, alignment, responsiveness, and visual consistency.
* **Cross-browser Testing:**  
  To ensure compatibility across Chrome, Firefox, and Edge.
* **Negative Testing:**  
  To validate system behavior for invalid inputs (e.g., empty fields, wrong credentials).
* **Security Testing (Basic):**  
  To check for input validations and session timeouts (no in-depth penetration testing).

**4.Tools & Frameworks**

The testing team uses the following tools to manage testing, find bugs, and complete test cycles smoothly:

* **Selenium WebDriver** – Used to automate UI testing across different browsers and platforms
* **Postman / RestAssured** – Used to test and validate APIs
* **JIRA** – Used for bug tracking, sprint planning, and team collaboration
* **TestRail / Zephyr** – Used to write, manage, and track test cases and test cycles

**5.Testing Environments**

The testing process will use different environments for various testing needs:

* **Development Environment**  
  Used by developers for unit testing and early integration  
  – Includes mock services and test data
* **QA / Staging Environment**  
  Mirrors production setup and used for functional, regression, and exploratory testing  
  – Latest stable build after each sprint  
  – Realistic test data sets  
  – Payment gateway sandbox configured
* **UAT (User Acceptance Testing) Environment**  
  Used by business users and product owners to validate user stories  
  – Final sprint deliverables  
  – Role-based access for different users  
  – Logs and monitoring tools for collecting feedback
* **Performance Testing Environment**  
  Used to simulate high user load and test system under stress  
  – Scaled setup to mimic production load  
  – Load testing tools like **JMeter** or **Gatling**  
  – Dashboards for collecting performance metrics

**6.Roles & Responsibilities**

In the testing project, each team member has specific tasks to ensure smooth execution and product quality:

* **Test Lead** – Plans and manages the full testing process
* **QA Engineer** – Writes and runs test cases
* **Developer** – Fixes bugs and helps with testing support
* **Business Analyst** – Explains and clarifies the requirements
* **Project Manager** – Tracks overall progress and updates stakeholders

**7.Risks**

Possible problems we may face during testing:

* **Limited time** – May not be able to test everything in short sprints
* **Missing bugs** – Some bugs may stay hidden if all combinations are not tested
* **Website changes often** – In Agile, features may change during sprints, and old test cases might not work

**How to handle:**

* Update test cases quickly after any changes
* Focus on the most important features first
* Do group testing to cover more scenarios

**8. Entry Criteria**

Things that must be ready before starting testing:

* Website is working and accessible
* The feature (like login or add to cart) is developed
* Test data is ready (username, password, product name, etc.)
* Browser and internet access are available
* User stories are clearly written

**9. Exit Criteria**

Things that must be completed to finish testing:

* All planned test cases are executed
* Major and high-priority bugs are fixed
* Main features (like login or checkout) are working properly
* Team confirms the feature is ready for release

**10. Test Deliverables**

Files we will prepare and share after testing:

* **Test Case File**  
  – List of all test cases (Excel or Google Sheet)  
  – Example: Steps to test login, add to cart, checkout, etc.
* **Bug Report**  
  – List of all bugs found during testing  
  – Includes where the bug occurred, what the bug is, and how to check it
* **Test Summary Report**  
  – Number of test cases passed/failed  
  – Number of bugs found  
  – Final result of testing.