**Demoblaze E-Commerce Test Plan**

# 1. Introduction

This document defines the **test plan** for the DemoBlaze web application (https://www.demoblaze.com). DemoBlaze is a demo e-commerce site that allows users to browse products sign up, log in, add products to the cart, and place orders.

Due to the limited time, testing scope is limited only for macOS platform and Google Chrome browsers.

# 2. Scope

### In-Scope

* Functional testing of UI.
* Google Chrome
* Positive and negative test cases for login, cart, and purchase flow.
* Several E2E automated test cases.

### Out-of-Scope

* API testing
* Performance testing.
* Security penetration testing.
* Other types of non-functional tests
* Cross-browser compatibility (Firefox, Edge, etc.).

### 3. (Supposed) Roles and Responsibilities

* **QA Lead**: Prepare and approve test plan, oversee execution.
* **Test Engineer**: Design and execute test cases, log defects.
* **Automation Engineer**: Develop and maintain automated test scripts.
* **Developer**: Fix reported defects.

# 4. Testing Approach

* **Testing Levels:** E2E testing.
* **Testing Types:** Functional, UI.
* **Automation:** Selenium (UI), Cucumber, Junit,
* **Manual Testing:** Exploratory and functional edge cases.

**UI – E2E tests**

* + Responsibility: QA Team
  + Tools: Selenium WebDriver, Cucumber BDD
  + Expected number of test cases: 5

## 5. Testing Across Environments

|  |  |  |  |
| --- | --- | --- | --- |
| **Environment** | **Purpose** | **Tests to Execute** | **Notes** |
| **\*DEV** | Active development, unstable builds | - Unit testing (by devs)  - Smoke testing (basic sanity)  - Early functional checks | Often local or dev server; frequent deployments; not for formal QA sign-off |
| **\*Test / QA** | Full QA testing | - Full functional testing  - Regression testing  - Negative testing  - UI / usability testing  - API testing  - End-to-end testing | Stable build deployed for QA; issues logged and fixed before UAT |
| **\*UAT** | Client / stakeholder validation | - Critical business flows  - Exploratory testing for approval | Mimics production closely; final sign-off environment |
| **PROD** | Live environment | - Smoke monitoring  - Sanity checks after deployment  - Post-deployment validation | Testing here is minimal and controlled; usually automated monitoring or real-time checks only |

*\*testing is limited only to PROD environment.*

## 6. Definition of Ready (DoR)

- User story is clearly defined and understood by the team.  
- Acceptance criteria are documented.  
- Dependencies are identified and resolved.  
- Test data and environment requirements are available.  
- Story is estimated and prioritized in the backlog.

## 7. Definition of Done (DoD)

- Code has been reviewed and merged into the main branch.  
- Unit tests have been written and passed.  
- Functional and regression tests are executed successfully.  
- No critical or high-severity defects remain open.  
- Documentation and release notes are updated.  
- Feature has been deployed to staging and approved by QA.

## 8. Entry Criteria

* Requirements are signed off.
* Test environment is ready.
* Test data is prepared.
* Application build deployed and stable.

## 9. Exit Criteria:

* 100% of test cases executed.
* All critical defects fixed and retested.
* No open Critical / High severity bugs.

## 10. Suspension Criteria

Testing will be **suspended** if:

* Application build is unstable (frequent crashes, blocking issues).
* Any of test cases are blocked.
* Critical environment issues (e.g., server downtime, API unavailability).

## 11. Resumption Criteria

Testing will **resume** once:

* Blocking defects are resolved and verified.
* Test environment is stable.
* A new stable build is deployed.

## 12. Risks and Mitigation

Not assessed for this task.

## 13. Defect Management / Reporting

* All defects will be logged in **Github**.
* Severity & priority levels will be used to triage issues.
* Test Cases (Manual + Automation)
* Defect Reports
* Test Summary Report
* Automation Test Scripts

## 14. Release Process

Not assessed for this task.

# 15. Deliverables

* All defects will be logged in the Github.
* Severity & priority levels will be used to triage issues.
* Test Plan
* RTM (Requirement Traceability Matrix)