# **OpenCart Ecommerce Test Overview**

|  |  |
| --- | --- |
| **Field** | **Value** |
| **Test Plan ID:** | TP-OPEN-CART-001 |
| **Test Plan Name:** | Opencart.com Functionality Test Plan  Test Overview |
| **Created By:** | Dalonda Ikhimokpa |
| **Creation Date:** | Mar 1, 2024 |
| **Version:** | 1.0 |

## **1. Introduction**

This test plan outlines the approach, resources, and schedule for testing the OpenCart e-commerce platform. It provides a comprehensive framework to ensure that all aspects of the application are thoroughly tested before deployment to production.

## **2. Test Objectives**

* Verify that all functional requirements of the e-commerce platform are implemented correctly
* Ensure the application provides a seamless and intuitive user experience
* Validate that the system is secure and protects sensitive customer information
* Confirm that the application performs well under various load conditions
* Verify cross-browser and cross-device compatibility

## **3. Scope**

### **3.1 In Scope**

The following features will be tested:

|  |  |  |
| --- | --- | --- |
| **Feature** | **Description** | **Priority** |
| User Authentication | Account creation, login, password recovery | High |
| Product Catalog | Browsing, searching, filtering products | High |
| Shopping Cart | Adding, removing, updating items | High |
| Checkout Process | Address entry, shipping selection, order review | High |
| Payment Processing | Integration with payment gateways, transaction security | Critical |
| Order Management | Order history, status tracking, invoice generation | Medium |
| User Account | Profile management, address book, wishlist | Medium |
| Admin Backend | Product management, order processing, customer management | High |

### **3.2 Out of Scope**

* Third-party services beyond API integration points
* Server hardware performance
* Database optimization
* Content creation and copywriting quality

## **4. Test Deliverables**

The following documents will be produced during the testing process:

* Test Strategy Document
* Test Cases and Scripts
* Test Execution Reports
* Defect Reports
* Performance Test Results
* Security Test Results
* User Acceptance Test Results
* Final Test Summary Report

## **5. Test Environments**

### **5.1 Environment Details**

|  |  |  |
| --- | --- | --- |
| **Environment Name** | **URL** | **Purpose** |
| Development | dev.opencart.com | For developers to test new features |
| QA | demo.opencart.com | Main testing environment |
| UAT | uat.opencart.com | User acceptance testing |
| Pre-Production | preprod.opencart.com | Final validation before production |
| Production | app.opencart.com | Live environment |

### **5.2 Environment Configurations**

#### **Hardware Requirements:**

* Server: 8GB RAM, 4 Core CPU, 100GB SSD
* CI/CD Pipeline: Jenkins server with test automation capabilities

#### **Software Requirements:**

* Operating Systems: Windows 10, macOS, Ubuntu Linux
* Web Browsers: Chrome (latest), Firefox (latest), Safari (latest), Edge (latest)
* Mobile Devices: Android (latest 2 versions), iOS (latest 2 versions)

#### **Network Configuration:**

* Internet connection with minimum 10 Mbps
* Firewall configuration allowing HTTP/HTTPS traffic
* SSL certificates installed and configured

## **6. Test Strategy**

Our testing approach will incorporate multiple testing techniques to ensure comprehensive coverage:

### **6.1 Testing Types**

* **Functional Testing**: Verify all features work according to requirements
* **Usability Testing**: Ensure intuitive user interface and experience
* **Compatibility Testing**: Verify functionality across browsers and devices
* **Performance Testing**: Validate system behavior under load
* **Security Testing**: Ensure protection of sensitive data
* **Regression Testing**: Verify new changes don't break existing functionality
* **Integration Testing**: Test interactions between different system components

### **6.2 Testing Techniques**

* Equivalence Class Partitioning
* Boundary Value Analysis
* Decision Table Testing
* State Transition Testing
* Use Case Testing
* Exploratory Testing
* Error Guessing

### **6.3 Testing Approach**

1. **Shift-Left Testing**: Begin testing early in the development cycle
2. **Context-Driven Testing**: Adapt testing approach based on application context
3. **Risk-Based Testing**: Prioritize test cases based on risk assessment
4. **End-to-End Flow Testing**: Test complete user journeys

### **7. Test Schedule**

|  |  |  |  |
| --- | --- | --- | --- |
| **Phase** | **Start Date** | **End Date** | **Duration** |
| Test Planning | March 1, 2025 | March 3, 2025 | 3 days |
| Test Case Design | March 4, 2025 | March 8, 2025 | 5 days |
| Environment Setup | March 6, 2025 | March 9, 2025 | 4 days (overlapping) |
| Test Execution | March 10, 2025 | March 22, 2025 | 13 days |
| Bug Reporting & Verification | March 23, 2025 | March 29, 2025 | 1 week |
| Test Closure & Reporting | March 30, 2025 | April 3, 2025 | 5 days |

## **8. Entry and Exit Criteria**

### **8.1 Entry Criteria**

#### **Requirement Analysis**

* Complete requirements documentation available
* Requirements reviewed and approved by stakeholders
* Test team briefed on requirements

#### **Test Execution**

* Test environment is ready and stable
* Test cases have been reviewed and approved
* Required test data is available
* Smoke test has passed successfully

### **8.2 Exit Criteria**

#### **Test Execution**

* All planned test cases executed
* No critical or high severity defects remain open
* All required test reports completed
* Performance metrics meet defined thresholds
* Security vulnerabilities addressed

#### **Test Closure**

* Test summary report completed and approved
* Lessons learned documented
* Test artifacts archived

## **9. Defect Management**

### **9.1 Defect Reporting Procedure**

1. Tester identifies a defect during testing
2. Defect is documented in JIRA with appropriate severity and priority
3. Defect is assigned to the appropriate team (Frontend/Backend/DevOps)
4. Developer fixes the defect and marks it as resolved
5. Tester verifies the fix and closes the defect or reopens if not fixed

### **9.2 Defect Points of Contact**

|  |  |
| --- | --- |
| **Component** | **Point of Contact** |
| Frontend | Devesh |
| Backend | Sonal |
| DevOps | Prajeeth |

### **9.3 Defect Priority and Severity Classification**

|  |  |
| --- | --- |
| **Severity** | **Description** |
| Critical | Application crash, data loss, security breach |
| High | Major feature not functioning, no workaround |
| Medium | Feature not functioning as expected, workaround exists |
| Low | Minor UI issues, cosmetic defects |

|  |  |
| --- | --- |
| **Priority** | **Description** |
| P1 | Must be fixed immediately |
| P2 | Should be fixed in current sprint |
| P3 | Should be fixed in next sprint |
| P4 | Can be fixed when time permits |

## **10. Risks and Mitigations**

|  |  |  |  |
| --- | --- | --- | --- |
| **Risk** | **Impact** | **Probability** | **Mitigation Strategy** |
| Resource unavailability | High | Medium | Maintain a backup resource plan with cross-trained team members |
| Environment instability | High | Medium | Establish environment maintenance schedule and monitoring |
| Tight timeline | Medium | High | Prioritize test cases using risk-based approach |
| Incomplete requirements | High | Medium | Regular requirement review sessions with stakeholders |
| Integration issues with payment gateways | Critical | Medium | Early integration testing with mock APIs |

## **11. Tools**

|  |  |
| --- | --- |
| **Testing Activity** | **Tool** |
| Test Management | TestRail |
| Defect Tracking | JIRA |
| Automation Testing | Selenium WebDriver (to add) |
| Performance Testing | JMeter (to add) |
| Security Testing | OWASP ZAP ( to add) |
| API Testing | Postman (to add) |
| Screenshots | Snipping Tool |
| Documentation | MS Office, Confluence |

## **Approvals**

This test strategy is subject to review and approval by the testing team lead and project stakeholders before test planning begins.

**Document Version:** 1.0  
**Last Updated:** Mar 2, 2024  
**Author:** Dalonda Ikhimokpa