Opencart project Test Plan

Project Name: OpenCart

- **(Frontend) Test Plan**
- **Admin Panel**

Document Author	Document Version	Revised Date	Approved By
Khaled Wael	1.0	21- 3- 2025	PM

Table of Content

1. Introduction

This document describes some test functionality of OpenCart website which is PHP-based, using a MySQL database and HTML components. It includes the objectives, scope, test responsibilities, entry and exit criteria, schedule of major milestones, approaches, risks, and acronyms.

2. Scope

2.1. In Scope

Front Page

- **Register**
  - Well-organized menu.
  - Ability to input "First Name," "Last Name," "E-Mail," "Telephone," "Password," etc. to create a new account.
  - Should have a back option.

- **Login and Logout**
  - Well-organized menu.
  - Ability to choose currency.
  - Options for "Contact," "My Account," "Wish List," "Shopping Cart," "Checkout," and "Item."
  - Forgot the password link.
  - Should have a back option.

- **Login and Logout**
  - Well-organized menu.
  - Ability to input E-Mail address.
  - Instruction text on top.
  - Back and continue option.

- **Search**
    - Organized Search bar

- **Product Compare**
    - Allows comparing specifications, features, and prices of multiple products.
    - Should have back and Show more/Show less options.
    - Add to cart and remove options.

- **Your Store Page**
  - Currency choice.
  - Options for "Contact," "My Account," "Wish List," "Shopping Cart," "Checkout," and "Item."
  - Clickable company logo on the top left.
  - Search product feature.
  - Organized by category.
  - Show all features.
  - "Show more/Show less" after product description.
  - More product options.
  - Clickable products.
  - Clickable cart sign-on product view.
  - "Wish list" and "compare" options.
  - Navigation to OpenCart Homepage.
  - Functional page numbers and next/previous buttons.
  - The price displayed on products.
  - Scrollbar on products.

- **Product Detail Page**
  - Product image with alternate views.
  - Product details: code, availability, price.
  - Cart functionality: select quantity, add to cart, wish list, compare.
  - Rating/Sharing: rate the product, and share on social media.
  - Description and review tabs.

- **Other features**
  - Add To Cart
  - Wish List
  - Shopping Cart
  - Currencies
  - Checkout page
  - My Account Page
  - Order History

- Download Page
- Contact Us Page
- Menu Option
- Footer Option
- Category option

Admin Panel

Customers Menu:

- **Filters Page:**
  1- Add Filters
  2- Delete Filters
  3- Filters Checklist

- **Options Page**
  1. **Add Options**
  2. **Delete Options**
  3. **Options Checklist**
  4. **Edit Options**

- **Reviews Page**
  1- Add Reviews
  2- Delete Reviews
  3- Reviews Checklist
  4- Edit Reviews

- **Manufacturers Page**
  1- Add Manufacturers
  2- Delete Manufacturers
  3- Manufacturers Checklist
  4- Edit Manufacturers

- **Information Page**
  1- Add Information
  2- Delete Information
  3- Information Checklist
  4- Edit Information

Sales Menu

- **Orders:**
  1- Add Orders
  2- Delete Orders
  3- Orders Checklist
  4- Edit Orders
  5- Filter Orders
  6- Print Orders Invoice
  7- Print Shipping list

- **Recurring Orders**
  1- Add Recurring Orders
  2- Delete Recurring Orders
  3- Recurring Orders Checklist
  4- Edit Recurring Orders
  5- Filter Recurring Orders

- **Returns**
  1- Add Returns
  2- Delete Returns
  3- Returns Checklist
  4- Edit Returns
  5- Filter Returns

Sales Menu

- **Customer Groups**
  1- Add Customer Groups
  2- Delete Customer Groups
  3- Customer Groups Checklist
  4- Edit Orders
  5- Filter Orders

- **Custom Fields**
  1- Add Custom Fields
  2- Delete Custom Fields
  3- Custom Fields Checklist
  4- Edit Custom Fields
  5- Filter Custom Fields

KPIs

1.Bug Detection Rate : 15-30%

2. Test Coverage Percentage

- Ensures at least 80% or more of the system's features are covered in test cases.
- Reduces the chances of missing defects in critical functionalities.
- Guarantees that major functionalities like product catalog, checkout, and admin configurations are verified.

3. API Response Time

- <200ms for product searches and browsing-related APIs.
- <500ms for checkout and payment processing APIs.
- <1s for complex database queries or bulk actions.
- Ensures a smooth user experience by optimizing system performance and backend response times.

Quality Objective

The objective of this Test Plan is to ensure reliability, system assurance, and optimal performance and capacity utilization. This test plan focuses on comparing the application's performance concerning end-user expectations with the current and new production environment, essentially enhancing and updating the existing system.

3. Testing Methodologies

3.1. Overview

This project will follow the Agile methodology, with work updated based on weekly iterations. The work is distributed according to a requirement-based strategy specified in this test plan. Each feature will be tested separately and together after each phase, with reports delivered according to the schedule. Test cases should be created during exploratory testing. Testing methodologies (In scope/Out of Scope) are given below:

3.2. Test Levels

In Scope

- **Functional Testing:** Black-box testing geared to the application's functional requirements.
- **Integration Testing:** Verify combined functionality after integration.
- **Performance Testing:** Check if the system meets performance requirements.
- **Cross-browser Testing:** Ensure system support across different browsers.
- **Security Testing (Payment):** Check application security against internal and external threats.
- **UAT (User Acceptance Testing):** From a tester’s perspective.
- **Regression Testing:** Validate existing functionalities as part of regression tests (Cart, Checkout Flow).
- **Progression Testing:** Validate new functionalities.
- **Fixed defects validation.**

Out of Scope

- Database Testing
- API Testing
- Automation Testing

3.3. Test Strategy

Test case preparation and execution

- QA will be preparing test cases. This will cover all scenarios for requirements.
- Peer review will be conducted from time to time for test cases and test matrix by the QA Lead.
- The respective Author of the Test Case and Test Matrix will provide comments or suggestions on test cases and test coverage if needed.

- The author will re-work the suggestions/improvements that have been given on Test Cases/Matrix and will send them for approval. Re-worked enhancements will be reviewed and approved by the reviewer.
- Respective QA will execute Test Cases on the test site based on designed scenarios, test cases, and test data.
- (Actual Result with Expected Result-> Pass/Fail) Test results will be updated in the test case document.

Defect login and reporting

- QA will log the defects/bugs in the prospective management tool. After this, QA will inform the respective developer about the defects/bugs that have been logged.

Retesting

- Respective QA will re-test for fixed bugs after the respective developer resolves it. Bug/defect status will be updated accordingly. In certain cases, if it’s required then regression testing will be done.

Deployment / Delivery

- Once all bugs/defects that have been reported after complete testing are fixed, afterward if no other bugs are found, the report will be deployed to the test site by PM.
- Once QA is done with the testing round on the test site and if it is required, the Report will be delivered along with sample output by email to the respective team member.
- QA will hand over the hard copy of the delivery documents to the respective developer.

3.4. Bug Triage

- Define the resolution type for each bug.
- Prioritize bugs and determine a schedule for all "To Be Fixed" bugs.

3.5. Suspension Criteria and Resumption Requirements

- Suspension criteria define when to suspend testing.
- Resumption criteria determine when testing can resume after suspension.

3.6. Test Completeness

- 100% test coverage.
- All manual test cases are executed.
- All open bugs are fixed or will be fixed in the next release.

4. Roles and Responsibility

Contact Name and Information	Responsibilities
Khaled Wael Talaat	Write and maintain the Test Plan and SRS documentation. - Assign tasks to team members and manage project workflow. - Review and validate test cases, ensuring coverage of all requirements. - Oversee API, automation, and manual testing. - Manage and track bug reports. - Ensure compliance with testing standards and best practices Develop and execute automated test scripts using testing frameworks. - Perform API testing to validate response codes, data consistency, and error handling. - Conduct performance testing for APIs under high traffic conditions. - Identify and report API-related defects.

	- Work with developers to resolve API integration issues. - Maintain test automation frameworks and update test scripts as needed.
Ahmed Gad	Develop and execute automated test scripts using testing frameworks. - Perform API testing to validate response codes, data consistency, and error handling. - Conduct performance testing for APIs under high traffic conditions. - Identify and report API-related defects. - Work with developers to resolve API integration issues. - Maintain test automation frameworks and update test scripts as needed.

Sara Abdulwahab	Develop and execute automated test scripts using testing frameworks. - Perform API testing to validate response codes, data consistency, and error handling. - Conduct performance testing for APIs under high traffic conditions. - Identify and report API-related defects. - Work with developers to resolve API integration issues. - Maintain test automation frameworks and update test scripts as needed.

Marwa Ali	Develop and execute automated test scripts using testing frameworks. - Perform API testing to validate response codes, data consistency, and error handling. - Conduct performance testing for APIs under high traffic conditions. - Identify and report API-related defects. - Work with developers to resolve API integration issues. - Maintain test automation frameworks and update test scripts as needed.

Maha Mahmdouh	Develop and execute automated test scripts using testing frameworks. - Perform API testing to validate response codes, data consistency, and error handling. - Conduct performance testing for APIs under high traffic conditions. - Identify and report API-related defects. - Work with developers to resolve API integration issues. - Maintain test automation frameworks and update test scripts as needed.

Roqaya Risk	Develop and execute automated test scripts using testing frameworks. - Perform API testing to validate response codes, data consistency, and error handling. - Conduct performance testing for APIs under high traffic conditions. - Identify and report API-related defects. - Work with developers to resolve API integration issues. - Maintain test automation frameworks and update test scripts as needed.

Maria Kameel	Develop and execute automated test scripts using testing frameworks. - Perform API testing to validate response codes, data consistency, and error handling. - Conduct performance testing for APIs under high traffic conditions. - Identify and report API-related defects. - Work with developers to resolve API integration issues. - Maintain test automation frameworks and update test scripts as needed.

5. Entry and Exit Criteria

5.1. Entry Criteria

- QA resources have completely understood the requirements and completed the FRS Document.
- QA resources have sound knowledge of functionality.
- Test Scenarios & Test Cases Approved.
- Test Plan
- All the necessary documentation, design, and requirements information should be available that will allow testers to operate the system and judge the correct behavior.
- Unit Test cases should pass
- Application Smoke Test Completed (If Applicable)

5.2. Exit Criteria

- Test Cases Execution Completed.
- A certain level of requirements coverage has been achieved.
- Outstanding Severity 1 & 2 Defects Completed
- No high priority is left outstanding.
- Completion of planned test case execution
- UAT Test Evidence Collected
- Test Closure Memo completed and signed off

6. Test Deliverables

- **Before Testing phase**
    - FRS document.
    - Test plans document.
    - Test scenario.
    - Test cases documents.
    - Test design specifications.

- **During Testing**
    - Test tool simulators.
    - Test data.
    - Test Traceability Matrix.
    - Error logs and execution logs.

- **After Testing Cycles**
    - Test results/reports.
    - Defect report.
    - Installation/test procedures guidelines.
    - Release notes.

7. Resources & Environment Needs

7.1. Resources

- **Description**
    - Database server with MySQL.
    - Web server with Apache.
    - LAN with at least 5 Mb/s speed.

- **Network**
    - Windows 10, 4GB RAM, 3.4GHz CPU.

7.2. Testing Tools

- **Test Case creation:** Microsoft Excel.
- **Test Case Tracking:** Microsoft Excel.
- **Test Case Execution:** Manual.
- **Test Case Management:** Microsoft Excel.
- **Defect Management:** Jira.
- **Test Reporting:** Microsoft Excel & Jira.

7.3. Configuration Management

- **Github**

7.4. Test Environment

- **Support Level (Browsers)**

  Windows 10: Edge, Chrome (latest), Firefox (latest), Safari (latest).

8. Risk & Assumptions

- **Risks:**

- Payment methods may fail with internet fluctuations.
- The system may fail to load on old/no-updated browsers with JavaScript disabled.

- **Assumptions:**

- For some optimization of resources, the test plan may fail.
- Maintain backup resources.
- Availability of test tools.
- Skill level of resources.

9. Terms/Acronyms

Terms/Acronyms	Definition
API	Application Programming Interface
PM	Project Manager
QA	Quality Assurance
RTM	Requirement Traceability Matrix
UAT	User Acceptance Testing