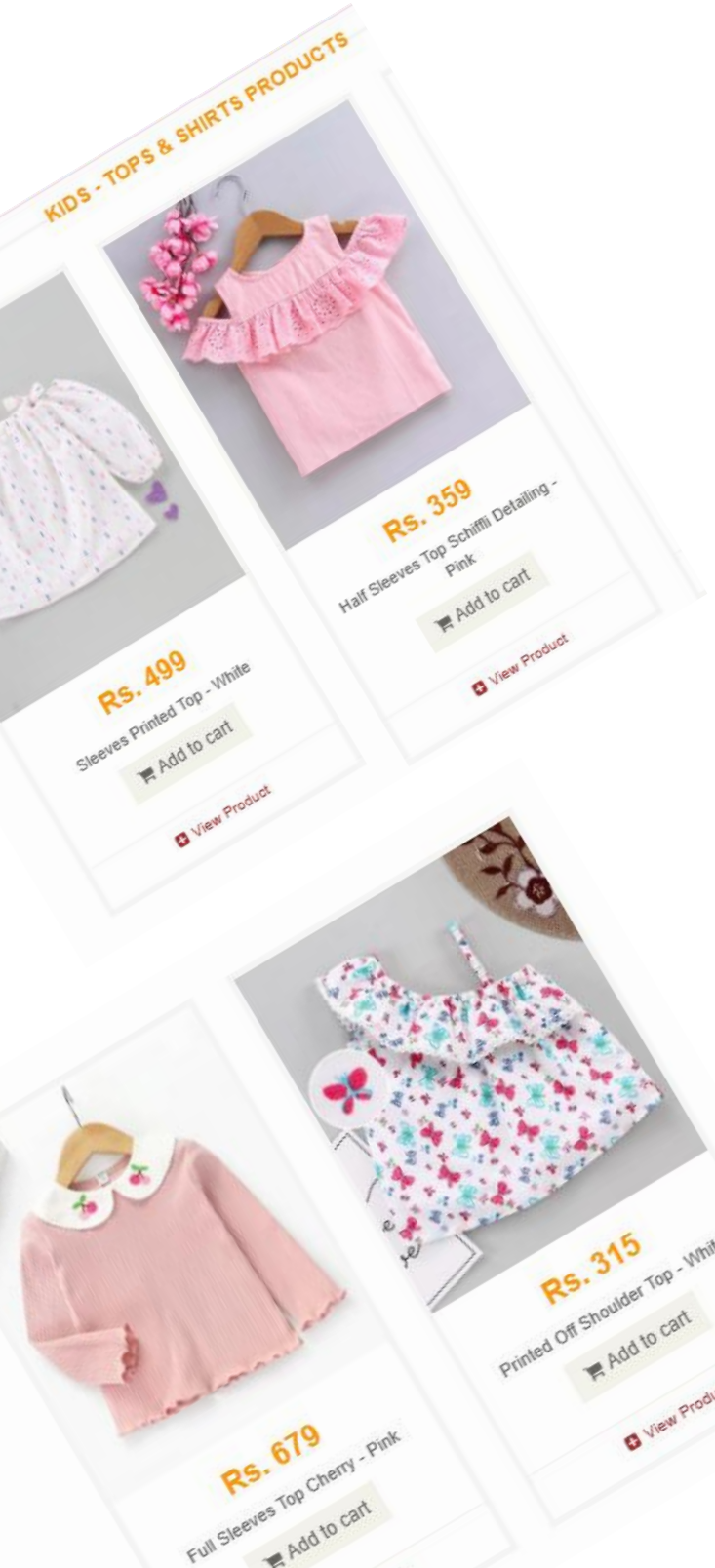
Test Plan



Automation Exercise

Products > Kids > Tops & Shirts

Prepared by: Quality Assurance Project Team

1. Introduction

This plan explains how we will test the Automation Exercise website. The goal is to make sure everything works properly – the pages load, buttons work, products show up correctly, and users can add items to the cart and checkout. "Most of the tests are **automation-focused, using Selenium WebDriver, TestNG, and Java."**

2. Objectives

TO :

- Make sure the website looks and behaves correctly.
- Test all important user actions: browsing products, adding to cart, checkout, subscription, and search.
- Find and report any issues so they can be fixed.
- Automate repetitive tasks so testing is faster next time.

3. Scope

(In-Scope):

- **Homepage:** logo, navigation, banners, features, newsletter, footer.
- **Product pages:**
  - All products listing and check
  - **Kids category** and its sub-categories: **Dress** and **Tops & Shirts**
    Testing it manually and most of cases tested automation
  - Product details page: name, price, availability, condition, quantity, Add to Cart
- **Cart & Checkout:** quantity, totals, login/signup, payment, order confirmation
- **Search functionality** and hover effects
- **Newsletter subscription and updates**

(Out-of-Scope):

- Performance testing (website speed, stress testing)
- **Man/Women categories** (they are not included in this test)
- **Brand filters or brand-specific features**
- Backend database or server-side validation

4. Test Items

- Homepage
- Product Pages (All Products, Category, Details)
- Cart & Checkout
- Newsletter subscription
- Search bar

5. Test Strategy & Approach

- Automated **functional and UI tests** using Selenium WebDriver + TestNG.
- Validate all visible elements: text, buttons, images, hover effects, clickable links.
- Functional tests: Add to Cart, search, checkout, subscriptions.
- Use **test data** for signup, login, and checkout.
- Capture **screenshots and logs** for every step.
- Generate **HTML reports** for each test run.

6. Test Environment

- **Browser:** Chrome (version matching ChromeDriver)
- **OS:** Windows 11
- **Tools:** Java Selenium, Maven, Eclipse/IDE, TestNG
- **Requirements:** Stable internet connection, test data

7. Roles & Responsibilities

- **Tester** Write and run automated scripts, check results, documentation
- **Project Lead:** Write and run automated scripts, combine automation results, and help prioritize issues

8. Deliverables

- Maven-based automation project
- TestNG suite file (testng.xml)
- Logs for all actions
- Screenshots for every step
- HTML test report
- Manual Test Case Sheet
- SRS Document
- Bug Report
- RTM
- Test Summary Report

9. Schedule

Day	Stage	Goal and Deliverables
21/10/2025	Analysis & Planning	• Select website for testing. • Select workflows • SRS documentation • Create test plan
22/10/2025	Design	• Create traceability matrix • Design test cases (positive and negative) for the selected workflows
23/10/2025 24/10/2025	Implementation	• Java maven project setup • Automation coding
25/10/2025 26/05/2025	Implementation	• Continue automation coding • Generate TestNG report

10. Risks & Constraints

- Internet speed may affect automated tests.
- Google Ads may overlap elements if not removed.
- Chrome version must match Chrome Driver.
- Wrong or incomplete test data may cause failures.

Resource Allocation:

- **Software resource**
    - Google Sheets.
    - **Eclipse IDE:** fully set up with all required dependencies (TestNG, Selenium)
    - ExtentManager
- **Hardware resource**
    - Laptop