OPEN CART – TEST PLAN

**Product Name: OpenCart (Frontend)**

**Website:** [**https://demo.opencart.com/**](https://demo.opencart.com/)

**Overview**:

This document outlines the test plan for the manual testing of the OpenCart frontend on the website <https://demo.opencart.com/>. The plan includes the scope of testing, test environments, test strategy, defect reporting procedure, roles/responsibilities, test schedule, test deliverables, pricing, entry/exit criteria, suspension/resumption criteria, tools, risks, and approvals.

**Scope:**

The scope of this testing project covers the following features of the OpenCart frontend on the website <https://demo.opencart.com/>:

* Register
* Login & Logout
* Forgot Password
* Search
* Product Compare
* Product Display Page
* Add to Cart
* Wish List
* Shopping Cart
* Currencies
* Home Page
* Checkout Page
* My Account Page
* Order History Page
* Downloads Page
* Contact Us Page
* Menu Options
* Footer Options
* Category Pages

**Test Environments**:

The testing will be conducted on the following platforms and browsers:

* Windows 10/11: Chrome, Firefox, and Edge
* Mac OS: Safari
* Android Mobile OS: Chrome
* iPhone Mobile OS: Safari

**Exclusions**:

The following items are excluded from this testing project:

* Features not mentioned in the "Scope" section
* Third-party features or Payment gateways
* Test Automation

**Test Strategy**:

The testing approach for this project includes the following steps:

1. Creation of Test Scenarios and Test Cases for the identified features, utilizing various Test Designing techniques.
2. Testing process, including Smoke Testing, in-depth testing, and multiple resources testing simultaneously on supported environments.
3. Reporting defects in a bug tracking tool and providing daily status updates via email.
4. Conducting Smoke Testing, Regression Testing, Retesting, Usability Testing, Functionality Testing, and UI Testing.
5. Applying best practices such as Context Driven Testing, Shift Left Testing, Exploratory Testing, and End-to-End Flow Testing.

Defect Reporting Procedure: During test execution, any deviation from expected behavior, usability issues, or defects encountered will be documented. Defects will be retested for reproducibility, and screenshots with reproduction steps will be captured. Daily defect reports will be sent along with observations in an email.

**Roles/Responsibilities:**

* Test Manager: Escalations
* Test Lead: Test plan creation, test case execution, defect reporting, coordination, and daily updates.
* Senior Test Engineer: Test case execution and defect reporting.
* Test Engineer: Test case execution and defect reporting.

**Test Schedule**: The test schedule for this project is as follows:

* Creating Test Plan: Start Date to End Date
* Test Case Creation: Start Date to End Date
* Test Case Execution: Start Date to End Date
* Summary Reports Submission: Date

**Test Deliverables**: The following deliverables will be provided to the client:

* Test Plan: Details on the scope, strategy, schedule, resource requirements, and deliverables.
* Functional Test Cases: Test cases created for the identified scope.
* Defect Reports: Detailed descriptions of identified defects, including screenshots and steps to reproduce.
* Summary Reports: Summary of bugs by Bug#, Functional Area, and Priority.

**Pricing**: Pricing information is not applicable to this testing project.

**Entry and Exit Criteria**: Entry and exit criteria are defined for each phase of the Software Testing Life Cycle (STLC). These include requirement analysis, test planning, test designing, test execution, and test closure.

**Suspension and Resumption Criteria**: The project may be suspended or resumed based on client decisions. Resource allocation will be adjusted accordingly.

**Tools**: The following tools will be utilized for this project:

* XYZ Bug Tracking Tool
* Mind map Tool
* Snipping Screenshot Tool
* Word and Excel documents

**Risks and Mitigations**: The following risks have been identified, along with their mitigations:

* Risk: Non-Availability of a Resource Mitigation: Backup Resource Planning
* Risk: Build URL is not working Mitigation: Resources will work on other tasks
* Risk: Less time for Testing Mitigation: Dynamic resource allocation based on client needs

**Approvals**: Various documents, including the Test Plan, Test Scenarios, Test Cases, and Reports, will be sent for client approval before proceeding with the testing activities. Testing will only continue upon receiving these approvals.