

Test Strategy Document - nopCommerce Storefront (demo.nopcommerce.com)

Application Under Test (AUT): nopCommerce Storefront (Public Demo)

URL: <https://demo.nopcommerce.com/>

Testing Methodology: Agile-based Continuous Testing

1. Scope

This Test Strategy covers validation of the nopCommerce public storefront available at <https://demo.nopcommerce.com/>. The goal is to ensure that customer-facing features are functional, usable, performant, secure, and meet business requirements.

Modules in Scope:

- Home page, navigation, and banners
- Product catalog: product listing pages (PLP), product details pages (PDP), attributes, variants
- Search & filtering (including autosuggest)
- Shopping cart and mini-cart
- Checkout flow (guest and registered users) and order placement
- Payments (using demo/mock gateways) and order confirmation
- User account management: registration, login, profile, addresses, change password
- Orders: view order history, order details
- Wishlist, compare products, product reviews and ratings
- Promotions, discounts, coupon codes
- Localization: currency, language, regional settings
- Responsive behavior across desktop/tablet/mobile
- Accessibility basics (WCAG checks) and SEO elements
- REST/API endpoints used by the storefront (where accessible)

Out of Scope:

- Live financial transactions with real payment processors (use mocks/sandbox)
- Third-party custom plugins not present in the demo instance
- Production environment changes

2. Test Approach

Process:

- Follow Agile sprint cadence with continuous testing and CI integration.
- Create acceptance criteria and test cases per user story.
- Prioritize automation for regression and critical business flows (checkout, login, search).

- Execute manual exploratory testing for new features and UX validation.

Testing Levels:

- Unit tests (developer responsibility).
- Integration tests (APIs, front-end to back-end flows).
- System/End-to-End tests (complete checkout, order lifecycle).
- Regression and Sanity tests per build.
- User Acceptance Testing (UAT) with stakeholders.

Types of Testing:

- Functional Testing
- UI / Usability Testing
- Cross-browser and Cross-platform Testing
- API Testing (where endpoints are available)
- Performance Testing (load, stress for key scenarios)
- Security Testing (OWASP Top 10 checks, authentication, session management)
- Accessibility (WCAG 2.1 basic checks)
- Localization and Internationalization Testing

3. Test Environment

Environments:

- QA Environment: mirror of demo instance. Use a dedicated QA instance where possible.
- Staging: for final validation before handoff.
- Browsers/Devices: Latest versions of Chrome, Firefox, Edge, Safari; Responsive testing on common screen sizes.

Hardware/Software:

- Server: Linux/Windows as per demo setup
- Database: SQL Server / MySQL (as applicable)
- Network: Stable internet and VPN where required

Test Data Requirements:

- Seeded products with variants, categories, and images.
- Test user accounts (guest, registered, admin where necessary).
- Sample coupon codes and promotions.
- Sample addresses for shipping and billing.

Environment Management:

- Maintain a versioned environment configuration in Confluence/Git.

- Database snapshots/seed scripts to reset state before major regression cycles.

4. Roles and Responsibilities

- QA Lead: Test strategy ownership, prioritization, test sign-off.
- QA Engineers: Test design, execution, automation, reporting.
- Developers: Unit/integration tests and fixing defects.
- Product Owner: Define acceptance criteria, UAT sign-off.
- Scrum Master: Facilitate process and remove impediments.
- DevOps: CI/CD and environment provisioning.

5. Test Design & Coverage

Key Scenarios to Automate (Priority):

1. User registration, login, logout, password reset.
2. Search (keyword, autosuggest), filters, and sorting on PLP.
3. Product details validation (images, price, inventory, attributes).
4. Add to cart, update quantity, remove item, mini-cart updates.
5. Checkout (guest and registered) including shipping and billing addresses.
6. Apply discount/coupon, validate prices and totals.
7. Order placement, order confirmation email (if available in demo), and order history.
8. Wishlist and move-to-cart flows.
9. Responsive checks for important pages.

Manual / Exploratory Areas:

- Usability of navigation and product discovery.
- Visual and content checks for banners and promotions.
- Accessibility spot checks (tab navigation, alt text).

6. Test Data & Test Case Management

Test Cases: Maintain in Jira / TestRail with traceability to requirements and user stories.

Test Data Sources:

- CSV/Excel or Database-driven data for automation.
- Randomized data where suitable (emails, addresses).

Data Privacy: Use anonymized or synthetic data only.

Test Case Types:

- Positive, negative, boundary, and edge cases.

7. Testing Tools

- Selenium WebDriver + TestNG / JUnit for UI automation (POM pattern).
- Playwright (alternative) for fast cross-browser automation.
- Postman / REST-Assured for API testing.
- JMeter / k6 for performance/load testing.
- OWASP ZAP or Burp (light) for security scans.
- Jira for test case tracking and defect management.
- Jenkins / GitHub Actions for CI integration.
- BrowserStack / Sauce Labs for cross-browser/device testing (optional).

8. Automation Strategy

Framework: Page Object Model (POM) + data-driven approach.

CI: Execute smoke and regression suites on each merge to develop branch.

Reporting: Generate HTML/Allure reports and store artifacts (screenshots, logs) on failures.

Failure Handling: Capture screenshots and page logs on failures.

9. Performance & Security

Performance Testing:

- Identify KPIs (page load < 3s for PLP/PDP, checkout < 5s under normal load).
- Simulate concurrent users for search and checkout flows.

Security Testing:

- Input validation, XSS, CSRF checks, session expiry, password storage verification.
- Verify secure cookies and HTTPS usage across the site.

10. Release Control & CI/CD

- Regression suite execution prior to release.
- Automated smoke tests for each deployed build.
- Maintain release notes and test sign-off in Jira/Confluence.
- Rollback plan for hotfixes.

11. Risk Analysis & Mitigation

Risks:

- Frequent UI changes may break automation.
- Demo instance limitations (shared public demo may be reset).
- Third-party integrations behavior differs in demo versus prod.

Mitigation:

- Keep automation robust with stable locators and abstractions.
- Maintain a private QA instance where possible and use mocks for integrations.
- Regularly refresh test data and maintain seeding scripts.

12. Metrics & Reporting

Key Metrics:

- Test case pass/fail rate per build.
- Defect density and mean time to resolve.
- Automation coverage and test execution time.
- Performance KPIs (response times, error rates).

Reporting Cadence:

- Daily build summary in stand-ups.
- Sprint-level QA report with trend charts.

13. Review and Approvals

Reviewers: QA Lead, Product Owner, Scrum Master.

Approvers: Project Manager, Business Stakeholders.

Sign-off: Document version, reviewer names and dates should be recorded.

14. References

- nopCommerce Demo: <https://demo.nopcommerce.com/>
- Selenium WebDriver Docs, Playwright Docs, JMeter Docs.
- OWASP Top 10.