Test Plan Ecommerce

Test Plan Document

Project: NopCommerce e-Commerce Application

Version: 1.5

Date: January 8, 2026

Status: Approved Draft

1. Introduction

The **NopCommerce Test Plan** outlines the comprehensive strategy for verifying the quality, functionality, and stability of the public demo store. This document serves as a blueprint for the QA team, defining *what* to test, *how* to test, and *when* to test.

1.1 Objective

To ensure the NopCommerce frontend allows a Guest and Registered User to successfully browse, search, and purchase products without critical errors.

1.2 Scope

In-Scope Components:
- **Functional Testing**: Registration, Login, Cart, Checkout, Search.
- **UI/UX Testing**: Layout verification, broken images, spelling.
- **Cross-Browser Testing**: Chrome, Firefox, Edge.
- **Smoke Testing**: Critical path daily checks.

Out-of-Scope:
- Database security penetration testing.
- API load testing.
- Mobile App native testing (Responsive Web only).

2. Test Strategy

2.1 Functional Testing

We will validate that each feature conforms to the requirements defined in the SRS.

- **Positive Testing**: Verifying expected behavior with valid data (e.g., successful login).
- **Negative Testing**: Verifying error handling with invalid data (e.g., invalid email format).

2.2 Integration Testing

Verifying the interaction between modules:

- Add to Cart -> Cart Update.
- Checkout -> Order History Update.

2.3 Compatibility Testing

Ensuring the application renders correctly across:

- **Windows 10/11**: Chrome, Firefox, Edge.
- **MacOS**: Safari.
- **Mobile Web**: Android (Chrome) and iOS (Safari).

3. Test Environment

Component	Specification
Application URL	`https://demo.nopcommerce.com/`
Test Client OS	Windows 11 Enterprise
Browsers	Google Chrome (Latest), Mozilla Firefox (Latest)
Network	High-speed Broadband (> 50 Mbps)

4. Test Schedule

Phase	Activity	Start Date	End Date	Responsibility
Planning	Requirement Analysis, Test Plan Creation	Jan 01	Jan 03	QA Lead
Design	Writing Test Cases (Excel)	Jan 04	Jan 06	QA Engineer

Execution	Running Test Cases, Logging Defects	Jan 08	Jan 10	QA Team
Retesting	Validating Fixes	Jan 11	Jan 12	QA Team
Closure	Final Report & Sign-off	Jan 13	Jan 13	QA Lead

5. Defect Management (Bug Life Cycle)

We successfully track defects using the following lifecycle states:

1. **New**: The defect is posted for the first time.
2. **Open**: The lead approves the defect is valid.
3. **Assign**: The defect is assigned to a developer.
4. **Fixed**: Example fix is deployed.
5. **Retest**: QA verifies the fix.
6. **Closed**: The fix is confirmed.
7. **Reopen**: If the fix failed.

5.1 Severity Levels

- **Critical (S1)**: Blocker. System crash or Checkout failure. No workaround.
- **Major (S2)**: Major feature failure (e.g., Search not working). Workaround exists.
- **Minor (S3)**: UI alignment issues, spelling mistakes.

6. Risks & Mitigation

Risk	Impact	Mitigation Strategy
Data Reset	High - Test data is wiped daily.	Create new accounts for every test session.
3rd Party Downtime	Medium - Payment gateway errors.	Use "Check/Money Order" for functional valida
Requirement Changes	Medium - Rework needed.	Freeze requirements during execution phase.

7. Deliverables

At the end of the High-Level Test Cycle, the following artifacts will be delivered:
- **Test Plan** (This document).
- **Test Case Suite** (Excel: 50+ Scenarios).
- **Defect Report** (Excel: Summary of bugs found).
- **Traceability Matrix** (Mapping Requirement <-> Test Case).
- **Test Execution Report** (Pass/Fail summary).

8. Approvals

Name	Role	Date	Signature
John Doe	QA Lead	Jan 08	*[Signed]*
Jane Smith	Project Manager	-	

(End of Test Plan)