# Test Plan and Test Strategy

## 1. Introduction

This document outlines the test plan and strategy for the automation testing of the e-commerce website using Playwright. The tests will ensure the functionality of various features such as login, address info, contact us, account creation, navigation, product search, and product-related actions.

## 2. Scope

The scope of this test plan includes:  
  
1. User login functionality.  
2. Address information entry.  
3. Contact Us form submission.  
4. Account creation.  
5. Navigation verification.  
6. Product search and interaction.  
7. Product sharing features.

## 3. Objectives

The objectives of the tests are to:  
  
1. Validate user login with valid credentials.  
2. Verify the ability to enter and save address information.  
3. Ensure the Contact Us form works correctly and sends messages.  
4. Confirm that users can create accounts using provided data.  
5. Check the navigation to the home page and verify its title.  
6. Test product search functionality and interaction with search results.  
7. Validate sending product information to a friend.

## 4. Test Environment

The tests will be conducted in the following environments:  
  
1. Browsers: Chrome, Firefox, and Safari.  
2. Devices: Web Application  
3. Test Data: Stored in testData.json.

## 5. Tools

The following tools will be used for the automation testing:  
  
1. Automation Framework: Playwright.  
2. Languages: JavaScript  
3. Page Object Model: Used for better maintainability and readability.

## 6. Test Data

Test data will be stored in a JSON file (testData.json). This includes user credentials, address information, contact messages, and other necessary details for the tests.

## 7. Test Strategy

### 7.1 Types of Testing

The following types of testing will be conducted:  
  
1. Functional Testing: Verify that each function of the application operates in conformance with the requirement specification.  
2. Regression Testing: Ensure that new code changes do not adversely affect the existing functionality.  
3. Integration Testing: Test combined parts of the application to ensure they work together as expected.  
4. UI Testing: Verify the user interface elements and their interactions.

### 7.2 Test Design

Tests will be designed following the Page Object Model (POM) to enhance readability and maintainability.