Table of Contents

|  |  |
| --- | --- |
| Section | Page No. |
| 1. Introduction |  |
| * 1. Purpose of the Test Plan |  |
| 1. Scope |  |
| 2.1 Feature to be Tested (Inclusions) |  |
| 2.2 Features Not to be Tested (Exclusions) |  |
| 1. Approach |  |
| 3.1 Testing Type |  |
| 3.2 Testing Level |  |
| 3.3 Testing Strategy |  |
| 1. Test Deliverables |  |
| 4.1 Test Plan |  |
| 4.2 Test Cases |  |
| 4.3 Test Logs |  |
| 4.4 Bug Reports |  |
| 4.5 Test Summary Reports |  |
| 1. Test Schedule |  |
| 5.1 Test Start Date |  |
| 5.2 Test End Date |  |
| 5.3 Test Case Execution |  |
| 5.4 Summary Reports Submission |  |
| 1. Resources |  |
| 6.1 Testing Team |  |
| 6.2 Tools |  |
| 1. Test Environment |  |
| 7.1 Hardware Requirements |  |
| 7.2 Software Requirements |  |
| 1. Test Case Design |  |
| 8.1 Test Case Format |  |
| 8.2 Traceability |  |
| 1. Entry and Exit Criteria |  |
| 9.2 Suspension and Resumption Criteria |  |
| 1. Risk and Mitigation |  |
| 1. Approvals |  |
| 1. Sign of Test Case |  |

1.Introduction

The OpenCart web application “'https://demo.opencart.com” is designed to provide users an intuitive interface to create account Search functionality, add to cart, compare the different products, buy the products and payment functionality is there. The main objective of this testing is to ensure that it meets the functional requirements and non-functional requirements outlined in the **Functional requirements specification** guaranteeing a seamless user experience and reliable performance. This high-level test plan outlines the projects scope test strategy, resources availability and deliverables.

**1.1 Purpose of the test plan**

The **Test Plan** for OpenCart aims to ensure the **stability, functionality, and usability** of the e-commerce platform by systematically testing its core features. This document outlines the **testing approach, scope, objectives, test environment, and execution strategy** to validate that OpenCart meets the required specifications and performs as expected.

* **Verify Core Functionalities** → Ensure product listing, cart, checkout, and payment processes work as intended.
* **Identify Defects** → Detect and document bugs before deployment to avoid critical failures.
* **Improve User Experience** → Validate UI/UX elements to ensure a smooth shopping experience.
* **Ensure Security & Performance** → Test for vulnerabilities and system responsiveness under load.
* **Enable Regression Testing** → Maintain test cases for future updates and enhancements.

2.1 Scope

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

* User Register
* User Login & Logout
* Forgot Password
* Product 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

**2.2 Exclusions (Features not to be tested)**

* Features mentioned in the inclusions
* Non-functional can be done by developers

**3.Approach**

The testing approach for OpenCart follows a structured and systematic method to ensure the platform's functionality, stability, and user experience. It includes different types of testing to cover all critical aspects of the application.

**1. Smoke Testing**

* Ensures that the core functionalities of OpenCart are working before detailed testing begins.
* Basic checks like homepage loading, admin login, and product display.
* If smoke tests fail, further testing is halted until issues are resolved.

**2. Functional Testing**

* Validates that OpenCart’s features work as expected.
* Covers key functionalities like:
  + Product listing and search
  + Adding items to the cart
  + Checkout and payment processing
  + Order confirmation and email notifications
* Test cases are executed, and results are compared with expected behaviour.

**3. Exploratory Testing**

* Unscripted testing to uncover hidden defects or edge cases.
* Testers navigate OpenCart freely, trying various scenarios that might break the system.
* Helps find usability issues and unexpected behaviours.

**4. UI/UX Testing**

* Evaluates the user interface for consistency and responsiveness.
* Ensures proper alignment, font readability, button clicks, and navigation flow.
* Checks compatibility on different screen sizes and browsers.

**5. Integration Testing**

* Verifies that different modules of OpenCart work together smoothly.
* Focuses on interactions like:
  + Payment gateway integration
  + Order processing and database updates
  + Email notifications after order placement

**6. Regression Testing**

* Ensures that new updates or bug fixes do not break existing functionality.
* Re-executes previously passed test cases after every release or code change.
* Maintains software stability over time.

**3.2 Testing Levels**

1. **Unit Testing**:
   * To be conducted by developers to validate individual components or modules (not part of this test plan but will align with it).
2. **Integration Testing**:
   * Validate interactions between different modules (e.g., login and dashboard).
3. **System Testing**:
   * End-to-end testing of the entire application to ensure it meets the requirements.
4. **Acceptance Testing**:
   * Validate the application against the business requirements to ensure it’s ready for deployment.

**3.3 Testing Strategy**

1. **Requirement Analysis**:
   * Understand the functional and non-functional requirements.
   * Identify test scenarios based on the features and workflows.
2. **Test Case Design**:
   * Write detailed test cases in alignment with the scope and testing types.
3. **Test Execution**:
   * Execute test cases in a prioritized order.
   * Perform manual testing for features listed in the test plan.
4. **Defect Reporting and Tracking**:
   * Log defects in **Jira** with detailed descriptions and steps to reproduce.
   * Track defect resolution and re-test fixed issues.
5. **Test Closure**:
   * Prepare a final test summary report documenting test coverage, defect trends, and unresolved issues.

**4. Test Deliverables**

The following are to be delivered to the client

|  |  |  |
| --- | --- | --- |
| **Deliverables** | **Description** | **Target Completion** |
| Test Plan |  |  |
| Test Cases |  |  |
| Test Logs |  |  |
| Bug Reports |  |  |
| Test Summary Reports |  |  |

**5. Test Schedule**

Following is the test schedule plan for the project

|  |  |
| --- | --- |
| Task | Time Duration |
| Test Creating Date |  |
| Test End Date |  |
| Test Case Execution |  |
| Summary Report |  |

.

**6. Resources**

**6.1 Testing Team**

|  |  |
| --- | --- |
| Category | Details |
| Test Leads |  |
| QA |  |
| Devs |  |
| Project Managers |  |

**6.2 Testing Tools**

|  |  |
| --- | --- |
| **Category** | **Details** |
| Test Management Tool |  |
| Documentation |  |
| Browser Testing |  |
| Version Control |  |
| Debugging Tools |  |
| Reporting |  |

**7. Test Environment**

**7.1 Hardware Environment**

* This application does not require any specific hardware configuration this testing will be carried out on only standard devices and browsers.

**7.2 Software Environment**

* Windows 10 -11
* Browser (Chrome or Safari)
* Android Mobile OS (Chrome)
* iPhone Mobile OS (Safari)

**8. Test Case Design**

**8.1 Test case format**

* **Test Case ID**: A unique identifier for the test case.
* **Test Scenario**: A brief description of what the test will validate.
* **Test Description**: A more detailed explanation of the test case’s purpose.
* **Pre-Conditions**: Any necessary setup or conditions required before executing the test case.
* **Test Steps**: A step-by-step guide on how to execute the test.
* **Expected Result**: The expected behaviour of the system after executing the test steps.
* **Actual Result**: The result after the test case execution (filled in after the test).
* **Status**: The outcome of the test case (Pass/Fail).
* **Defects**: Any defects identified during testing, referenced by defect IDs.

**8.2 Traceability**

Test case traceability will be maintained to ensure all requirements are covered, and defects can be traced back to test cases. This is typically managed using Jira.

The Traceability matrix will be maintained as follows.

|  |  |  |  |
| --- | --- | --- | --- |
| Test case ID | Requirement ID | Scenarios | Test Status |
|  |  |  |  |
|  |  |  |  |

**9. 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

**9.2 Suspension and Resumption Criteria**

Based on the Client decision, we will suspend and resume the Project. We will ramp up and ramp down the resources as per Client needs.

**10. Risks and Mitigations**

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

**11.Approvals and Sign-off**

Team will send different types of documents for Client Approval like below:

• Test Plan

• Test Scenarios

• Test Cases

• Reports

Testing will only continue to the next steps once these approvals are done.