Table of Contents

1 Introduction

1.1 Purpose

The purpose of this Test Plan is to define the overall testing strategy, scope, approach, resources, environment, and schedule for the OpenCart Demo e-commerce project.

It describes the testing process that will ensure the functionality, usability, security, and performance of the system meet the expected standards.

The plan also outlines the test objectives, entry and exit criteria, defect management process, and roles and responsibilities of the testing team.

1.2 Project Overview

OpenCart Demo is a complete online shopping platform designed to simulate real-world e-commerce activities.
It enables users to register, browse products, manage shopping carts, perform checkout, and complete payment workflows.
The system also includes an administrative backend that allows product management, order tracking, and reporting.
Testing will focus on validating both frontend (user interface) and backend (admin and database) functionalities..

2 Scope

2.1 In-Scope

The testing efforts will focus on the following core functionalities and areas:

*• **Customer Frontend:** User Registration/Login, Product Browsing (search, filters), Shopping Cart Management, Checkout Process (with Cash on Delivery/Bank Transfer methods), Order History verification.*

*• **Admin Backend:** Admin Login, Product Creation/Editing, Order Status Updates, Customer Management (view/edit).*

*• **API Testing:** Verification of critical product catalog and customer data endpoints (using network analysis/proxy tools if direct API access is restricted).*

*• **Database Testing:** Verification of data integrity for key tables (e.g., oc_customer, oc_order, oc_product) following critical transactions.*

*• **Testing Types:** Manual Functional, UI Automation (Regression), API/Security, Database Integrity, Mobile Responsiveness Check.*

2.2 Out-of-Scope

The following areas will not be covered due to resource constraints or platform limitations:

- **Performance testing under heavy load conditions:**
  Load and stress testing require dedicated servers, monitoring tools, and control over the hosting environment — none of which are available for the shared demo site.
- **Full integration with real third-party payment gateways (e.g., live PayPal, Stripe):**
  Real payment processing involves sensitive financial data and business accounts; only demo payment simulations will be tested for safety and accessibility reasons.
- **Testing on legacy or non-major browsers (e.g., Internet Explorer, Opera):**
  The focus will be on modern, widely used browsers such as Chrome, Edge, and Firefox. Legacy browsers are deprecated and not relevant to current user behavior.
- **Source code unit testing (as this is a black-box testing project):**
  The testing team does not have access to the source code of the OpenCart Demo site. Testing is performed from a user perspective (black-box) rather than at the code level.
- **Third-party plugin integrations not part of the demo:**
  The demo environment includes only default OpenCart modules; external plugins are not available or required for the current testing scope.
- **Localization and multi-language testing:**
  The demo version is presented in English only, and language translation features are not included within the current testing objectives.

3 Testing Strategy

3.1 Test Objectives

The Test Objectives define the team's goals and link them directly to assigned tasks and testing areas, ensuring that all project requirements are fully validated.

1. Functional Coverage Objective

- Achieve **100% test coverage** for all *Critical* and *High-priority* user and admin workflows.
- Design and execute comprehensive **manual test cases** covering:
  - Positive scenarios (expected user actions)
  - Negative scenarios (invalid inputs, boundary cases)
  - Edge-case scenarios (rare or extreme conditions)
- Ensure all **core E-commerce business flows** — such as registration, login, product browsing, cart management, and checkout — meet specified functional requirements.

2. Automation Objective

- Set up the **Playwright/Selenium Automation Framework** and the **Core Regression Suite** within the first two weeks of testing.
- Tasks include:
    - Selecting the appropriate automation framework
    - Configuring the environment
    - Scripting the top 5 critical, repeatable user paths (e.g., *Guest Checkout*, *User Login/Logout*)
- Establish a stable **regression foundation** for continuous validation throughout the project lifecycle.

3. API and Security Objective

- Validate **functionality, status codes, and security responses** for all critical product and user API endpoints.
- Use tools like **Postman** or browser **network traffic analysis** to:
    - Design API test collections
    - Verify data consistency
    - Identify unhandled errors or unauthorized data exposure
- Report any discrepancies or vulnerabilities discovered during testing.

4. Data Integrity Objective

- Verify **data consistency and persistence** after key transactions (e.g., order placement, user registration).
- Perform indirect **database validation** via frontend actions and API checks.
- Confirm that updates are accurately reflected in platform reports or network responses.

5. Quality and Reporting Goal

- Target an overall **minimum 90% test pass rate** across all executed test scripts (manual and automated).
- The **Test Lead (Nervana Issac)** will:
    - Monitor daily execution results
    - Manage defect tracking and severity categorization
        - Lead the final review and presentation of all testing outcomes in the consolidated test summary report

3.2 Test Assumptions

- The demo site (https://demo.opencart.com/) will be used as the primary testing environment.
- All features in the demo version are assumed to be functional and available.
- Each team member will create and maintain individual test accounts for executing assigned test cases.
- Defects will be tracked and managed using Jira.
- The OpenCart Demo platform will remain publicly accessible and stable during the testing period.
- Required testing tools (Selenium/Playwright and Postman) will be properly installed on all team members' devices.
- Admin credentials will be secured and used strictly for testing purposes.

3.3 Data Approach

Test data will be manually prepared and maintained to simulate realistic user and product scenarios within the OpenCart Demo environment.

- **Customer Data:**

  Test customer accounts will be created specifically for this project using fake data generators or manual entry. Each test account will be properly documented and maintained for regression and UAT testing.

- **Product Data:**

  Existing demo products will be utilized for most test cases. Additional test products may be created via the Admin panel when needed to validate workflows such as product creation, price updates, and stock management.

- **Data Reset:**

  Since the demo environment may periodically reset its data, any critical test data (e.g., users or products) will be backed up or re-created at the start of each major test cycle.

- **Data Validation:**

  Database verification will be performed indirectly by observing consistent data behavior through the UI and API responses (e.g., verifying order history and inventory updates).

All test data will be stored securely and reused where applicable, especially for regression and API testing.

3.4 LEVEL OF TESTING

Test Type	Description	Responsible Parties
Manual Functional	Comprehensive manual execution of all core E-commerce business flows (Positive/Negative scenarios).	All team members
UI Automation (Regression)	Scripting of key, repeatable user journeys (e.g., Search & Checkout) using Playwright/Selenium.	All team members
API Testing	Validating data consistency, status codes, and security headers of service endpoints (e.g., product details, cart updates).	All team members
Database Integrity	Verifying data persistence and consistency in the backend DB (simulated via application actions and network checks).	All team members

3.5 MANUAL FUNCTIONAL

- User Registration, Login/Logout, and Profile Management
- Product Browsing, Search, Filters & Sorting
- Cart Operations (Add, Remove, Update, Apply Coupon)
- Checkout Process (Shipping, Payment, Order Placement)
- Order History & Confirmation
- Admin Panel: Product, Order, Customer Management
- Admin Reporting & Dashboard

Participants:

Tester’s Name	Department/ Area	Responsibility
Nirvana Isaac	Software Testing	Executes manual test cases for user registration and checkout workflows.
Hadeer Mahmoud	Software Testing	Executes manual tests for product search, cart, and payment flow.
Amira Samy	Software Testing	Executes manual tests for admin panel operations and reporting.
Youssef Khaled	Software Testing	Executes manual tests for UI validation and order history verification.

3.6 UI Automation (Regression)

- User Login/Logout

- Product Search & Filters

- Add to Cart

- Checkout & Order Confirmation

- View Order History

Participants:

Tester's Name	Department/ Area	Responsibility
Nirvana Isaac	Software Testing	Creates and runs scripts for login and checkout automation.
Hadeer Mahmoud	Software Testing	Automates product search and cart functionalities.
Amira Samy	Software Testing	Validates automated test execution and reports script results.
Youssef Khaled	Software Testing	Reviews regression outputs and records any failed test cases.

3.7 API Testing

-Product Details API

- Cart Update API

- User Authentication API

- Order Submission API

- Payment Simulation API

Participants:

Tester's Name	Department/ Area	Responsibility
Nirvana Isaac	Software Testing	Executes API collections for product and user endpoints using Postman.
Hadeer Mahmoud	Software Testing	Validates API responses for cart and order modules.
Amira Samy	Software Testing	Designs and maintains API test data.
Youssef Khaled	Software Testing	Performs response validation and defect logging in Jira.

3.8 Database Integrity

-User Registration Records
- Order Data Consistency
- Product Stock Updates
- Transaction & Payment History

Participants:

Tester's Name	Department/ Area	Responsibility
Nirvana Isaac	Software Testing	Verifies user and order data consistency after frontend actions.
Hadeer Mahmoud	Software Testing	Checks stock updates and transaction records through UI validation.
Amira Samy	Software Testing	Performs indirect data verification using API responses.
Youssef Khaled	Software Testing	Monitors end-to-end data flow consistency across modules.

4 Execution Strategy

4.1 Entry Criteria

Testing execution will officially commence only when the following conditions are met:

Entry Criteria	Test Team	Technical Team	Notes
Test environment(s) is available	✓	✓	
Test data is available	✓		
Code has been merged successfully		✓	
Development has completed unit testing		✓	
Test scripts are completed, reviewed and approved by the Project Team	✓		

4.2 EXIT CRITERIA

Testing will be considered complete and the final report will be submitted when the following conditions are met:

Exit Criteria	Test Team	Technical Team	Notes
100% Test Scripts executed	✓		
90% pass rate of Test Scripts	✓		
No open Critical and High severity defects	✓	✓	
All remaining defects are either cancelled or documented as Change Requests for a future release	✓		
All expected and actual results are captured and documented with the test script	✓		
All test metrics collected based on reports from daily and Weekly Status reports	✓		
All defects logged in -Defect Tracker/Spreadsheet	✓		
Test environment cleanup completed and a new back up of the environment	✓	✓	

4.3 VALIDATION AND DEFECT MANAGEMENT

Validation: *Test cases will be validated by cross-checking results between Manual (Hadeer), Automation (Youssef), and API (Amira) efforts, ensuring data consistency (Nirvana).*

Defect Tracking: *Defects will be tracked using a shared spreadsheet (or similar tool).*

Responsibility: *It is the responsibility of the tester who finds the defect to log it, document reproduction steps, and verify the fix (if applicable). Since the team cannot fix the demo, defects will be logged, categorized, and summarized in the final report.* Defects found during the Testing should be categorized as below:

Severity	Impact
1 (Critical)	▪ *Core E-commerce functions are completely blocked, preventing order completion or essential user actions.* ▪ *Examples: Checkout failure, Add-to-Cart not working, Login/Register page crash.*
2 (High)	▪ *Major features malfunction with no valid workaround, affecting normal user or admin operations.* ▪ *Examples: Incorrect product pricing, payment options missing, orders not appearing in admin dashboard.*
3 (Medium)	▪ *Functionality partially affected but a workaround exists; does not block the main flow.* ▪ *Examples: Coupon not applied correctly, product image missing, partial search result*
4 (Low)	▪ *Minor visual or cosmetic defects with minimal impact on usability.* ▪ *Examples: UI alignment issue, typo in product description, slow loading of non-critical element*

5 Environment Requirements

5.1 Test Environments

Environment Requirements

- **Target URL:** Testing will use the OpenCart Demo URL: https://demo.opencart.com/.
- **Operating Systems:** Windows 10/11 for development and compatibility checks.
- **Browsers:** Latest versions of Chrome, Firefox, and Edge.
- **Automation Tools:** The framework uses Java with Selenium WebDriver (IDE: IntelliJ/Eclipse).
- **API Tools: Postman Desktop Application** for API execution and validation.
- **DB Simulation:** MySQL/SQL Client (e.g., DBeaver) for conceptual data integrity checks, supplemented by network analysis tools.

Security Requirements

Security checks are integrated into API and Functional testing, focusing on observable requirements.

- **Secure Authentication:** Verify correct handling of invalid credentials and proper session termination (e.g., Admin logout).
- **Basic Input Validation:** Test against simple injection attempts (HTML/SQL) in input fields to ensure proper data sanitization.
- **API Data Exposure:** Ensure sensitive data (passwords/tokens) is masked or omitted from API responses and network traffic.
- **Generic Error Handling**: Check that error messages are generic and do not expose internal system details (e.g., database structure).

6 Significantly Impacted Division/College/Department

Business Area	Business Manager	Tester(s)
Software Testing Track		**Nirvana Isaac**
		Hadeer Mahmoud
		Amira Samy
		Youssef Khaled

7 Dependencies

- **Platform Stability and Uptime:** The testing effort is entirely dependent on the continued availability and stability of the OpenCart Demo site (https://demo.opencart.com/). Unanticipated downtime or significant platform changes will directly halt testing and impact the schedule.
- **Tool Availability and Setup:** Successful installation and configuration of all required tools are mandatory, including Java/Selenium WebDriver.
- **Internet Connectivity:** Stable and reliable internet access is essential for all team members to consistently access the remote demo environment and the shared tracking tools.
- **Effective Team Coordination:** Timely and effective daily communication is required among all team members to prevent overlapping work, ensure prompt defect logging, and facilitate quick status updates.