

Test Strategy

Introduction

This document outlines the testing approach for the Magento e-commerce demo site <https://magento.softwaretestingboard.com/> as part of the QA project. The scope includes both manual exploratory testing and automated test scripting using Playwright.

Objectives

- Identify critical bugs affecting core user flows
- Validate key user journeys including search, product selection, cart operations, and checkout initiation
- Automate core flows for regression coverage and CI integration

Scope

- Manual exploratory testing of homepage navigation, search, product details, cart, and checkout initiation
- Automated testing of critical path: search -> product -> add to cart -> verify cart update
- CI integration using GitHub Actions to trigger test runs on code push and pull requests

Testing Types

- Exploratory Testing:
Conducted without predefined test cases to uncover functional, UI, and usability issues. Emphasis on user experience, consistency, and accessibility.
- Automated Testing:
Focused on high-value, repeatable test flows to support fast feedback and regression testing in CI pipelines.

Environment

- Test site: <https://magento.softwaretestingboard.com/>
- Browser: Chromium (via Playwright Test)
- Local setup: Node.js environment with Playwright dependencies installed

Tools & Frameworks

- Playwright: For UI automation and browser interaction
- GitHub Actions: For CI/CD pipeline integration
- Markdown: For structured bug reporting and documentation

Test Data

- Product listings and search terms available on the live Magento demo site
- Variations in product size, color, and quantity to simulate real-world cart scenarios

Test Coverage

- Search functionality on homepage and category filters
- Product detail pages: UI validation (size, color, image switching) and functionality
- Cart operations: add item, update quantity, remove, verify contents
- Checkout initiation and form field validation
- Accessibility and keyboard navigation issues discovered during manual testing

Risks & Mitigation

Risk: Dependency on demo site uptime

Mitigation: Prepare fallback instance or local test environment if needed

Risk: Limited automation scope

Mitigation: Expand test coverage manually through exploratory testing for edge cases and UI inconsistencies

Deliverables

- Detailed bug report documenting manual testing findings
- Automated test scripts located in the /tests directory (Playwright)
- CI pipeline configuration using GitHub Actions
- Project documentation: this test strategy, test cases, and README.md with setup instructions