Software Testing - Graduation Project: Automation Exercise Website (E-Commerce platform)

Comprehensive Manual & Automation Testing Implementation

by

1. Ahmed Fayed   2. Alaa Mohamed

3. Essam Ehab   4. Rawan Mohamed

5. Yousef Samy

AGENDA

1. Introduction: Project Overview
2. Objectives & Key Features
3. Testing Scope
4. Testing Strategy
5. Manual Testing Approach
6. Automation Testing Approach
7. Results & Findings
8. Conclusion & Lessons Learned

Introduction

Project Overview:

Automation Exercise is a full-featured e-commerce system that simulates every step of the online shopping journey:

- An online e-commerce platform simulating a real-world retail shopping experience
- Offers user registration, product browsing, shopping cart management, and checkout
- Supports secure payments (credit cards, PayPal), discount codes, and downloadable PDF invoices
- Enables post-purchase actions: order history, re-orders, and customer support
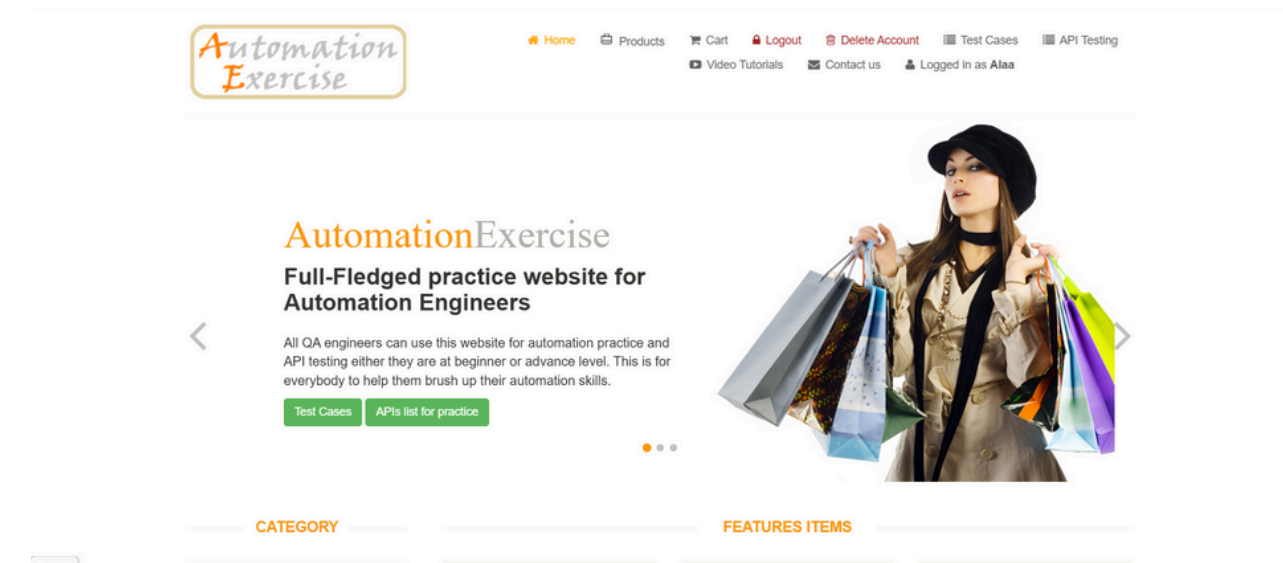
Test Objectives

1. Verify that all functional requirements behave as expected
2. Validate user interface (UI) elements for correctness, responsiveness, and consistency
3. Ensure smooth user workflows (e.g., login → cart → checkout → payment)
4. Identify functional defects through both manual and automated testing

Key Features Tested:

1. **User Authentication:** Registration, login, and account deletion.
2. **Product Management:** Browsing, searching, and filtering products using brands and categories.
3. **Shopping Cart:** Adding/removing items, and card viewing.
4. **Order Management & Checkout:** checkout processing, payment, and invoice download.
5. **Support:** Submitting inquiries via the contact form.

Scope of Testing

Included:
1. User actions such as authentication, product browsing, cart use, and checkout
2. Frontend features: form inputs, buttons, navigation, feedback messages
3. Invoice generation, order confirmation, product reviews, contact form, and subscriptions
4. Functional flow testing via the user interface only

Excluded:
1. Backend logic not exposed in the UI
2. API behavior is not visible from the frontend
3. Database structure,
4. Non-functional requirements, such as performance testing and security testing

Functional Requirements

Authentication & User Management

1. Users can sign up with a unique email and secure password
2. Login requires valid credentials.
3. Users stay logged in until they log out or the session times out
4. Accounts can be deleted, and personal data is removed securely

Product Management

1. Products are displayed with name, image, price, availability, and brand
2. Users can search, filter (by category/brand), and sort products
3. "Out of Stock" items are marked and cannot be purchased

Product Reviews

1. Users can submit reviews with their name, email, and a message
2. Reviews are displayed on product detail pages

FUNCTIONAL REQUIREMENTS

Shopping Cart & Checkout

1. Users can add, update, or remove items from the cart
2. Checkout is allowed only when the cart is not empty
3. Users enter shipping/billing addresses, apply discount codes, and review order summaries

Payments & Invoicing

1. Supports secure credit card and PayPal payments
2. Invalid payment details are rejected with clear messages
3. A unique order number is generated upon successful payment
4. Invoices are created automatically and are available as downloadable PDFs

Communication & Support

1. Users can subscribe to newsletters
2. The Contact Us form allows submitting inquiries and feedback
3. Confirmation emails are sent after order placement and form submission

Testing Strategy: Approaches

Manual Testing

- Executed step-by-step test cases to validate functional requirements and UI behavior
- Used structured test cases to cover all user scenarios, including edge cases

Automation Testing

- Developed UI test scripts using Selenium WebDriver with the Page Object Model (POM)
- Focused on repetitive flows: login, product actions, cart management, and checkout
- Automation scripts ensured faster re-testing and reduced human error

Manual Testing Approach

Testing Process

Created structured test cases based on the Software Requirements Specification (SRS)

Executed tests across all core user flows:

- User registration: signUp
- login/logout
- Products
- search and filtering functionality
- Cart operations
- checkout process
- Payment simulation
- invoice display
- Subscription
- Contact Us form interactions

Used positive and negative testing to validate expected and edge-case behaviors

Test Cases Summary

For each feature, we outline how test cases were designed (scope) and what happened after execution (post-test process).

Test analysis & design

👤 User Registration (SignUp)
• Input validation: required fields, email format, password rules
• Duplicate-email scenario
• Success flow confirmation

🔐 Login / Logout (SignIn)
• Valid and invalid credential combinations
• Session persistence and timeout
• Logout functionality

🏷️

Test Execution& design

👤 User Registration (SignUp)
Total Cases: 91   Pass: 44   Fail: 47
Key Issues:
Inconsistent email/password validation
Fields input constraints didn't apply

🔐 Login / Logout (SignIn)
Total Cases: 18 Pass: 12 Fail: 6
Key Issues:
Missing or unclear error messages on invalid login

Test Cases Summary

For each feature, we outline how test cases were designed (scope) and what happened after execution (post-test process).

🛍️ **Product Search & Filtering**
- Product listing display validation
- Product details page validation
- Product stock management
- Adding review on product
- Keyword search results accuracy
- Category, brand, and price-range filters
- "No results" and edge-case behavior

🛍️ **Product Listing**
Total Cases: 20 Pass: 11 Fail: 9
Key Issues:
No Zoom Functionality
Stock status not updating correctly
Inconsistent email validation on the product review form

🔍 **Search & Filtering**
Total Cases: 8 Pass: 6 Fail: 2
Key Issues:
No Price-range filter

Test Cases Summary

For each feature, we outline how test cases were designed (scope) and what happened after execution (post-test process).

Test analysis & design

Test Execution& design

🛒 **Shopping Cart Operations**
• Add/remove/update quantity
• Cart persistence across login/logout
• Empty-cart checkout prevention

🏷️ **Checkout Process**
• Address form validation (shipping vs. billing)
• Discount code application
• Out-of-stock item handling

🛒 **Shopping Cart**
Total Cases: 8 Pass: 6 Fail: 2
Key Issues:
Quantity fields non-editable
Cart not always cleared on logout/login

🏷️ **Checkout Process**
Total Cases: 16 Pass: 5 Fail: 11
Key Issues:
Cannot apply discount code
The address form lacks proper validation
Cannot edit the order list

Test Case Summary

For each feature, we outline how test cases were designed (scope) and what happened after execution (post-test process).

💳 Payment Simulation

- Credit-card field validations (CVC, expiry)
- Special character handling in name fields
- Error messaging on invalid inputs

📄 Invoice Display

- PDF generation and download format
- Sequential invoice numbering
- Content accuracy (order, address, tax)

💳 Payment Processing

Total Cases: 30 Pass: 9 Fail: 21

Key Issues:

Name fields accept special characters/numbers
No clear error on invalid CVC or expiry

📄 Invoice Generation

Total Cases: 4 Pass: 0 Fail: 4

Key Issues:

PDF invoices not generated (download as .txt)
Missing sequential invoice numbering

Test Case Summary

For each feature, we outline how test cases were designed (scope) and what happened after execution (post-test process).

Test analysis & design

Test Execution& design

✉ Subscription

• Email format validation (spaces, dots, "@")
Confirmation for subscription

Contact Us

• Fields input validations
• Form submission confirmation
• can send an attachment

✉ Subscription

Total Cases: 15 Pass: 8 Fail: 7
Key Issues:
Leading/trailing spaces in email are accepted
Dot-position and "@" validation gaps

✉ Contact Us

Total Cases: 27 Pass: 14 Fail: 13
Key Issues:
Inconsistent email validation
No submission confirmation is displayed

Automation Testing Approach

⚙ Framework & Architecture

Language & Tools
- Java with TestNG as the test runner and assertion framework
- Selenium WebDriver for browser automation

Design Pattern
Page Object Model (POM)
- pages.HomePage and pages.ContactUsPage encapsulates locators and interactions
- TestBase class handles WebDriver setup/teardown

Test Data Management
- @DataProvider methods supply input sets for data-driven testing
- Separate providers for simple form submissions and file-upload scenarios

Test Organization
- Tests grouped into packages (e.g., tests.ContactUs_happyScenario)
- @Test(priority=…) controls execution order
- Clear separation of "happy path" vs. attachment scenarios

Development Process

Test Case Identification
- Review the test cases listed in the manual
- Critical user flows (Login, Registration, Product Search, Checkout)
- Map each scenario to specific UI flows and data requirements

Page Object Creation
- Build POM classes for each page element and action (e.g., userCanContactUs(...))
- Encapsulate waits and element checks within page methods

Test Implementation
- Write test scripts using TestNG with data-driven @DataProvider methods
- Leverage POM methods to perform UI actions and assertions
- Organize tests by feature and priority for clarity and maintainability

Execution & Reporting
- Run suites via TestNG XML or IDE integration
- Generate HTML/XML reports automatically

Automation Test Coverage

🔓 **Authentication & Account Management**
- Register a new user flow, including newsletter opt-in and account deletion
- Log in with valid credentials and log out.
- Error handling for incorrect login attempts
- Secure logout and redirection back to the login page
- Validation for attempting to sign up with an existing email
- Navigation verification for the "Test Cases" page link

🛍️ **Product Catalog & Search**
- Display all products with correct details and "View Product" navigation
- Keyword search results and page header validation
- Category sidebar links and correct page headers
- Brand sidebar filtering behavior
- Cart item persistence after login following a search
- Add-to-cart functionality from the "Recommended Items" section

Automation Test Coverage

🛒 **Shopping Cart Operations**
- Adding multiple products and verifying total price calculations
- Updating product quantity and reflecting changes in the cart
- Cart persistence across login and logout actions
- Removing items via the cart's "X" button functionality

💳 **Checkout & Payment**
- Guest user registration during checkout, address confirmation, payment, and order confirmation
- Checkout flow for pre-registered users, including account cleanup
- Checkout for returning users, payment processing, and post-order deletion
- Verification of the shipping and billing address display before payment

Automation Test Coverage

📄 **Invoice Generation**

- Visibility of invoice link immediately after placing an order
- Automatic PDF invoice download and sequential invoice numbering

📧 **Communication & Engagement**

- Contact Us form submission both with and without attachments, and success message validation
- Newsletter subscription functionality on home and cart pages
- Product review submission and confirmation message display

THANK YOU