**Test Plan for NopCommerce Website Testing**

Website: <https://demo.nopcommerce.com/>  
Model: Agile SDLC  
Document Type: Test Plan

**1. Introduction**

This test plan outlines the strategy and approach to test the functionalities and user experience of the NopCommerce demo website. The document covers the scope of testing, objectives, and roles involved in the testing process to ensure a high-quality user experience and system reliability.

**1.1 Scope**

***1.1.1 In Scope***

• User registration and login  
• Logout functionality  
• Adding desktops to cart  
• Adding iPhone to wishlist  
• Comparing books and clearing comparison list  
• Updating product quantity in cart  
• Searching Adidas/Nike products and wishlist functionality  
• Opening Wishlist and Shopping Cart

***1.1.2 Out of Scope***

• Backend database performance and server load testing  
• API-level testing (covered in separate plan)  
• Mobile app testing (focus is only on web)

**1.2 Quality Objective**

The objective is to deliver a stable, bug-free, and functional demo platform that allows users to:  
• Register and log in successfully  
• Search and purchase products  
• Add/remove items in cart and wishlist  
• Compare products reliably

**1.3 Roles and Responsibilities**

|  |  |
| --- | --- |
| Role | Responsibility |
| Test Manager | Define test strategy, timelines, assign tasks, and monitor overall test progress. |
| QA Engineer | Design test cases, execute test scenarios, log defects, and perform regression testing. |
| Automation Tester | Develop and maintain Selenium-TestNG automation scripts. |
| Developer (Support) | Assist in reproducing and resolving defects reported by the QA team. |
| Product Owner | Validate business logic and sign off test coverage for each sprint/module. |

**2. Test Methodology (Agile Model)**

**2.1 Overview**

The Agile methodology is selected for this project as it supports continuous delivery and frequent feedback. It helps the team adapt quickly to changing requirements and ensures collaboration between testers and developers throughout sprint cycles.

**2.2 Test Levels**

• Unit Testing – Performed by developers  
• Integration Testing – Validate interaction between modules like registration, login, and cart  
• System Testing – End-to-end testing of user flows  
• Regression Testing – Run after each sprint  
• UAT – Conducted before release with stakeholders

**2.3 Bug Triage**

• Bugs prioritized based on severity and impact  
• Triage meetings held each sprint  
• Fix schedules planned for valid bugs

**2.4 Suspension Criteria and Resumption Requirements**

Testing will be suspended if major functionalities (e.g., login, checkout) are broken, test environment is unstable, or builds fail repeatedly. Testing will resume once blocking issues are fixed and stable builds are available.

**2.5 Test Completeness**

Testing is complete when:  
• All acceptance criteria are met  
• 100% of planned test cases executed  
• No critical/high defects remain  
• Regression runs pass successfully

**2.6 Sprints**

Sprint 1: User Registration, Login, Logout  
Sprint 2: Add Desktop to Cart, Add iPhone to Wishlist  
Sprint 3: Compare Books, Add Flower Bracelet (Qty=2)  
Sprint 4: Search Adidas/Nike, Add to Wishlist, Open Wishlist & Shopping Cart, Logout

**2.7 Test Scenarios**

|  |  |
| --- | --- |
| Scenario ID | Test Scenario |
| TS01 | Verify user can register with valid details |
| TS02 | Verify login succeeds with valid credentials |
| TS03 | Verify login fails with invalid credentials |
| TS04 | Verify user can logout successfully |
| TS05 | Verify desktops can be added to cart with configuration |
| TS06 | Verify iPhone can be added to wishlist |
| TS07 | Verify two books can be compared and cleared from list |
| TS08 | Verify Flower Bracelet quantity can be set to 2 and added to cart |
| TS09 | Verify Adidas/Nike products search works and item can be added to wishlist |
| TS10 | Verify Wishlist page opens successfully |
| TS11 | Verify Shopping Cart opens successfully |

**3. Test Deliverables**

• Test Plan  
• Test Cases  
• Bug Report  
• Test Report  
• Automation Scripts  
• Summary Report  
• Sign-off Sheet  
• Daily Updates

**4. Resource & Environment Needs**

**4.1 Testing Tools**

• Selenium WebDriver  
• TestNG  
• ExcelUtils (Apache POI)  
• JIRA (Bug Tracking)  
• Git/Maven  
• Jenkins (CI/CD)

**4.2 Test Environment**

• OS: Windows 10/11  
• Browsers: Chrome (latest), Firefox (latest)  
• IDE: Eclipse/IntelliJ  
• JDK: 8 or above  
• Separate environments for QA, Staging, and Production-like setup  
• Test data maintained in Excel files (RegisterData, LoginData)

**5. Terms/Acronyms**

|  |  |
| --- | --- |
| **TERM/ACRONYM** | **DEFINITION** |
| AUT | Application Under Test |
| QA | Quality Assurance |
| UAT | User Acceptance Testing |
| JIRA | Issue and bug tracking tool |
| CI/CD | Continuous Integration/Continuous Delivery |
| Selenium | Automation tool for web applications |
| TestNG | Testing framework for Java |
| ExcelUtils | Utility class for data-driven testing with Excel |
| Git | Version control system |
| Maven | Build automation tool |