**Test Plan (demoblaze.com)**

**TABLE OF CONTENT**

|  |  |  |
| --- | --- | --- |
| **SR NO.** | **TITLE** | **PAGE NO.** |
| 1 | [Objective](#_tyoevl8rsr) | 1 |
| 2 | [Scope](#_iiu9rdxvbojv) | 1 |
| 3 | [Inclusions](#_v0jbkqx2u9xu) | 1 |
| 4 | [Test Environments](#_kfhs7irweitd) | 1-2 |
| 5 | [Defect Reporting Procedure](#_hmb6zyi9404j) | 2 |
| 6 | [Test Strategy](#_aifn8ev7zqa1) | 2 |
| 7 | [Test Schedule](#_ez6hwie9oqia) | 3 |
| 8 | [Test Deliverables.](#_hnq2u2gp5ms) | 3 |
| 9 | [Entry and Exit Criteria](#_jj90dhdnw89n)   * [Entry Criteria:](#_oprnh6cxof3h) * [Exit Criteria:](#_1es4cb8zit02) | 3 |
| 10 | [Test Execution](#_6secjh67li5z) | 3 |
| 11 | [Test Closure](#_yyc7mksy1f2n)   * [Entry Criteria:](#_42yr3zly18wc) * [Exit Criteria:](#_bfi30sl6em27) | 3-4 |
| 12 | [Tools](#_rv4zw92wbnho) | 4 |
| 13 | [Risks and Mitigations](#_gi5cq358f37y) | 4 |
| 14 | [Approvals](#_3zamqx573fn1) | 4 |

**Objective**

The objective of this test plan is to ensure that the Demoblaze e-commerce platform meets all functional requirements, provides a seamless user experience, is secure, and performs efficiently under various load conditions.

* Frontend Technologies: HTML, CSS, JavaScript (React-based UI)
* Backend: APIs handling product catalog, user management, orders
* Database: Relational or NoSQL database for product and user data
* Web Server: Apache/Nginx or equivalent

**Scope**

Testing will cover the following critical areas of the Demoblaze website:

* User registration and login functionality
* Product browsing, search, and filtering
* Add to cart, update cart, and checkout process
* Payment gateway integration and order placement
* Order management and history
* Performance and load handling capabilities
* Security validations for user data and transactions

**Inclusions**

Included deliverables for this test plan:

* Test strategy document
* Test scenarios and detailed test cases
* Test execution reports
* Defect/bug reports
* Performance and load test reports

**Test Environments**

Testing will be conducted in multiple environments:

* Development environment
* Testing/Staging environment
* Production environment (limited testing)

Supported OS and browsers:

* Windows 10/11 (Chrome, Firefox, Edge)
* macOS (Safari, Chrome)
* Android Mobile (Chrome browser)
* iOS Mobile (Safari browser)

Device coverage includes desktops, laptops, tablets, and smartphones over various network conditions (Wi-Fi, 4G/5G).

**Defect Reporting Procedure**

* Defect identification based on deviation from expected functionality, UI issues, security flaws, or performance bottlenecks.
* Defects will be reported using JIRA with steps to reproduce, severity, priority, logs, and screenshots.
* Defects will be triaged and assigned to respective development teams.
* Status updates will be shared daily with stakeholders.

**Test Strategy:**

|  |  |
| --- | --- |
| **Component** | **Description** |
| Objectives | Validate all critical functionalities and ensure system stability and security |
| Test Levels | Unit testing by devs, integration testing, system testing, UAT |
| Test Types | Functional, Security, Usability, Regression, Performance, Load Testing |
| Test Techniques | Black-box, exploratory testing, boundary value analysis, decision table, use case testing |
| Test Deliverables | Test plans, test cases/scripts, defect reports, performance and usability reports |
| Test Environment | Configured across multiple OS, browsers, devices, and network conditions |
| Schedule | Testing spread across sprints with defined milestones |
| Resource Allocation | QA team divided into functional, performance, and automation testers |
| Risk Management | Address risks like limited environment access, build stability, and time constraints |
| Test Exit Criteria | Completion of planned test coverage with acceptable defect severity levels, performance benchmarks |

**Test Schedule**

|  |  |
| --- | --- |
| **Task** | **Dates** |
| Test planning | To be decided |
| Test case creation | To be decided |
| Test execution | To be decided |
| Report submission | To be decided |

Testing is planned across 2-3 sprints depending on build readiness.

**Test Deliverables**

* Approved Test Plan document
* Approved Test Scenarios and Test Cases
* Daily/Weekly Test Execution Reports
* Defect/Bug Reports
* Final Test Summary Report

**Entry and Exit Criteria**

* Entry Criteria: Requirements reviewed, test environment set up, test data prepared, test cases approved
* Exit Criteria: All critical test cases executed with passed results or defects logged, performance criteria met, test reports ready

**Test Execution**

* Testers to execute test cases in test cycle
* Regression testing post defect fixes
* Exploratory testing alongside scripted tests

**Test Closure**

* **Entry Criteria:**
  + Completion of all planned test cycles and execution of test cases.
  + No critical or high-priority defects remain unresolved; all defects are either fixed or accepted as deferred.
  + Test completion reports, including defect and test execution reports, are prepared and available.
* **Exit Criteria:**
  + Test summary report and lessons learned documents are reviewed and approved by stakeholders.
  + All test artifacts and deliverables are submitted and archived.
  + Formal sign-off is obtained from the project stakeholders to officially close the testing phase.

**Tools**

* EXCEL sheet to record Testcases
* JIRA for defect tracking
* Performance testing tools like GTmetrix
* Postman for API testing
* Load testing tools like BlazeMeter

**Risks and Mitigations**

* Risk: Delays in environment availability — Mitigation: Early setup and backup environments
* Risk: Incomplete requirements — Mitigation: Continuous communication with stakeholders
* Risk: Build instability — Mitigation: Smoke testing and delayed regression cycles
* Risk: Limited testing time — Mitigation: Prioritize critical test cases and resource ramp-up

**Approvals**

* Test Plan approved by QA Lead and Project Manager
* Test Cases approved by Business Analysts and Client (where applicable)
* Reports reviewed and signed off by stakeholders