QA Engineer Assessment - Automation & Manual Testing

Overview

This assessment evaluates your capabilities in both automation and manual testing.

Test Website

Demo Site: https://demowebshop.tricentis.com/

Part 1: Test Automation

Objective
Automate an end-to-end purchase flow using Playwright with a structured, maintainable approach.

Requirements

Technical Implementation
- Framework: Playwright (Node.js/TypeScript preferred, other languages acceptable)
- Architecture: Page Object Model for major screens (Home, Product, Cart, Checkout, Confirmation)
- Configuration: Environment configs and test data management
- Test Data: Data-driven testing with at least 2 different datasets

Test Scenario - E2E Purchase Flow
1. Open the demo website
2. Register a new user with unique email & password, then log in
3. Navigate to "14.1-inch Laptop" under Computers > Notebooks
4. Add product to cart and verify cart badge shows correct quantity
5. Proceed to checkout
6. Complete billing/shipping details (use dummy data)
7. Select shipping and payment methods
8. Complete the order
9. Validate "Thank you" page and capture Order Number

Required Assertions
- Product name and price consistency across PDP/cart/checkout
- Cart calculations (subtotal, tax, shipping if applicable)

- Address information accuracy on confirmation page
- Valid Order Number presence (use regex validation)

Quality Standards
- Stability: Use Playwright’s auto-waiting, avoid hard waits
- Observability: Screenshots on failure, video recording, trace collection
- Code Quality: Clean, maintainable code with meaningful comments

Deliverables
- Complete test suite with setup instructions
- Single command execution (e.g., npm ci && npx playwright test)
- README with setup steps, tools/versions, scenario descriptions

Part 2: Manual Test Cases

Objective
Design comprehensive manual test cases for the same e-commerce purchase flow.

Requirements

Create 20+ test cases covering:

- Positive scenarios: Happy path flows
- Negative scenarios: Invalid inputs, error handling
- Boundary cases: Edge conditions, limits
- Cross-browser compatibility: Different browsers/devices
- Accessibility: Basic accessibility scenarios

Test Case Format
Test ID: TC_XXX_ComponentName_Scenario
Objective: [Brief description]
Priority: [P0/P1/P2/P3]
Type: [Functional/UI/Integration/etc.]

Prerequisites:

[List any setup requirements]

Test Data:

[Required data inputs]

Test Steps:

[Step-by-step actions]

Expected Results:

[Expected outcome]

Acceptance Criteria:

[Pass/fail criteria]

Part 3: Regression Testing Strategy

Objective
Design a regression testing approach for the e-commerce application.

Deliverables

Create a regression testing document covering:

Risk Assessment Matrix
- Critical user journeys identification
- Feature priority classification
- Impact vs. probability analysis

Test Selection Strategy
- Smoke test suite definition (must-run tests)
- Full regression vs. targeted regression criteria
- Automation vs. manual testing decisions

Execution Framework
- Test case prioritization approach
- Regression suite maintenance strategy
- Execution timeline recommendations

Submission Requirements

Repository Structure
/project-root
■■■ README.md
■■■ package.json (or equivalent)
■■■ playwright.config.ts
■■■ /tests
■ ■■■ /e2e
■ ■■■ /specs
■■■ /pages
■■■ /fixtures
■■■ /utils
■■■ /test-data
■■■ /manual-test-cases
■■■ /regression-strategy

Documentation Requirements

Your README must include:

- Setup Instructions: Exact commands to run the suite
- Environment: Tools, versions, OS used
- Test Scenarios: Description of automated and manual test coverage
- Artifacts: Where to find traces, videos, screenshots

- Assumptions: Any testing assumptions or limitations noted

Evaluation Focus Areas

- Technical competency in automation framework implementation
- Testing strategy and comprehensive scenario coverage
- Code quality and maintainability considerations
- Documentation clarity and professional presentation
- Critical thinking demonstrated through test case design and edge case identification

Submission

Provide either:

- GitHub repository link (preferred)
- ZIP file with complete project

Include any notes, assumptions, or special considerations we should be aware of during evaluation.

Good luck!