Test Plan for OpenCart E-commerce

Objective

The objective of this test plan is to ensure that the e-commerce platform meets all the functional requirements, provides a user-friendly experience, is secure, performs well under various load conditions, and meets performance expectations.

Scope

This test plan covers testing for the following key aspects of the OpenCart E-commerce project:

- Login Functionality
- Registration Functionality
- Adding Products to Cart
- Checkout Process
- Payment Gateway Integration
- Order Management
- Performance Testing

Success Criteria

The success of testing will be evaluated based on the following criteria:

- Number of defects found and their severity.
- Time taken to complete the testing phase.
- User satisfaction ratings based on user feedback.

Inclusions

The following items are included in this test plan:

- Test strategy document
- Test cases document
- Test execution report

- Defect report
- Performance test report

Test Environments

The testing will be conducted in the following environments with varying configurations to ensure comprehensive coverage:

Operating Systems

- Windows 10 and 11
- macOS
- Linux (Ubuntu 20.04 LTS)

Browsers and Versions

- Google Chrome (latest stable version)
- Mozilla Firefox (latest stable version)
- Apple Safari (latest stable version)
- Microsoft Edge (latest stable version)

Device Types and Screen Sizes

- Desktop Computers (various screen sizes)
- Laptops (various screen sizes)
- Tablets (iOS and Android)
- Smartphones (iOS and Android)

Network Connectivity and Bandwidth

- High-Speed Broadband
- 4G and 5G Mobile Networks
- Simulated Low-Bandwidth (for performance testing)

Security Protocols and Authentication Methods

- HTTPS with TLS 1.2 and 1.3
- Two-factor Authentication (2FA)

- Secure APIs and Data Encryption

Payment Gateway Testing

- Payment gateway integration will be tested in a sandbox environment to ensure secure payment processing.

Environment	**URL**	**Access Permissions and Roles**
QA	demo.opencart.com	Testers: Full access
Pre Production	preprod.opencart.com	Testers: Limited access (staging)
User Acceptance Testing	uat.opencart.com	Testers: Limited access (user acceptance testing)
Production	app.opencart.com	Stakeholders: Read-only access, no testing permissions

Defect Reporting Procedure

1. Defect Identification

- Identify defects based on requirements deviation, user experience issues, or technical errors.

2. Reporting a Defect

- Use designated defect report template.
- Provide detailed defect information including description, steps, and environment.
- Attach relevant screenshots or logs when possible.

3. Triage and Prioritization

- Assign severity and priority levels.
- Assign defects to the appropriate team members for resolution.

4. Tracking and Management

- Use designated tracking software (e.g., Jira).

5. Roles and Responsibilities

- Testers identify and report defects.
- Developers review and resolve defects.
- Test Lead oversees the process.

6. Communication

- Regular updates to stakeholders through meetings, tracking tool, and notifications.

7. Metrics

- Measure the number of defects found, resolution time, closure rate, and defect success rate.

Role	**Responsibilities**	**POC**
Test Lead	Test planning, coordination, reporting, and tracking defects,.defect process management.	Aryan Vashishth
Frontend	Frontend and development, fixing defects in frontend.	Abhishek
Backend	Backtend development, fixing defects in backtend.	Mayank
Dev Ops	Test environment setup, maintenance, infrastructure management.	Vanshita

Test Strategy

Objectives:

1. Validate platform reliability and functionality.

2. Confirm security measures and identify vulnerabilities.
3. Assess system performance under various conditions.
4. Ensure compatibility across browsers and devices.

Key Strategies:

Component	**Description**
Test Levels	Cover unit, integration, system, and performance levels.
Test Types	Conduct functional, usability, security, performance, and compatibility testing.
Test Approach	Utilize both manual and automated testing for comprehensive coverage.
Defect Management	Track, prioritize, and resolve defects promptly using Jira.
Test Data	Use realistic test data to simulate various scenarios.
Test Deliverables	Prepare and document test deliverables including test plans, test cases and test reports
Test Environment	Establish and maintain test environments, including hardware and software setups.
Test Schedule	Create a detailed schedule outlining testing phases and milestones.
Resource Allocation	Allocate testing resources, including team members and testing tools.
Risk Management	Identify and mitigate potential risks that may impact the testing process.
Test Exit Criteria	Define the conditions that must be met for testing to conclude successfully.

Step 1: Test Case Development

- Create test scenarios and test cases for various features in scope.
- Use techniques like Equivalence Class Partition, Boundary Value Analysis, Decision Table Testing, State Transition Testing, and Use Case Testing.

- Apply expertise in creating test cases using Error Guessing and Exploratory Testing.
- Prioritize test cases.

Step 2: Testing Procedure

When we receive a testing request:

- Conduct smoke testing to check if essential functionalities work.
- If smoke testing fails – reject the build and wait for a stable one.
- With a stable build – perform in-depth testing using the prepared test cases.
- Multiple resources test the application on multiple supported environments simultaneously.
- Report bugs in a tracking tool and provide daily defect status emails.
- Testing types include Smoke Testing, Sanity Testing, Regression Testing, Retesting, Usability Testing, Functionality Testing, and UI Testing.
- Repeat test cycles until the desired product quality is achieved.

Step 3: Testing Best Practices

To enhance testing:

- Practice Context-Driven Testing – adapting to the application's context.
- Implement Shift Left Testing – start testing in early development stages.
- Utilize Exploratory Testing alongside scripted test cases.
- Perform End-to-End Flow Testing – simulate end-user scenarios involving multiple functionalities.

Test Schedule:

Following is the test schedule planned for the project –

Task	**Task Dates**
Creating Test Plan	[DATE]
Test Case Creation	[DATE]
Test Case Execution	[DATE]

Summary Reports Submission	[DATE]

** Two Sprints to Test the Application*

Test Deliverables:

The following are to be delivered to the client:

<u>**Deliverables**</u>	<u>**Description**</u>	<u>**Target Completion Date**</u>
Test Plan	Details on the scope of the Project, test schedule, resource requirements, test deliverables and schedule.	[DATE]
Functional Test Cases	Test Cases created for the scope defined.	[DATE]
Defect Reports	Detailed description of the defects identified along with screenshots and steps to reproduce on a daily basis.	[DATE]
Summary Reports	Summary Reports - Bugs by Bug#, Bugs by Functional Area and Bugs by priority.	[DATE]

Requirement Analysis Phase

Entry Criteria:

- We start when we receive the project's requirements or project details.

Exit Criteria:

- We finish when we have thoroughly understood all the project requirements, and any doubts are cleared.

Test Execution Phase

Entry Criteria:

- We begin testing when:
    1. The client has approved our Test Scenarios and Test Cases.
    2. The application is ready for testing.

Exit Criteria:

- We conclude testing when:
    - Test Case Reports and Defect Reports are prepared and ready.
    - The testing activities are completed according to the defined test plan.

Test Closure Phase

Entry Criteria:

- We enter this phase when Test Case Reports and Defect Reports are ready.

Exit Criteria:

- We complete this phase when we have generated the Test Summary Report, providing a comprehensive overview of the testing phase.
- All testing activities are formally closed, and testing documents are archived.