Test Plan

Project Title:- Test Automation for Demo webshop Using cypress and Page Object Model(POM)

Prepared By- Shiv Shanker Gupta
Date - 15/04/2024

1. Introduction

This document outlines the test plan for the automation of the Demo Web Shop application (http://demowebshop.tricentis.com) using Cypress as the test automation tool and the Page Object Model (POM) design pattern. The purpose of this test plan is to define the testing strategy, scope, objectives, deliverables, timeline, resources, and risk mitigation to ensure a high-quality and reliable user experience

2. Objective

To design and implement a scalable, maintainable, and reusable automation framework using Cypress and the POM approach to validate critical user workflows and functionalities of the Demo Web Shop application including navigation, search, product management, cart handling, checkout, and user authentication

3. Test Scope

3.1 In-Scope: -

Homepage navigation and redirection - Product search, filtering, and browsing - Product detail verification - Add/remove/update items in cart - Checkout workflow for guest and registered users - Login and registration functionality - Responsive behavior testing (mobile, tablet, desktop)

3.2 Out-of-Scope: -

Backend API and database validation - Payment gateway real transaction validation - Performance/load testing

4. Test Strategy

The test automation suite will be developed using Cypress with the POM structure for better code readability, reuse, and maintainability. The suite will cover functional, regression, and UI validations.

4.1 Test Types: -

Smoke Testing Ensuring application readiness. - Functional Testing Validating workflows and features. - Regression Testing Running automated scripts after each release. - Cross-Device Testing Verifying layout and functionality across devices using cy.viewport().

5. Test Design & Execution

5.1 Folder Structure:

cypress/
e2e/ -> Test scripts
pages/ -> Page Object classes
fixtures/ -> Static test data
support/ -> Custom commands and setup

5.2 Page Objects:

Separate JS classes will be created for:
- HomePage
- ProductPage
- CartPage
- CheckoutPage
- LoginPage

5.3 Sample Test Cases:

- Search for a product and validate results
- Add product to cart and verify total - Register new user and confirm login
- Checkout as a guest with valid address and payment

5.4 Test Data:

Stored in JSON format in fixtures/. Data will be parameterized for reusability.

5.5 Test Execution:

- CLI: npx cypress run
- GUI: npx cypress open

6. Test Environment

Component	Details
Application URL	http://demowebshop.tricentis.com
Test Framework	Cypress v13.x
Language	JavaScript (ES6+)
Browsers	Chrome (default),Firefox
OS	Windows/macOS/Linux
CI/CD (Optional)	GitHub Actions/Jenkins

7. Test Deliverables -

Cypress Automation Test Suite
- Modular Page Object Classes
- Test Data Sets
- Test Scenario Documentation
- Bug Reports
- Summary Execution Report
- Test Coverage Mind Map

8. Risks & Mitigation

Risk	Mitigation Strategy
UI changes may break tests	Use robust selectors, update objects timely
Data dependency across tests	Use mock or fixture data
Flaky tests on dynamic elements	Use appropriate waits and assertions

9. Timeline

Phase	Duration
Cypress Setup	1 day
POM Implementation	2 days
Script Development	34 days
Test Execution & Debugging	2 days
Documentation & Reporting	1 day

10. Approval

Prepared by: [Shiv Shanker Gupta]
Reviewed by: [Anjali Shaw]
Approved by: [Masai School]

This document outlines a detailed and scalable approach to automating the Demo Web Shop using Cypress and the POM pattern. It aims to ensure comprehensive test coverage, rapid feedback through automation, and maintainability for future enhancements.

The proposed strategy will act as a reliable reference for execution, debugging, and reporting across all phases of the test life cycle.