**Testing Strategy Document demo.nopcommerce.com website**

**Website:** [**https://demo.nopcommerce.com**](https://demo.nopcommerce.com?utm_source=chatgpt.com) **Model: Agile SDLC**

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Author** | **Description** |
|  |  |  |  |

**Table of Contents**

1. Scope
2. Test Approach
3. Test Environment
4. Sprint Plan
5. Testing Tools
6. Release Control
7. Risk Analysis
8. Review and Approval

**1. In Scope**

**1.1**

**The testing process will validate all core functionalities of a nopCommerce-based website as seen on:**

admin-demo.nopcommerce.com

[**https://demo.nopcommerce.com**](https://demo.nopcommerce.com)

**1.2 Scope includes validation of:**

* Product catalog browsing
* Shopping cart and checkout
* User account management
* Admin functionalities like product, category, and order management

**1.3**

The project follows a Agile Model, where each phase is tested and reviewed before moving to the next.

**1.4 Out of Scope:**

* Mobile app testing
* External third-party ERP/system integrations
* Multilingual and multi-currency support (deferred to future phase)

**2. Test Approach**

**2.1 Process of Testing**

* 2.1.1 Requirements Analysis
* 2.1.2 Test Planning
* 2.1.3 Test Design
* 2.1.4 Test Execution
* 2.1.5 Defect Management
* 2.1.6 Test Closure

**2.2 Testing Levels**

* Unit Testing
* Integration Testing
* System Testing
* User Acceptance Testing (UAT)

**2.3 Roles & Responsibilities**

|  |  |
| --- | --- |
| **Role** | **Responsibility** |
| QA Manager | Strategy approval, risk mitigation |
| Test Lead | Coordination, daily status, reviews |
| Testers | Execute test cases, log & re-test defects |
| Developers | Fix issues, support QA during defect triage |

**2.4 Types of Testing**

* Functional Testing
* UI/UX Validation
* Security Testing
* Regression Testing
* Compatibility Testing
* Performance Testing

**2.5 Testing Tools and Methodology**

* Manual testing for user/admin flows
* Selenium + TestNG for regression automation
* JIRA for defect tracking

**2.6 Defect Management & Sign-Off**

* Log all defects with severity, steps, screenshots
* Weekly defect triage meetings
* Regression before every major release
* Sign-off only after all major/critical defects are closed

**3. Test Environment**

**3.1 Required Environments**

|  |  |
| --- | --- |
| **Environment** | **Details** |
| Development | Local builds with mock data |
| QA/Test | Main testing site with regular builds |
| UAT/Staging | Mirror of production; masked live data |
| Production | Live website; SSL-enabled; hosted on Azure or AWS |
| Performance | Load simulation environment for cart/checkout flows |

**3.2 Test Data Strategy**

* Synthetic test data for orders, customers, products
* Mask all PII in non-prod environments
* Daily incremental + weekly full backups
* Restore scripts before major test cycles

**4. Sprint Plan**

Sprint 1 (Weeks 1 & 2)

Epic: User Authentication (Week 1)

Tasks:

* Sign Up with Valid Details
* Password Validation (during signup)
* Sign Up with Existing Email
* Login with Correct Credentials
* Login with Wrong Password
* Password Validation check

Epic: Web UI Testing (Week 2)

Tasks:

* Understand UI requirements of nopCommerce
* Write UI test cases (buttons, forms, links)
* Check layout and alignment of forms/pages
* Validate field inputs (email, password, address)
* Click all buttons and links for navigation
* Check responsiveness (mobile/tablet)
* Cross-browser testing (Chrome, Firefox, Edge)

**Sprint 2 (Weeks 3 & 4)**

Epic: Product Management (Week 3)

Tasks:

* Verify adding products from categories (Computers, Electronics, Books, etc.)
* Validate product details page (title, description, price, images)
* Verify product configuration (RAM, HDD, etc.)
* Add products to cart from different categories

Epic: Shopping Cart & Checkout (Week 4)

Tasks:

* View shopping cart page
* Update product quantity in cart
* Apply discount code/coupon if available
* Estimate shipping cost (popup validation)
* Proceed to checkout page
* Validate checkout steps (billing, shipping, payment, confirmation)

**Sprint 3 (Weeks 5 & 6)**

Epic: Orders & Digital Products (Week 5)

Tasks:

* Validate order confirmation page
* Verify order appears in “My Account → Orders”
* Test Digital Downloads purchase flow
* Test Books category purchase flow

Epic: Resources & Content Pages (Week 6)

Tasks:

* Navigation to Blog, Help, and FAQ pages in nopCommerce
* Test search functionality within Help/FAQ
* Validate links, categories, and article content
* Check mobile responsiveness of resource pages
* Test contact us form/chatbot (if available)

**Sprint 4 (Weeks 7 & 8)**

Epic: Payment & Policies (Week 7)

Tasks:

* Validate available payment methods (COD, Card, PayPal)
* Verify tax/shipping calculations
* Validate order invoice generation
* Test cancellation & refund policies

Epic: User Profile & Account Settings (Week 8)

Tasks:

* Edit user profile details (name, email, address, password)
* Validate saved addresses and payment methods
* Check wishlist and compare products functionality
* Test newsletter subscription & unsubscribe

**5. Testing Tools**

**5.1 Automation & Test Execution**

|  |  |
| --- | --- |
| **Tool** | **Purpose** |
| Selenium | UI Automation |
| TestNG | Organize and execute scripts |
| Jenkins | Automated test runs |
| Postman | API Testing |
| JIRA | Defect Tracking |
| TestRail | Test Case Management |
| Allure | Visual Test Reports |

**5.2 Browser/Device Testing**

|  |  |
| --- | --- |
| **Tool** | **Purpose** |
| BrowserStack | Cross-browser/device validation |

**6. Release Control**

6.1 Follows Waterfall lifecycle with formal approvals  
6.2 Versioning follows vX.Y.Z (Major.Minor.Patch)  
6.3 All test cases must pass before deployment  
6.4 Critical fixes handled through hotfix branches with regression testing  
6.5 Maintain release notes, change logs, and sign-off records

**7. Risk Analysis**

**7.1 Major Risks**

* Changing or incomplete requirements
* Third-party plugin compatibility
* Payment security concerns
* Load performance issues
* UI issues across devices

**7.2 Risk Mitigation Strategies**

* Stakeholder workshops to freeze requirements
* Load and performance tests on staging
* Limit use to supported nopCommerce plugins
* Cross-browser testing using BrowserStack

**7.3 Contingency Plans**

* Backup rollback for critical deployments
* Manual fallback for cart/payment failures
* Use redundant cloud zones in AWS/Azure

**8. Review and Approval**

1. The Test Strategy document will be initially reviewed by the Test Lead or QA Manager.
2. After review, the document will be approved and signed off by:
   * Project Manager
   * Business Team
   * Development Team
3. Each Agile Model phase will begin only after the previous one is reviewed and signed off.
4. All review actions must be signed by the business, development, and project teams to ensure alignment.
5. A summary of changes (date, reviewer, description) will be maintained at the top of this document.
6. Final it is granted only after:
   * All major/critical defects are fixed
   * All test cases are executed
   * UAT is successfully completed
   * Stakeholders agree the product is release-ready