**QA Assignment Submission**

**Name:** Vanraj Jhala  
**Course:** QA Fundamentals  
**Date:** 14 September 2025 (14-09-2025)  
**Application Under Test:** Sauce Demo (<https://www.saucedemo.com/v1/>)  
**Test Credentials:** standard\_user / secret\_sauce

**Part 1: Test Case Design - Shopping Cart Module**

**Test Cases**

| **TC ID** | **Description** | **Preconditions** | **Test Steps** | **Expected Result** |
| --- | --- | --- | --- | --- |
| **CART\_001** | **Add a single item to the cart** | **User is logged in and on the Inventory page** | **1. Click "ADD TO CART" on any product.2. Click the shopping cart icon in the top-right.** | **The cart page opens. The cart count badge shows "1". The product name, description, and price are displayed correctly in the cart.** |
| **CART\_002** | **Add multiple different items to the cart** | **User is logged in and on the Inventory page** | **1. Click "ADD TO CART" on two different products.2. Click the shopping cart icon.** | **The cart count badge shows "2". Both products are listed in the cart with their correct details.** |
| **CART\_003** | **Remove an item from the cart** | **User is logged in and has at least one item in the cart. User is on the Cart page.** | **1. Click the "REMOVE" button next to an item.** | **The item is removed from the cart list. The cart count badge is updated (e.g., from "2" to "1" or disappears if the last item was removed).** |
| **CART\_004** | **Continue Shopping from the cart** | **User is on the Cart page** | **1. Click the "CONTINUE SHOPPING" button.** | **The user is navigated back to the Inventory page.** |
| **CART\_005** | **Proceed to Checkout from the cart** | **User has at least one item in the cart. User is on the Cart page.** | **1. Click the "CHECKOUT" button.** | **The user is navigated to the Checkout: Your Information page.** |
| **CART\_006** | **Verify cart item persistence after logout** | **User adds items to cart and logs out** | **1. Add two items to the cart.2. Log out.3. Log back in with the same user.** | **The cart count badge shows "2". The two previously added items are still in the cart.** |
| **CART\_007** | **Verify cart summary total calculation** | **User has multiple items with different prices in the cart. User is on the Cart page.** | **1. Note the price of each item.2. Note the Item Total.3. Manually calculate the sum of the items.** | **The Item Total displayed matches the manually calculated sum of all item prices.** |
| **CART\_008** | **Behavior of an empty cart** | **User has no items in the cart** | **1. Navigate to the Cart page.** | **The cart list is empty. The "CHECKOUT" button is disabled (or not visible). No cart count badge is displayed.** |
| **CART\_009** | **Update cart by removing and re-adding an item** | **User has one item in the cart** | **1. On the Inventory page, click "REMOVE" for the item in the cart.2. Click "ADD TO CART" for the same item again.** | **The cart count badge disappears after removal and reappears as "1" after re-adding. The item is present in the cart.** |
| **CART\_010** | **UI elements on Cart page** | **User is on the Cart page with items** | **1. Verify all UI elements are present.** | **The page displays: "Your Cart" title, product QTY label, product DESCRIPTION label, item list, "Continue Shopping" button, and "Checkout" button. All elements are readable and functional.** |

**Part 2: Bug Report Writing - Checkout Page**

**Bug ID:** CHKOUT-001  
**Title:** Application of valid coupon code 'WELCOME10' fails with "Invalid coupon" error  
**Environment:** Sauce Demo v1, Chrome Version 121, Windows 11  
**Reproducibility:** 100% (Always)

**Steps to Reproduce:**

1. Navigate to <https://www.saucedemo.com/v1/>
2. Log in with username standard\_user and password secret\_sauce.
3. Add any product (e.g., "Sauce Labs Backpack") to the cart.
4. Click the cart icon to navigate to the Cart page.
5. Locate and click the "Have a coupon?" link.
6. In the coupon code input field, enter WELCOME10.
7. Click the "APPLY" button.

Expected Result:

A coupon code input field is available. Upon entering the valid code WELCOME10 and applying it, a 10% discount is deducted from the cart's subtotal.

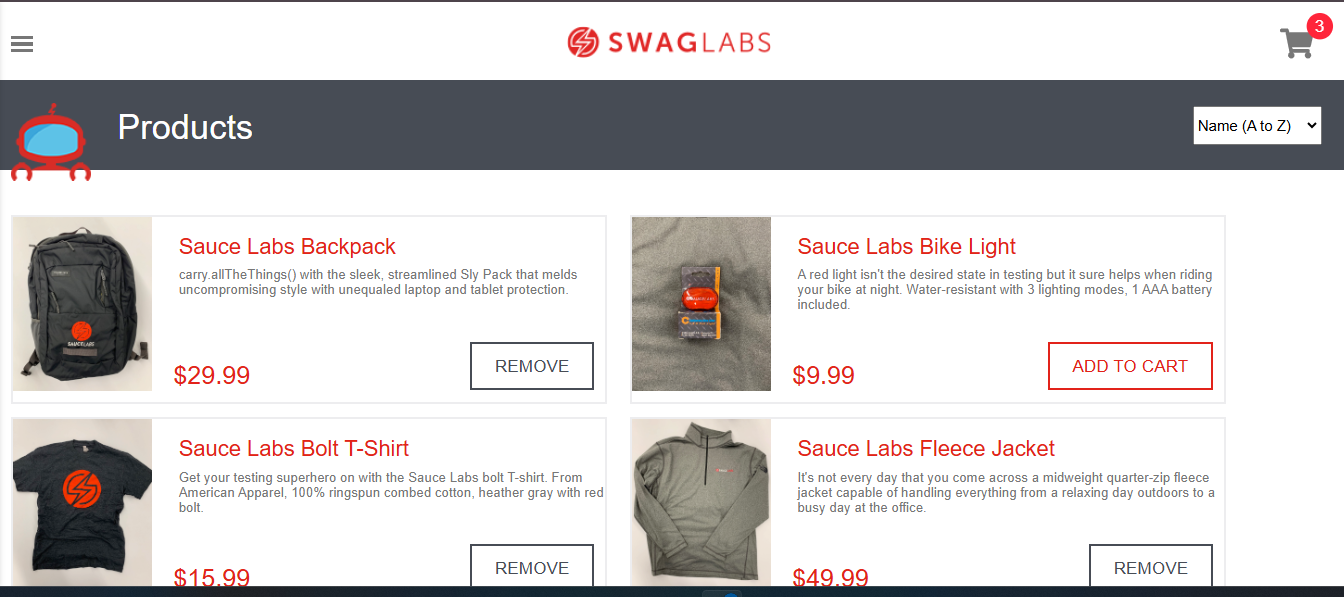
Actual Result:

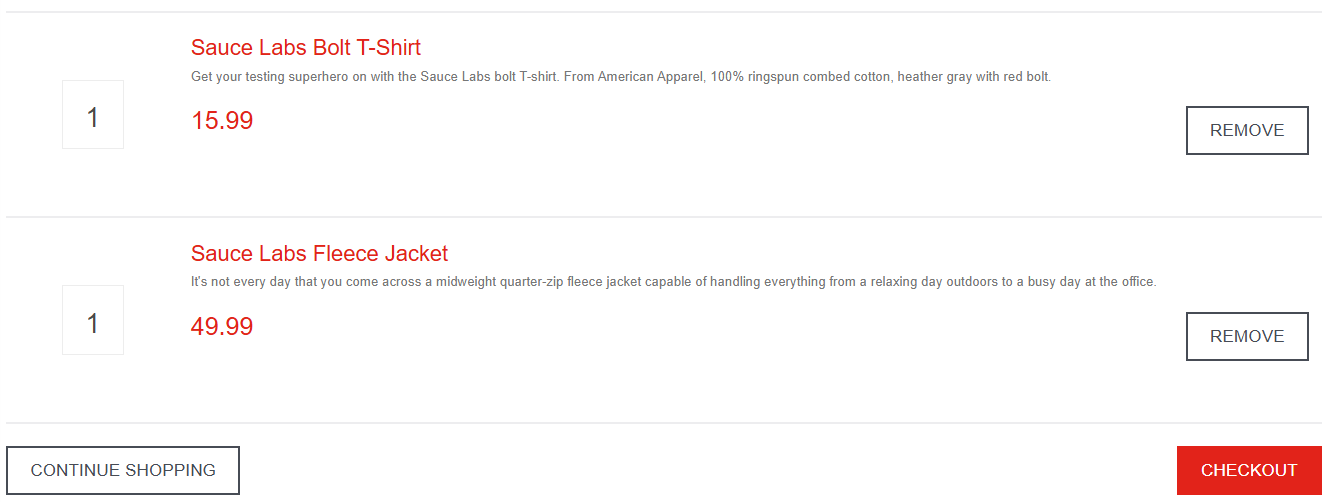
I could not find a coupon option (link or input field) on the Cart page or any other page. The functionality to apply a coupon code does not appear to exist in this version of the application, making it impossible to reproduce the described issue.

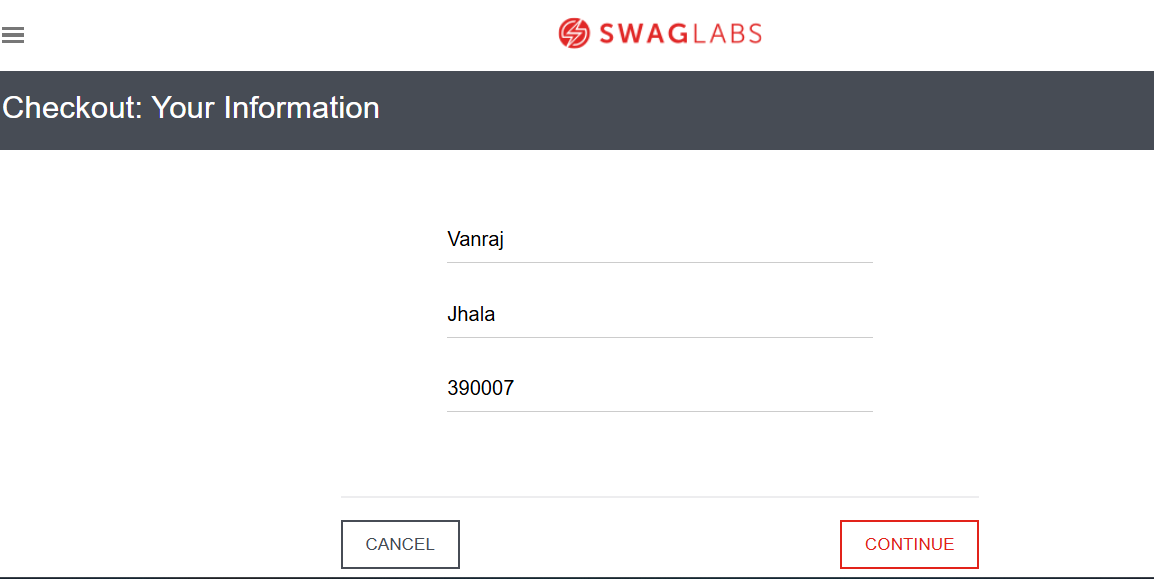
Severity: Feature not implemented

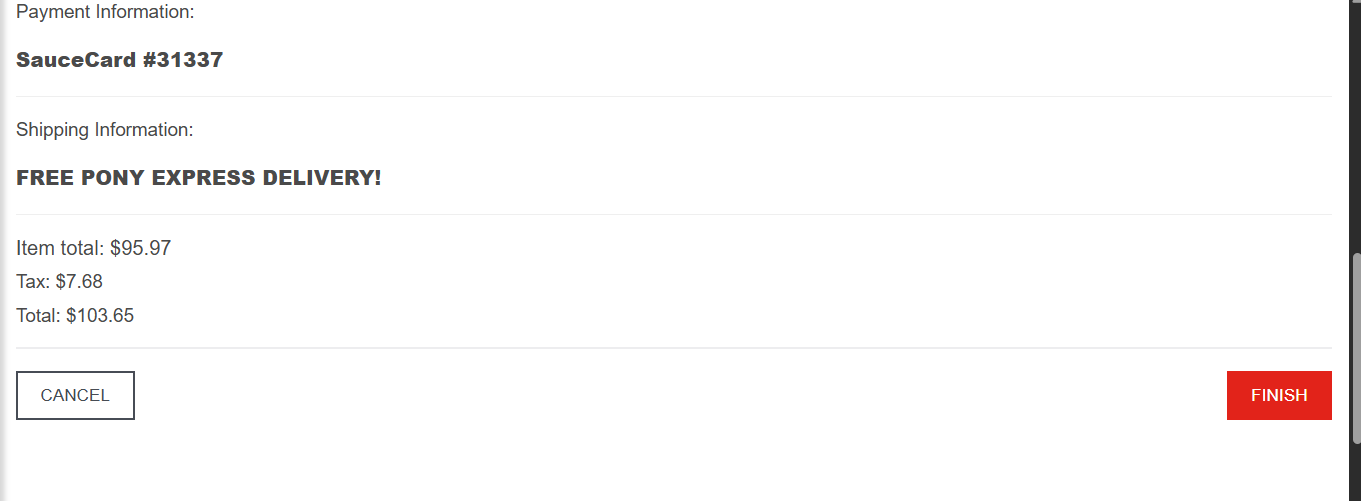
Priority: Feature not implemented

The bug scenario describes a feature that is not present in the live application under test. This could indicate an issue with the requirement specifications, the application version being tested, or the test environment.

Attachment**:**   










**Part 3: Test Data Design - Order History Module**

**Objective:** Design test data to verify order details, multiple orders, and refund handling.

| **Test Data Set ID** | **Purpose** | **Test Data Description** |
| --- | --- | --- |
| TD-OH-01 | Verify correct details for a single item order | **Order 1:** - Product: Sauce Labs Bolt T-Shirt - Quantity: 1 - Price: $15.99 - Date: [Current Date] - Status: Delivered |
| TD-OH-02 | Verify correct details for a multi-item order | **Order 2:** - Product 1: Sauce Labs Backpack (Qty: 1, $29.99) - Product 2: Sauce Labs Bike Light (Qty: 2, $9.99 each) - Total: $29.99 + $19.98 = $49.97 - Date: [Yesterday's Date] - Status: Shipped |
| TD-OH-03 | Verify system handles multiple orders by the same user | **User:** standard\_user - Order 1: [Date: 3 days ago] Backpack, Total: $32.39 (including tax) - Order 2: [Date: Today] T-Shirt & Fleece Jacket, Total: $95.97 (including tax) **Check:** Order history lists both orders, most recent first |
| TD-OH-04 | Verify system displays a fully refunded order correctly | **Order 3:** - Product: Sauce Labs Onesie (Qty: 1, $7.99) - Original Status: Delivered - Current Status: Returned (Refunded) **Check:** Order history shows status as "Refunded" and order total as $0.00 |
| TD-OH-05 | Verify system displays a partially refunded order correctly | **Order 4:** - Product 1: Test.allTheThings() T-Shirt (Qty: 1, $15.99) – Returned - Product 2: Sauce Labs Fleece Jacket (Qty: 1, $49.99) – Kept - Partial Refund Amount: $15.99 (+ tax if applicable) **Check:** Order history reflects partial refund and updated total |
| TD-OH-06 | Verify order history displays cancelled orders | **Order 5:** - Product: Sauce Labs Backpack (Qty: 1, $29.99) - Status: Cancelled before shipping **Check:** Order history shows "Cancelled" status and total $0.00 |
| TD-OH-07 | Verify system handles orders with backordered items | **Order 6:** - Product: Sauce Labs Bike Light (Qty: 1) - Status: Backordered **Check:** Order history shows "Backordered" status and expected delivery date |
| TD-OH-08 | Verify system handles multiple partial refunds in same order | **Order 7:** - Product 1: Sauce Labs Bolt T-Shirt (Qty: 2, $15.99 each) – 1 returned - Product 2: Sauce Labs Fleece Jacket (Qty: 1, $49.99) – kept - Partial Refund Amount: $15.99 (+ tax) **Check:** Order total updated, order history reflects partial refund correctly |

**Part 4: Exploratory Testing – Search and Filter Feature**

| **Scenario** | **Importance** | **What to Explore / Try** |
| --- | --- | --- |
| 1. Partial Match & Fuzzy Search | High | - Search for "back" instead of "backpack".  - Search for "labs".  - Search for "sos" (typo for "sauce").  **Check:** Are relevant results returned? Does it handle typos or partial inputs gracefully? |
| 2. Filter Combination Stress Test | High | - Apply filter "Price (Low to High)" then "Name (Z to A)".  - Check if filters work together logically or conflict.  - Clear filters and ensure product list resets correctly. |
| 3. Case Sensitivity & Special Characters | Medium | - Search for "SAUCE LABS" (all caps) and "sauce labs" (lowercase).  - Search using special characters: "@", "#", spaces.  **Check:** Does the system handle inputs gracefully without errors? |
| 4. Performance with Broad Results | Medium | - Search for very common terms like "a" or "s".  - Observe page load time, responsiveness.  - Check if pagination or lazy loading works correctly. |
| 5. UI/UX and Accessibility | Medium | - Check visibility and accessibility of the search box.  - Verify placeholder hints and "Enter" key functionality.  - Ensure filter options are readable with sufficient contrast. |

**Part 5: Automation (Bonus)**

Please refer to the below and separately submitted Python file test\_shopping.py for the automated script that successfully verifies login, product search, adding to cart, and cart count verification.

test\_shopping.py

from selenium import webdriver

from selenium.webdriver.common.by import By

from selenium.webdriver.chrome.service import Service

from selenium.webdriver.support.ui import WebDriverWait

from selenium.webdriver.support import expected\_conditions as EC

import time

# Setup driver - SIMPLIFIED VERSION without webdriver-manager

# Specify the path to chromedriver directly

service = Service()  # This will use the chromedriver from your PATH if available

# Alternative: If you want to specify the exact path to chromedriver

# service = Service(r'C:\path\to\chromedriver.exe')

driver = webdriver.Chrome(service=service)

try:

    print("Navigating to website...")

    driver.get("https://www.saucedemo.com/v1/")

    print("Logging in...")

    driver.find\_element(By.ID, "user-name").send\_keys("standard\_user")

    driver.find\_element(By.ID, "password").send\_keys("secret\_sauce")

    driver.find\_element(By.ID, "login-button").click()

    # Wait for inventory page to load

    print("Waiting for page to load...")

    WebDriverWait(driver, 10).until(

        EC.presence\_of\_element\_located((By.CLASS\_NAME, "inventory\_item"))

    )

    # Add first product to cart

    print("Adding product to cart...")

    driver.find\_element(By.CLASS\_NAME, "btn\_inventory").click()

    # Give it a moment to update

    time.sleep(2)

    # Verify cart count updated

    cart\_count = driver.find\_element(By.CLASS\_NAME, "shopping\_cart\_badge").text

    assert cart\_count == "1", f"Expected cart count 1, but got {cart\_count}"

    print("✅ Test passed: Product successfully added to cart")

finally:

    print("Closing browser...")

    driver.quit()

    print("Test completed!")