

Test Plan Document for Clothing Store Automation Testing

Introduction

This document outlines the test strategy, test cases, and the approach for automating the user flows on the clothing store website, leveraging Cypress as the test automation framework. It aims to ensure that all critical functionalities perform as expected through comprehensive testing, including regression and smoke tests.

Test Cases

Automation is available at repo: [edgarysabel/automation-exercise \(github.com\)](https://github.com/edgarysabel/automation-exercise)

1. Select Product
 - Objective: Verify that users can successfully search for products.
 - Steps: Navigate to the home screen, scroll to the middle of the screen and find a specific product, select this product, and verify that the product redirects to the product page.
2. Add Product to Cart
 - Objective: Ensure users can add products to their cart.
 - Steps: Select a product, adjust the quantity if needed, add it to the cart, and verify that the cart reflects the added product correctly.
3. Sign Up
 - Objective: Test the account creation process.
 - Steps: Navigate to the sign-up page, fill in the required information, submit the form, and verify successful account creation.
4. Checkout
 - Objective: Confirm the checkout process works as intended.
 - Steps: With products in the cart, proceed to checkout, enter payment information, place the order, and verify order confirmation.
5. Logout
 - Objective: Ensure the logout functionality works correctly.
 - Steps: From any authenticated state, initiate the logout process and verify that the user is logged out.
6. Login
 - Objective: Verify login functionality for returning users.
 - Steps: Navigate to the login page, enter user credentials, submit, and confirm successful login.
7. Contact Us
 - Objective: Test the "Contact Us" feature for user inquiries.
 - Steps: Navigate to the "Contact Us" page, fill in the required fields, submit the form, and verify successful submission.

Test Strategy

- Smoke Testing: Conduct smoke tests to cover the most critical paths, such as login, search product, and checkout processes, ensuring the application's core functionality works before moving to more detailed testing.
- Regression Testing: After each update or bug fix, run a full suite of tests to ensure that new changes haven't adversely affected existing functionalities.
- Data-Driven Testing: Use varied datasets for testing account creation, product searches, and checkout processes to simulate real-world usage scenarios.
- Cross-Browser Testing: Ensure compatibility across different browsers by running tests on Chrome, Firefox, Safari, and Edge.
- Responsive Testing: Verify the application's responsiveness and functionality on various devices, including desktops, tablets, and smartphones.

Test Environment Setup

Please refer to the ReadMe for environment setup and execution. Available in the repo:

Reporting

- Generate custom reports using Mochawesome for more comprehensive insights.

Reports can be found in Github Pages generate automatically by the CI: <https://edgarysabel.github.io/automation-exercise/>

Deliverables

- Cypress Test Suite: A complete set of Cypress test files (.spec.js) covering all defined test cases.
- Test Execution Video: A video recording of the test suite execution, showcasing the passing and failing tests.
- Test Report: A detailed test report generated by Cypress Dashboard or Mochawesome, including test outcomes, execution time, and logs.

This document serves as a foundation for implementing a robust test automation strategy using Cypress, ensuring the clothing store website operates flawlessly across all user flows and scenarios.