**TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM**

**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**Vũ Tuấn Anh – 520V0016**

**Thái Gia Bảo – 52000014**

**Trần Sỹ Thanh Trâm-52100853**

**TIỂU LUẬN GIỮA KỲ MÔN**

**KIỂM THỬ PHẦN MỀM TỰ ĐỘNG**

**Test Report Kiểm Thử Trang Web**

**Bán Hàng Swag Labs**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2024**

**TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM**

**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**Vũ Tuấn Anh – 520V0016**

**Thái Gia Bảo – 52000014**

**Trần Sỹ Thanh Trâm-52100853**

**TIỂU LUẬN GIỮA KỲ MÔN**

**KIỂM THỬ PHẦN MỀM TỰ ĐỘNG**

**Test Report Kiểm Thử Trang Web**

**Bán Hàng Swag Labs**

**Người hướng dẫn**

**ThS. Nguyễn Thanh Quân**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2024**

# INDEX

[**INDEX 3**](#_ydzvtjh5puh)

[**1-OVERALL 4**](#_feqkbfmgwanr)

[1.1-Timeline 4](#_3pfeifoczvze)

[1.2-Staff 4](#_gooq3x44z2v9)

[1.3-Goals 4](#_cd8vksgwp896)

[1.4-Introduction 5](#_mw5s6xy1v50)

[**2-SPECIFICATION 6**](#_4qr16ty5dqwk)

[2.1-Testing scope 6](#_fqdogoivfgfz)

[2.2-Test cases 7](#_k32ljdd17og0)

[2.3-Results and defects found 27](#_z120ne2pzh14)

# 1-OVERALL

## 1.1-Timeline

| STT | Thời Gian | Công Việc-Người Thực Hiện | Tiến Độ |
| --- | --- | --- | --- |
| 1 | 04/03-10/03 | Thiết kế test plan-Thanh Trâm | 100% |
| 2 | 11/03-17/03 | Viết test design-Thanh Trâm | 100% |
| 3 | 18/03-24/03 | Viết test case cho usecase 1/5/8- Gia Bảo | 100% |
| Viết test case cho usecase 2- Thanh Trâm | 100% |
| Viết test case cho usecase 3/4 - Tuấn Anh | 100% |
| 4 | 25/03-31/03 | Viết test case cho usecase 5/6 - Gia Bảo | 100% |
| Viết test case cho usecase 5/6 - Tuấn Anh | 100% |
| 5 | 01/04-07/04 | Tổng Hợp Các Modules | 100% |
| 6 | 08/04-18/05 | Viết Báo Cáo Test Result, Test Report, Defect Report | 100% |

## 

## 1.2-Staff

Nhóm 6

1. Vũ Tuấn Anh - 520V0016
2. Thái Gia Bảo - 52000014
3. Trần Sỹ Thanh Trâm-52100853

## 1.3-Goals

Chức năng Đăng Nhập & Đăng Xuất (Login Functionality):

* Đảm bảo rằng người dùng có thể đăng nhập vào trang web bằng các thông tin đăng nhập hợp lệ.
* Xác nhận rằng quá trình đăng nhập thành công chuyển hướng người dùng đến trang sản phẩm.
* Kiểm tra xem thông báo lỗi được hiển thị đúng khi người dùng nhập sai thông tin đăng nhập.

Chức năng Mua Hàng & Thanh Toán (Shopping Functionality):

* Xác định xem người dùng có thể thêm sản phẩm vào giỏ hàng, thay đổi số lượng hoặc loại bỏ sản phẩm khỏi giỏ hàng không.
* Kiểm tra tính chính xác của giỏ hàng, bao gồm số lượng sản phẩm và tổng số tiền.
* Đảm bảo rằng quá trình thanh toán và hoàn tất đơn hàng diễn ra đúng cách.

Tính Nhất Quán và Hiệu Suất:

* Kiểm tra tính nhất quán của giao diện người dùng trên các trình duyệt phổ biến: Chrome, Firefox
* Đảm bảo rằng trang web có thể xử lý các tải trọng đồng thời từ nhiều người dùng mà vẫn duy trì hiệu suất ổn định.

.

## 1.4-Introduction

## 

**Saucedemo.com cho phép người dùng truy cập bằng username và password. Thực hiện các thao tác mua sắm sản phẩm và thanh toán. Dưới đây là tổng quát hệ thống:**

Người dùng đăng nhập vào hệ thống bằng username và password. Yêu cầu username và password không được để trống. Nhấn button “Log In” để đăng nhập

vào hệ thống. Sau khi đăng nhập, người dùng chọn một sản phẩm trong mục “All Items”. Thông tin sản phẩm bao gồm: tên sản phẩm, hình ảnh, mô tả, giá tiền. Khách nhấn vào tên sản phẩm hoặc hình ảnh sản phẩm để xem thông tin sản phẩm. Sau khi xem sản phẩm, khách hàng nhấn button “Add to cart” để thêm sản phẩm vào giỏ hàng. Hoặc nhấn button “Remove” để xóa sản phẩm ra khỏi giỏ hàng. Sau khi nhấn button “Add to cart”, button đổi tên thành “Remove” để khách hàng có thể xóa sản phẩm.

Khách hàng có thể chọn chức năng sắp xếp sản phẩm bao gồm có: Name(A to Z), Name(Z to A), Price(Low to High), Price(High to Low).

* Name(A to Z): sắp xếp sản phẩm theo theo thứ tự tên từ A đến Z.
* Name(Z to A): sắp xếp sản phẩm theo thứ tự tên từ Z đến A.
* Price(Low to High): Sắp xếp sản phẩm theo giá từ thấp đến cao.
* Price(High to Low): Sắp xếp sản phẩm theo giá từ cao đến thấp.

Sau khi thêm sản phẩm vào giỏ hàng, số lượng sản phẩm được thêm vào hiển thị trên button giỏ hàng của trang web. Khách hàng nhấn vào button để thực hiện thanh toán. Ở trang thanh toán bao gồm thông tin sản phẩm và số lượng sản phẩm trong giỏ hàng, Khách hàng có thể nhấn vào tên sản phẩm để xem xem thông tin sản phẩm hoặc nhấn button “Remove” để xóa sản phẩm khỏi giỏ hàng.

Khách hàng nhấn vào button “Continue shopping” để quay lại trang mua sắm hoặc nhấn button “Checkout” để thực hiện thanh toán. Khách hàng điền thông tin thanh toán bao gồm: Firstname, Lastname, Zip/Postal Code. Yêu cầu Firstname, Lastname, Zip/Postal Code không được để trống. Khách hàng tiếp tự nhấn button “Continue” để hoàn tất thanh toán. Thông tin hóa đơn bao gồm: Payment Information, Shipping Information, Price Total, Tax, Total. Khách hàng nhấn button “Finish” để hoán tất thành toán hoặc “Cancel” để hủy bỏ quá trình thanh toán.

Sau khi xem và mua hàng, khách hàng muốn đăng xuất chọn mục “Log out” để đăng xuất khỏi trang web.

Các use case của hệ thống bao gồm: đăng nhập, đăng xuất, xem sản phẩm, thanh toán.

# 

# 2-SPECIFICATION

## 2.1-Testing scope

**Test Type:** GUI Test, Function Test, Load & Performance Test

**Features:** Bảng Use Case

**Compatibility:** Chrome & FireFox

**Correctness Verification:**

* Người dùng có thể đăng nhập thành công bằng thông tin đăng nhập hợp lệ hay không.
* người dùng có thể xem thông tin sản phẩm một cách chính xác, bao gồm hình ảnh, mô tả và giá cả
* sản phẩm có được thêm vào giỏ hàng một cách chính xác và cập nhật số lượng sản phẩm trong giỏ hàng đúng cách
* sản phẩm có được xóa khỏi giỏ hàng một cách chính xác và cập nhật số lượng sản phẩm trong giỏ hàng đúng cách
* sắp xếp hoạt động đúng cách theo các tiêu chí.
* thanh toán đơn hàng đầy đủ tổng giá trị đơn hàng và xác nhận địa chỉ giao hàng
* Người dùng hủy đơn hàng thanh toán
* Người dùng có thể đăng xuất khỏi hệ hệ thống.

**Performance**

* Thời gian đăng nhập vào hệ thống
* Lượng người đăng nhập đồng thời.
* Thời gian để thêm sản phẩm vào giỏ hàng.
* Hiệu suất khi có nhiều người dùng cùng thêm sản phẩm vào giỏ hàng cùng một lúc.
* Hệ thống xử lý đúng khi có nhiều người cùng thực hiện thao tác xóa.
* Thời gian phản hồi sau khi người dùng thực hiện sắp xếp.
* Thời gian hoàn thành thanh toán.
* Nhiều người dùng đăng xuất hệ thống cùng một lúc.

## 2.2-Test cases

| **Use Case ID** | **Use Case Name** | **Number Of Test Cases** |
| --- | --- | --- |
| 1 | Đăng nhập | 7 |
| 2 | Xem thông tin sản phẩm | 1 |
| 3 | Thêm sản phẩm | 9 |
| 4 | Xóa sản phẩm | 4 |
| 5 | Sắp xếp | 3 |
| 6 | Checkout(Thanh toán). | 2 |
| 7 | Cancel(Hủy thanh toán) | 7 |
| 8 | Đăng xuất | 1 |
| 9 | Go Back | 1 |

**Test Case 1:** **Module1\_1**

**Documentation:** Check login with correct username and password.

**Step by step:**

1. Open the browser and navigate to ${URL}.
2. Wait until the page contains "Swag Labs".
3. Enter the correct username "${correct\_username}".
4. Enter the correct password "${correct\_password}".
5. Click on the login button.
6. Verify that the page does not contain "Epic sadface: Username and password do not match any user in this service".
7. Close the browser.

**Expected results:** Users should be successfully logged in and directed to the dashboard/homepage.

**Test Data:**

○ Valid username: **standard\_user**

○ Valid password: **secret\_sauce**

**Result:** Login successfully

**Test Case 2:** **Module1\_2**

**Documentation:** Check login with correct username and password.

**Step by step:**

1. Open the browser and navigate to ${URL}.
2. Wait until the page contains "Swag Labs".
3. Enter the correct username "${correct\_username}".
4. Enter an incorrect password "${incorrect\_password}".
5. Click on the login button.
6. Verify that the page contains "Epic sadface: Username and password do not match any user in this service".
7. Close the browser.

**Expected results:** The login attempt should fail due to the incorrect password, and the page should display the message "Epic sadface: Username and password do not match any user in this service".

**Test Data:**

○ Valid username: **standard\_user**

○ Valid password: **incorrect\_password**

**Result:** Login attempt failed as expected due to the incorrect password.

**Test Cases 3:** **Module1\_3**

**Documentation:** Check login with incorrect username & correct password.

**Step by step:**

1. Open the browser and navigate to ${URL}.
2. Wait until the page contains "Swag Labs".
3. Enter an incorrect username "${incorrect\_username}".
4. Enter the correct password "${correct\_password}".
5. Click on the login button.
6. Verify that the page contains "Epic sadface: Username and password do not match any user in this service".
7. Close the browser.

**Expected results:** The login attempt should fail due to the incorrect username, and the page should display the message "Epic sadface: Username and password do not match any user in this service".

**Test Data:**

○ Valid username: **incorrect\_username**

○ Valid password: **secret\_sauce**

**Result:** The login attempt failed as expected due to the incorrect username.

**Test Cases 4:** **Module1\_4**

**Documentation:** Check login and leave the password blank.

**Step by step:**

1. Open the browser and navigate to ${URL}.
2. Wait until the page contains "Swag Labs".
3. Enter the correct username "${correct\_username}".
4. Click on the login button without entering the password.
5. Verify that the page contains "Epic sadface: Password is required".
6. Close the browser.

**Expected results:** The login attempt should fail as the password field is left blank, and the page should display the message "Epic sadface: Password is required".

**Test Data:**

○ Valid username: **standard\_user**

○ Valid password:

**Result:**  The login attempt failed as expected due to the empty password field.

**Test Cases 5:** **Module1\_5**

**Documentation:** Check login and leave the username blank.

**Step by step:**

1. Open the browser and navigate to ${URL}.
2. Wait until the page contains "Swag Labs".
3. Enter the correct password "${correct\_password}".
4. Click on the login button without entering the username.
5. Verify that the page contains "Epic sadface: Username is required".
6. Close the browser.

**Expected results:** The login attempt should fail as the username field is left blank, and the page should display the message "Epic sadface: Username is required".

**Test Data:**

○ Valid username:

○ Valid password: **secret\_sauce**

**Result:** The login attempt failed as expected due to the empty username field.

**[End of Test Case]**

**Test Cases 6:** **Module1\_6**

**Documentation:** Check login with blank username and password.

**Step by step:**

1. Open the browser and navigate to ${URL}.
2. Wait until the page contains "Swag Labs".
3. Click on the login button without entering the username and password.
4. Verify that the page contains "Epic sadface: Username is required".
5. Close the browser.

**Expected results:** The login attempt should fail as both the username and password fields are left blank, and the page should display the message "Epic sadface: Username is required".

**Test Data:**

○ Valid username:

○ Valid password:

**Result:** The login attempt failed as expected due to both the username and password fields being left blank.

**Test Case 7:** **Module1\_7**

**Documentation:** Check login with locked username.

**Step by step:**

1. Open the browser and navigate to ${URL}.
2. Wait until the page contains "Swag Labs".
3. Enter the locked username "${locked\_out\_username}".
4. Enter the correct password "${correct\_password}".
5. Click on the login button.
6. Verify that the page contains "Epic sadface: Sorry, this user has been locked out".
7. Close the browser.

**Expected results:** The login attempt should fail as the username is locked, and the page should display the message "Epic sadface: Sorry, this user has been locked out".

**Test Data:**

○ Valid username: **locked\_out\_user**

○ Valid password: **secret\_sauce**

**Result:** The login attempt failed as expected due to the locked username.

**Test Case : Module2\_1**

**Documentation**: Verify that product information is displayed correctly on the inventory page.

**Step by Step:**

1. Open the browser and navigate to ${URL}.
2. Enter the correct username ${correct\_username}.
3. Enter the correct password ${correct\_password}.
4. Click on the login button.
5. Wait until the page contains the inventory list ${inventory}.
6. Verify that the product “[Sauce Labs Backpack](https://www.saucedemo.com/inventory.html#)” is present on the inventory page.
7. Close the browser.

**Expected Result:** The product information should be correctly displayed on the inventory page, with the product “[Sauce Labs Backpack](https://www.saucedemo.com/inventory.html#)” visible.

**Test Data:**

* Username: standard\_user
* Password: secret\_sauce
* Product: [Sauce Labs Backpack](https://www.saucedemo.com/inventory.html#)

**Result:** The product information was correctly displayed on the inventory page, with the product “[Sauce Labs Backpack](https://www.saucedemo.com/inventory.html#)” visible.

**Test Case : Module2\_2**

**Documentation:** Verify that the product image is displayed correctly on the inventory page of the SauceDemo website.

**Step by Step:**

1. Open the browser and navigate to ${URL}.
2. Enter the correct username ${correct\_username}.
3. Enter the correct password ${correct\_password}.
4. Click on the login button.
5. Wait until the page contains the inventory list ${inventory}.
6. Verify that the product image “[Sauce Labs Backpack](https://www.saucedemo.com/inventory.html#)” is displayed on the inventory page.
7. Close the browser.

**Expected Result:** The product image should be correctly displayed on the inventory page.

**Test Data:**

* Username: standard\_user
* Password: secret\_sauce
* Image: [Sauce Labs Backpack](https://www.saucedemo.com/inventory.html#)

**Result:** The product image was correctly displayed on the inventory page.

**Test Case : Module2\_3**

**Documentation:** Verify that the product name is displayed correctly on the inventory page.

**Step by Step:**

1. Open the browser and navigate to ${URL}.
2. Enter the correct username ${correct\_username}.
3. Enter the correct password ${correct\_password}.
4. Click on the login button.
5. Wait until the page contains the inventory list ${inventory}.
6. Verify that the product name “[Sauce Labs Backpack](https://www.saucedemo.com/inventory.html#)” is displayed on the inventory page.
7. Close the browser.

**Expected Result:** The product name should be correctly displayed on the inventory page.

**Test Data:**

* Username: standard\_user
* Password: secret\_sauce
* Name: “[Sauce Labs Backpack](https://www.saucedemo.com/inventory.html#)”

**Result:** The product name was correctly displayed on the inventory page.

**Test Case : Module2\_4**

**Documentation:** Verify that the product details are displayed correctly on the inventory page of the SauceDemo website.

**Step by Step:**

1. Open the browser and navigate to ${URL}.
2. Enter the correct username ${correct\_username}.
3. Enter the correct password ${correct\_password}.
4. Click on the login button.
5. Wait until the page contains the inventory list ${inventory}.
6. Verify that the product details “[Sauce Labs Backpack](https://www.saucedemo.com/inventory.html#)” are displayed on the inventory page.
7. Close the browser.

**Expected Result:** The product details should be correctly displayed on the inventory page.

**Test Data:**

* Username: standard\_user
* Password: secret\_sauce
* Product: “[Sauce Labs Backpack](https://www.saucedemo.com/inventory.html#)”

**Result:** The product details “[Sauce Labs Backpack](https://www.saucedemo.com/inventory.html#)” were correctly displayed on the inventory page.

**Test Case : Module4\_5**

**Documentation:** Verify that the product price is displayed correctly on the inventory page of the SauceDemo website.

**Step by Step:**

1. Open the browser and navigate to ${URL}.
2. Enter the correct username ${correct\_username}.
3. Enter the correct password ${correct\_password}.
4. Click on the login button.
5. Wait until the page contains the inventory list ${inventory}.
6. Verify that the product price “[Sauce Labs Backpack](https://www.saucedemo.com/inventory.html#)” is displayed on the inventory page.
7. Close the browser.

**Expected Result:** The product price should be correctly displayed on the inventory page.

**Test Data:**

* Username: standard\_user
* Password: secret\_sauce
* Product: “[Sauce Labs Backpack](https://www.saucedemo.com/inventory.html#)”

**Result:** The product price was correctly displayed on the inventory page.

**Test Case : Module2\_6**

**Documentation:** Verify that the product image is displayed correctly after clicking the product on the inventory page .

**Step by Step:**

1. Open the browser and navigate to ${URL}.
2. Enter the correct username ${correct\_username}.
3. Enter the correct password ${correct\_password}.
4. Click on the login button.
5. Wait until the page contains the inventory list ${inventory}.
6. Click on the product ${product4}.
7. Verify that the product image “[Sauce Labs Backpack](https://www.saucedemo.com/inventory.html#)” is displayed.
8. Close the browser.

**Expected Result:** The product image should be correctly displayed after clicking the product.

**Test Data:**

* Username: standard\_user
* Password: secret\_sauce
* Product: “[Sauce Labs Backpack](https://www.saucedemo.com/inventory.html#)”

**Result:** The product image was correctly displayed after clicking the product.

**Test Case : Module2\_7**

**Documentation:** Verify that the product name is displayed correctly after clicking the product on the inventory page.

**Step by Step:**

1. Open the browser and navigate to ${URL}.
2. Enter the correct username ${correct\_username}.
3. Enter the correct password ${correct\_password}.
4. Click on the login button.
5. Wait until the page contains the inventory list ${inventory}.
6. Click on the product ${product4}.
7. Verify that the product name “[Sauce Labs Backpack](https://www.saucedemo.com/inventory.html#)” is displayed.
8. Close the browser.

**Expected Result:** The product name should be correctly displayed after clicking the product.

**Test Data:**

* Username: standard\_user
* Password: secret\_sauce
* Product: “[Sauce Labs Backpack](https://www.saucedemo.com/inventory.html#)”

**Result:** The product name was correctly displayed after clicking the product.

**Test Case : Module2\_8**

**Documentation:** Verify that the product details are displayed correctly after clicking the product on the inventory page of the SauceDemo website.

**Step by Step:**

1. Open the browser and navigate to ${URL}.
2. Enter the correct username ${correct\_username}.
3. Enter the correct password ${correct\_password}.
4. Click on the login button.
5. Wait until the page contains the inventory list ${inventory}.
6. Click on the product ${product4}.
7. Verify that the product details “[Sauce Labs Backpack](https://www.saucedemo.com/inventory.html#)” are displayed.
8. Close the browser.

**Expected Result:** The product details should be correctly displayed after clicking the product.

**Test Data:**

* Username: standard\_user
* Password: secret\_sauce
* Product: “[Sauce Labs Backpack](https://www.saucedemo.com/inventory.html#)”

**Result:** The product details were correctly displayed after clicking the product.

**Test Case : Module2\_9**

**Documentation:** Verify that the product price is displayed correctly after clicking the product on the inventory page of the SauceDemo website.

**Step by Step:**

1. Open the browser and navigate to ${URL}.
2. Enter the correct username ${correct\_username}.
3. Enter the correct password ${correct\_password}.
4. Click on the login button.
5. Wait until the page contains the inventory list ${inventory}.
6. Click on the product ${product4}.
7. Verify that the product price “[Sauce Labs Backpack](https://www.saucedemo.com/inventory.html#)” is displayed.
8. Close the browser.

**Expected Result:** The product price should be correctly displayed after clicking the product.

**Test Data:**

* Username: standard\_user
* Password: secret\_sauce
* Product: “[Sauce Labs Backpack](https://www.saucedemo.com/inventory.html#)”

**Result:** The product price was correctly displayed after clicking the product.

**TC:** **Module3\_1**

**Documentation**: Sort products from A to Z.

**Step by step:**

1. Open the browser and navigate to ${URL}.
2. Enter the correct username "${Correct\_Username}".
3. Enter the correct password "${Correct\_Password}".
4. Click on the login button.
5. Wait until the page contains the product list.
6. Get the web elements of product names "${ProductNames}".
7. Create a list of product names.
8. Log the product names.
9. Close the browser.

**Expected results:** The products should be sorted alphabetically from A to Z, and the list of product names should be logged.

**Test Data:**

○ Valid username: **standard\_user**

○ Valid password: **secret\_sauce**

**Result:** The products were sorted alphabetically from A to Z, and the list of product names was successfully logged.

**TC9:** **Module3\_2**

**Documentation:** Sort products from Z to A.

**Step by step:**

1. Open the browser and navigate to ${URL}.
2. Enter the correct username "${Correct\_Username}".
3. Enter the correct password "${Correct\_Password}".
4. Click on the login button.
5. Wait until the page contains the product list.
6. Get the web elements of product names "${ProductNames}".
7. Create a list of product names.
8. Sort the product names from Z to A.
9. Log the sorted product names.
10. Close the browser.

**Expected results:** The products should be sorted alphabetically from Z to A, and the list of product names should be logged.

**Test Data:**

○ Valid username: **standard\_user**

○ Valid password: **secret\_sauce**

**Result:** The products were sorted alphabetically from Z to A, and the list of product names was successfully logged.

**Test Cases 10 (TC10):** **Module3\_3**

**Documentation:** Sort products from high to low.

**Step by step:**

1. Open the browser and navigate to ${URL}.
2. Enter the correct username "${correct\_username}".
3. Enter the correct password "${correct\_password}".
4. Click on the login button.
5. Wait until the page contains the product list.
6. Click on the sort dropdown.
7. Select the option to sort products from high to low.
8. Wait until the page contains the sorted product list.
9. Get the web elements of product prices "${ProductPrice}".
10. Create a list of product prices.
11. Sort the product prices from high to low.
12. Verify that the sorted prices match the original prices.
13. Close the browser.

**Expected results:** The products should be sorted from high to low based on their prices, and the sorted prices should match the original prices.

**Test Data:**

○ Valid username: **standard\_user**

○ Valid password: **secret\_sauce**

**Result:** The products were successfully sorted from high to low based on their prices, and the sorted prices matched the original prices.

**Test Cases 11:** **Module3\_4**

**Documentation:** Sort products from low to high on SauceDemo website.

**Step by step:**

1. Open the browser and navigate to ${URL}.
2. Enter the correct username "${correct\_username}".
3. Enter the correct password "${correct\_password}".
4. Click on the login button.
5. Wait until the page contains the product list.
6. Click on the sort dropdown.
7. Select the option to sort products from low to high.
8. Wait until the page contains the sorted product list.
9. Get the web elements of product prices "${ProductPrice}".
10. Create a list of product prices.
11. Sort the product prices from low to high.
12. Verify that the sorted prices match the original prices.
13. Close the browser.

**Expected results:**  The products should be sorted from low to high based on their prices, and the sorted prices should match the original prices.

**Test Data:**

○ Valid username: **standard\_user**

○ Valid password: **secret\_sauce**

**Result:** The products were successfully sorted from low to high based on their prices, and the sorted prices matched the original prices.

**Test Case : Module4\_1**

**Documentation:** Verify the number of products in the cart.

**Step by step:**

* Open the browser and navigate to ${URL}.
* Wait until the page contains "Swag Labs".
* Login to the website.
* Get the number of products in the cart.
* Verify that the number of products in the cart is equal to 6.
* Close the browser.

**Expected result:** The number of products in the cart should be 6

**Test Data:**

* Valid username: standard\_user
* Valid password: secret\_sauce

**Result:** The number of products in the cart was equal to 6.

**Test Case : Module4\_2**

**Documentation:** Add products to the cart and verify quantity in the cart.

**Step by step**:

* Open the browser and navigate to ${URL}.
* Wait until the page contains "Swag Labs".
* Login to the website.
* Loop through each product.
* Click on the product to add it to the cart.
* Verify that the quantity in the cart increases after adding each product.
* Close the browser.

**Expected result:** The quantity in the cart should increase after adding each product.

**Test Data:**

* Valid username: standard\_user
* Valid password: secret\_sauce

**Result:** The quantity in the cart increased after adding each product.

**Test Case : Module4\_3**

**Documentation:** Verify removing products from the cart.

**Step by step:**

* Open the browser and navigate to ${URL}.
* Wait until the page contains "Swag Labs".
* Login to the website.
* Click on "Add to Cart" buttons.
* Get the cart count before removing products.
* Remove products from the cart.
* Verify that the cart count becomes zero after removing products.
* Close the browser.

**Expected result:** The cart count should become zero after removing products from the cart.

Test Data:

* Valid username: standard\_user
* Valid password: secret\_sauce

**Result:** The cart count became 0 after removing products from the cart.

Test Case: Module4\_5

**Documentation:** Visit each product page.

**Step by step:**

* Open the browser and navigate to ${URL}.
* Wait until the page contains "Swag Labs".
* Maximize the browser window.
* Login to the website.
* Loop through each product.
* Click on each product to visit its page.
* Close the browser.

**Expected result:** Each product page should be visited successfully.

**Test Data:**

* Valid username: standard\_user
* Valid password: secret\_sauce

**Result:** Each product page was visited successfully.

**Test Case: Module4\_6**

**Documentation:** Proceed to checkout.

**Step by step:**

* Open the browser and navigate to ${URL}.
* Wait until the page contains "Swag Labs".
* Maximize the browser window.
* Login to the website.
* Click on the shopping cart icon.
* Click on the checkout button.
* Close the browser.

**Expected result:** The user should be able to proceed to the checkout page successfully.

**Test Data:**

* Valid username: standard\_user
* Valid password: secret\_sauce

**Result:** The user was able to proceed to the checkout page successfully.

**Test Case Module4\_7**

**Documentation:** Fill out shipping information and proceed to payment.

**Step by step:**

* Open the browser and navigate to ${URL}.
* Wait until the page contains "Swag Labs".
* Maximize the browser window.
* Login to the website.
* Click on the shopping cart icon.
* Click on the checkout button.
* Enter the shipping information.
* Click on the continue button.
* Close the browser.

**Expected result:** The shipping information should be filled out successfully, and the user should be able to proceed to payment.

**Test Data:**

* First name: John
* Last name: Doe
* Postal code: 12345
* Valid username: standard\_user
* Valid password: secret\_sauce

**Result:** The shipping information was filled out successfully, and the user was able to proceed to payment.

**Test Case Module4\_8**

**Documentation:** Verify total amount and tax in the checkout summary.

**Step by step:**

* Open the browser and navigate to ${URL}.
* Wait until the page contains "Swag Labs".
* Maximize the browser window.
* Login to the website.
* Click on the shopping cart icon.
* Click on the checkout button.
* Verify the total products amount displayed in the checkout summary.
* Calculate the expected total products amount.
* Verify that the calculated total products amount matches the displayed amount.
* Verify the tax amount displayed in the checkout summary.
* Calculate the expected tax amount.
* Verify that the calculated tax amount matches the displayed tax amount.
* Verify the total amount due displayed in the checkout summary.
* Calculate the expected total amount due including tax.
* Verify that the calculated total amount due matches the displayed total amount due.
* Close the browser.

**Expected result:** The total products amount, tax amount, and total amount due should be displayed correctly in the checkout summary, and the calculated amounts should match the displayed amounts.

**Test Data:**

* Valid username: standard\_user
* Valid password: secret\_sauce

**Result:** The test passed as expected, as the total products amount, tax amount, and total amount due were displayed correctly in the checkout summary, and the calculated amounts matched the displayed amounts.

**Test Case: Module4\_9**

**Documentation:** Complete the checkout process.

**Step by step:**

* Click on the "Finish" button.
* Verify that the page contains the message "Thank you for your order!".

**Expected result:** The checkout process should be completed successfully, and the page should display the message "Thank you for your order!".

**Test Data:** None required.

**Result:** The checkout process was completed successfully, and the page displayed the message "Thank you for your order!".

**Test Case : Module4\_11**

**Documentation:** Verify error message when first name is not provided during checkout.

* Step by step:
* Open the browser and navigate to ${URL}.
* Maximize the browser window.
* Login to the website.
* Click on the shopping cart icon.
* Click on the checkout button.
* Click on the continue button without providing the first name.
* Verify that the page contains the error message "Error: First Name is required".

**Expected result:** The error message "Error: First Name is required" should be displayed when the first name is not provided during checkout.

**Test Data:**

* Valid username: standard\_user
* Valid password: secret\_sauce

**Result:** The error message "Error: First Name is required" was displayed when the first name was not provided during checkout.

**Test Case : Module4\_12**

**Documentation:** Verify error message when last name is not provided during checkout.

**Step by step:**

* Open the browser and navigate to ${URL}.
* Wait for 3 seconds.
* Maximize the browser window.
* Login to the website.
* Click on the shopping cart icon.
* Click on the checkout button.
* Enter "John" in the first name field.
* Enter "12345" in the postal code field.
* Click on the continue button.
* Verify that the page contains the error message "Error: Last Name is required".
* Close the browser.

**Expected result:** The error message "Error: Last Name is required" should be displayed when the last name is not provided during checkout.

**Test Data:**

* Valid username: standard\_user
* Valid password: secret\_sauce

**Result:** The error message "Error: Last Name is required" was displayed when the last name was not provided during checkout.

**Test Case : Module4\_13**

**Documentation:** Verify error message when postal code is not provided during checkout.

Step by step:

* Open the browser and navigate to ${URL}.
* Maximize the browser window.
* Login to the website.
* Click on the shopping cart icon.
* Click on the checkout button.
* Enter the first name and last name.
* Click on the continue button without providing the postal code.
* Verify that the page contains the error message "Error: Postal Code is required".

**Expected result:** The error message "Error: Postal Code is required" should be displayed when the postal code is not provided during checkout.

**Test Data:**

* First name: John
* Last name: Doe.
* Valid username: standard\_user
* Valid password: secret\_sauce

**Result:** The error message "Error: Postal Code is required" was displayed when the postal code was not provided during checkout.

## 2.3-Results and defects found

**Testing results:**

* Total number of TCs: 36
* Total number of defects found: 0
* Testing coverage: 100%