**Test Plan for DemoBlaze.com**

**1. Introduction**

**1.1 Purpose**

**The purpose of automation testing** is to **boost speed, reliability, and test coverage** while reducing **manual effort, time, and cost**—making software delivery faster and more reliable.

**1.2 Objectives**

**1. Increase Test Efficiency & Coverage**

* Automate **repetitive** test cases like login, checkout, etc.
* Run tests across **multiple browsers** or configurations quickly.
* Execute **large test suites** without manual effort.

**2. Ensure Faster Feedback**

* Provide **instant feedback** on code changes (especially in Agile/CI-CD environments).
* Help developers and QA catch bugs early in the pipeline.

**3. Enable Regression Testing**

* Quickly test existing features after new updates or bug fixes.
* Ensure that **new code doesn’t break old functionality**.

**4. Improve Accuracy**

* Eliminate **human error** in repetitive manual testing.
* Ensure consistent steps and data usage across tests.

**2. Scope of Testing**

**In-Scope:**

* **Functional Testing**:
* User Sign-Up and Login
* Product Browsing and Search
* Cart Management (Add/Remove Items)
* Checkout Process (Order Confirmation)
* **Non-Functional Testing**:
* UI/UX Responsiveness
* Cross-Browser Compatibility (Chrome, Firefox, Safari)
* Performance Testing (Page Load Times)

**Out of Scope:**

* Backend API Testing
* Payment Gateway Integration (as it's a mock site)
* Mobile Application Testing

**3. Testing Approach**

***Phase 1: Requirement Analysis***

* Understand test scenarios from DemoBlaze (login, cart, purchase).
* Identify automation candidates (e.g., smoke and regression tests).

***Phase 2: Test Planning***

* Define scope, tools (Selenium/TestNG/Cypress), environments, and timelines.
* Prioritize test cases (high ROI for automation).

***Phase 3: Framework Design***

* Choose framework type: Page Object Model (POM) + Data-Driven.
* Integrate test reporting (ExtentReports, Allure).
* Version control with Git.

***Phase 4: Test Script Development***

* Write scripts using chosen tools (e.g., Java + Selenium + TestNG).
* Use Page Object Model to keep locators and logic separate.
* Parameterize using Excel, JSON, or DataProviders.

***Phase 5: Test Execution***

* Run scripts locally and on CI/CD pipelines (e.g., Jenkins).
* Perform cross-browser runs with Selenium Grid or cloud tools.

***Phase 6: Reporting & Defect Logging***

* Generate HTML or Allure reports.
* Capture screenshots on failure.
* Log issues in Jira or similar tools

**4. Test Deliverables**

* **Test Plan Document** (this document).
* **Test Cases** covering all functionalities in scope.
* **Test Execution Reports** summarizing test results.

**RTM (Requirements Traceability Matrix)** linking test cases to requirements.

**7. Test Environment & Tools**

**7.1** Test Environment

1. Application Under Test: [https://www.demoblaze.com](https://www.demoblaze.com/)

2. Operating System: Windows 10/11

3. Browsers: Firefox

2. Automation Tools Stack

1.java Eclipse

2.SeleniumWebDriver

3.TestNg

**8. Roles & Responsibilities**

**8.1 Responsibilities**

| **Role** | **Responsibility** |
| --- | --- |
| **QA Team** | Plan execution,Design and Execute test cases, Report defects. |

**8.2 QA Team Names**

|  |  |
| --- | --- |
| **QA Team** | Bashir Rimawi  Dina Abuqaree  Mark Qarraa |

**9. Entry & Exit Criteria**

**9.1 Entry Criteria**

✔ Test environment is set up.   
✔ Functional development is complete.   
✔ Test cases are reviewed and approved.   
✔ Test data is available.

**9.2 Exit Criteria**

✔ All major test cases have been executed.   
✔ No critical or high-severity defects remain open.

**10. Defect Management**

* **Daily defect triage meetings** to discuss and resolve high-priority issues.
* **Bug life cycle tracking** from detection to resolution.

**11. Test Execution Plan**

| **Phase** | **Start Date** | **End Date** |
| --- | --- | --- |
| **Test Planning** | 17/4/2025 | 19/4/2025 |
| **Test Case Design** | 19/4/2025 | 20/4/2025 |
| **Test Execution** | 22/4/2025 | 24/4/2025 |
| **Defect Fixing & Retesting** | --- | --- |