# **Test Strategy for Opencart.com**

# **Introduction**

The **Test Strategy Document** outlines the overall approach to testing for the project. It serves as a high-level guide to ensure that all testing activities are well-structured, efficient, and aligned with project requirements.

**Objectives**: The objective is to test the end-to-end functionality, usability and performance of this website and ensure it meets the business and technical requirements.

## **Scope**

### In scope:

**User Authentication** – Login, Registration, Password Reset  
 **Product Search & Filters** – Search functionality, category filters  
 **Shopping Cart** – Add, update, and remove items  
 **Checkout & Payments** – Order placement, payment gateway integration  
 **Order Management** – Order history, status updates  
 **UI & Usability Testing** – Layout, responsiveness, navigation

### Out scope:

* Backend database performance testing (handled separately),
* Third-party payment gateway security (handled by provider),
* Email/SMS notifications (covered by integration testing).
* Mobile App Testing
* External System Integrations
* Performance and Load Testing

## **Testing Levels**

* Unit Testing
* Integration Testing
* System Testing

## **Test Environments**

**OS:** Windows, macOS, Android, iOS

**Browsers:** Chrome, Firefox, Edge, Safari

**Devices:** Desktop, Tablet, Mobile

**Tools:**

* **Test Management:** JIRA + Zephyr Scale

# **Testing Techniques**

* Blackbox testing
* Regression Testing
* Functional Testing
* Non-Functional Testing
* Exploratory Testing

**Entry & Exit Criteria**

**Entry Criteria (When to Start Testing?)**

Requirements are finalized.  
 Test cases are written and reviewed.  
 Test environment is set up and stable.  
 Application build is deployed for testing.

**Exit Criteria (When to Stop Testing?)**

All critical defects are fixed.  
 95% of test cases pass successfully.  
 Regression and performance testing are completed.  
 Test reports are reviewed and approved.

## **Risk and Mitigation**

Risk: Late requirement changes

Mitigation: Regular sprint meetings with stakeholders

Risk: Payment gateway failures

Mitigation: Simulate multiple test scenarios with mock data

Risk: Performance issues

Mitigation: Conduct load testing early in development

Risk: Cross-browser compatibility issues

Mitigation: Perform parallel testing on different browsers

## **Deliverables**

Test plans

Test Scenarios and Cases

Defect Report

Final Test Summary Reports