



OpenCart

# TEST PLAN

Product Name: OpenCart(Frontend)

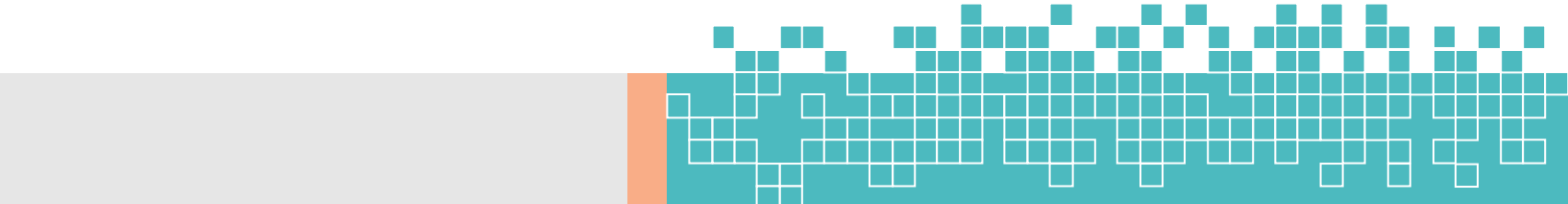
Prepared by: Josina Jose

Date May 10,2024





# Table of Contents

1. Introduction
  2. Purpose
  3. Feature Overview
  4. Feature Owners
  5. Test Plan
  6. Test Environment
  7. Test Strategy
  8. Test Deliverable
  9. Defect Management
  10. Roles/Responsibilities
  11. Test Schedule
  12. Entry and Exit criteria
  13. Risk and Mitigation
  14. Test Dependency
- 

## 1. Introduction

ABC Software Solution has been engaged by OpenCart to conduct comprehensive testing on the functionality of the OpenCart demo ecommerce application. This high-level test plan outlines the scope, test strategy, schedule, resource requirements, and deliverables to ensure the application meets functional, performance, and security standards, with a focus on validating core eCommerce features, identifying defect and ensuring a seamless user experience.

## 2. Purpose

The purpose of this test plan is to outline the testing activities for the current and upcoming sprints of the OpenCart Demo eCommerce application. It defines the testing scope, strategy, and schedule to ensure that all functionalities are thoroughly tested within the sprint timelines. This plan helps the QA team efficiently track progress, identify defects, and ensure a high-quality release.

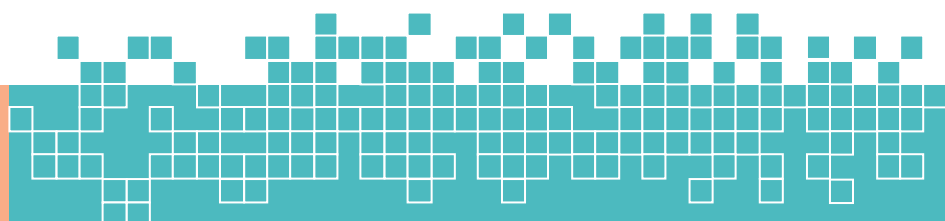
## 3. Feature Overview

The OpenCart Demo eCommerce application provides a platform for online shopping with essential eCommerce functionalities. The key features include:

- User Management – Customer registration, login, and account management.
- Product Catalog – Browsing, searching, and filtering products.
- Shopping Cart & Checkout – Adding/removing products, applying discounts, and completing orders.
- Payment Integration – Supporting multiple payment methods.
- Order Management – Viewing order history, tracking orders, and managing returns

## 4. Feature Owners

The table below outlines the key feature owners and their responsibilities.



Feature	Owner (Role/Team)	Responsibility
Shopping Cart & Checkout	QA Lead / Dev Team	Validate cart functionality, order placement, and payment processing.

## 5. Test Plan

### Scope of Testing

#### ✓ In-Scope Features (Covered in Testing)

The project scope encompasses testing the following features of the 'https://demo.opencart.com/' web application:

- User Authentication: User Registration, Login, Logout, Forgot password
- Product Exploration: Product Search, product Compare, Product Display page
- Shopping and Wishlist: Add to cart, Wishlist, Shopping Cart
- E-Commerce Transactions: Currencies, Checkout Page
- User Account management: My Account page, Order History page, Downloads Page
- Site Navigation: Home Page, Count Us Page, Menu Options, Footer Options, Category Pages

#### ✗ Out-of-Scope Features (Excluded from Testing)

- Third-Party Payment Gateways (e.g., PayPal, Stripe API integrations)
- Performance Testing Under High Load (Beyond standard user traffic)
- Database Migration & Backup Testing
- End-to-End Mobile App Testing (Only mobile web testing is included)

## 6. Test Environment

The testing of the OpenCart Demo eCommerce application will be conducted across the following environments:

- Windows 11 – Chrome and Edge browsers
- Mac OS – Safari browser
- Android Mobile OS – Chrome browser
- iPhone Mobile OS – Safari browser

## 7. Test Strategy

### 1. Test Scenario and Test Case Creation:

- Utilizing various techniques including Equivalence Class Partition, Boundary Value Analysis, Decision Table Testing, State Transition Testing, and Use Case Testing.
- Incorporating expertise-based methods such as Error Guessing and Exploratory Testing.
- Prioritizing Test Cases.

### 2. Testing Process:

- Conducting Smoke Testing initially.
- Rejecting builds failing Smoke Testing and waiting for a stable build.
- Performing in-depth testing on stable builds.
- Simultaneously testing on multiple supported environments.
- Reporting bugs daily to the development team.

### 3. Types of Testing:

- Smoke Testing and Sanity Testing
- Regression Testing and Retesting
- Usability Testing, Functionality & UI Testing

### 4. Best Practices:

- Context Driven Testing
- Shift Left Testing
- Exploratory Testing
- End to End Flow Testing

## 8. Test Deliverable

The following test deliverables will be provided at the end of the testing cycle:

Deliverable	Description	Target Completion Date
Test Plan Document	Detailed plan outlining scope, strategy, and schedule.	Aug 10, 2024
Test Cases & Test Scripts	Comprehensive test cases and automated test scripts for functionality validation.	Date
Test Execution Report	Summary of executed tests, pass/fail rates, and any deviations.	
Defect Reports	Logs of identified defects, including details and status.	
Test Summary Report	Final report summarizing testing results, coverage, and product readiness.	

## 9. Defect Management

During test execution, any deviation or usability issue will be noted, documented, and reported daily, including screenshots and steps for reproducibility. Defects will be recorded in an Excel sheet, and test scenarios/cases will be documented in an Excel document.

## 10. Roles/Responsibilities

Name	Role	Responsibility
Sam	Test manager	Escalation
	Test Lead	<ul style="list-style-type: none"><li>Create the Test Plan and get the client signoffs</li><li>Interact with the application, create and execute the test cases</li><li>Report defects</li><li>Coordinate the test execution. Verify validity of the defects being reported.</li><li>Submit daily issue updates and summary defect reports to the client.</li><li>Attend any meeting with client.</li></ul>
	Senior Tester	<ul style="list-style-type: none"><li>Interact with the application</li><li>Create and Execute the Test cases.</li><li>Report defects</li></ul>

	Tester	<ul style="list-style-type: none"> <li>▪ Interact with the application</li> <li>▪ Execute the Test cases.</li> <li>▪ Report defects</li> </ul>
--	--------	--

## 11. Test Schedule

Task	Time Duration
Creating Test Plan Start Date to End Date	Start Date to End Date
Test Case Creation	
Test Case Execution	
Summary Reports Submission	

## 12. Entry and Exit criteria

The below are the entry and exit criteria for every phase of Software Testing Life Cycle:

### Requirement Analysis

Entry Criteria:

- Once the testing team receives the Requirements Documents or details about the Project

Exit Criteria:

- List of Requirements are explored and understood by the Testing team
- Doubts are cleared

### Test Planning

Entry Criteria:

- Testable Requirements derived from the given Requirements Documents or Project details.
- Doubts are cleared

Exit Criteria:

- Test Plan document (includes Test Strategy) is signed off by the Client

## Test Designing

Entry Criteria:

- Test Plan Document is signed off by the Client

Exit Criteria:

- Test Scenarios and Test Cases Documents are signed off by the Client

## Test Execution

Entry Criteria:

- Test Scenarios and Test Cases Documents are signed off by the Client
- Application is ready for Testing

Exit Criteria:

- Test Case Reports, Defect Reports are ready

## Test Closure

Entry Criteria:

- Test Case Reports, Defect Reports are ready

Exit Criteria:

- Test Summary Reports

## 13. Risk and Mitigation

The following are the list of risks possible and the ways to mitigate them:

- 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: Ramp up the resources based on the Client needs dynamically

## 14. Test Dependency

Test dependencies are factors or conditions that must be in place for testing to proceed smoothly. These include:

- Requirement Sign-off – Testing cannot begin until all requirements are finalized and approved.
- Development Completion – Testing can only start once the features to be tested are fully developed and stable.
- Test Environment Setup – Testing cannot proceed without a fully configured and accessible test environment.
- Availability of Test Data – Adequate and accurate test data must be available to perform test cases.
- Third-Party Integrations – Dependencies on external services or APIs (e.g., payment gateways) that must be functional for testing.
- Tools & Software – Availability of necessary tools (e.g., test management tools, automation frameworks) for conducting testing.
-