

Perion - Manual testing assignment

Submitted by: Moran Liv.
Phone: 054-4426668.
Email: moran.liv2006@gmail.com

Test Case ID	TC-001 (Negative scenario)
Description	Verify that the application prevents the user from proceeding with checkout when the shopping cart is empty, displaying an appropriate error message.
Preconditions	The user is logged in as standard_user and the cart is empty .
Steps	<ol style="list-style-type: none">1. Navigate to the Inventory Page (inventory.html).2. Click the Shopping Cart Icon (top right).3. On the Cart Page, click the "Checkout" button.4. Verify the displayed outcome.
Expected Results	The user should not be navigated to the "Checkout: Your Information" page (checkout-step-one.html). Instead, a specific, visible error message should appear on the current page, stating: "Error: Cart is empty" or similar.
Actual Results	
Pass/Fail	

Automation note: The verification is simple: asserting that the URL remains unchanged (navigation is blocked) and validating the specific error message text.

Test Case ID	TC-002
Description	Verify that a locked-out user is prevented from logging in and receives a clear, specific error message indicating the account status.
Preconditions	The user is on the Login Page (https://www.saucedemo.com/).
Steps	<ol style="list-style-type: none">1. Enter locked_out_user into the Username field.2. Enter secret_sauce into the Password field.3. Click the "Login" button.4. Verify the displayed outcome.
Expected Results	The user should not be redirected to the Inventory Page. An error message banner should appear on the login screen, specifically stating: "Epic sadface: Sorry, this user has been locked out."
Actual Results	
Pass/Fail	

Automation note: The automation executes simple form input and relies on a highly reliable assertion against the exact, expected error message displayed post-submission.

Test Case ID	TC-003
Description	Verify that selecting the " Price (low to high) " sort option correctly rearranges all products on the Inventory Page based on their monetary value, from the least expensive to the most expensive.
Preconditions	User is logged in as standard_user and is on the Inventory Page.
Steps	<ol style="list-style-type: none"> Click the sorting dropdown menu (usually displaying "Name (A to Z)" by default). Select the option "Price (low to high)". Manually observe the first two and the last two products in the list.
Expected Results	<p>The product list is immediately rearranged:</p> <ol style="list-style-type: none"> The first product displayed should have the lowest price (e.g., \$7.99). The last product displayed should have the highest price (e.g., \$49.99). All prices in between should strictly increase as you scroll down.
Actual Results	
Pass/Fail	

Automation note: Important scenario to automate, holds a crucial functionality. The test pulls the prices shown on the page, sorts them using the test logic, and then checks that they match exactly.

Test Case ID	TC-004
Description	Verify that the shopping cart badge (counter) accurately and instantaneously reflects the total number of unique items added to the cart, including immediate removal.
Preconditions	The user is logged in as standard_user , the cart is empty, and the cart badge displays no number .
Steps	<ol style="list-style-type: none"> Click "Add to Cart" for the "Sauce Labs Backpack." Click "Add to Cart" for the "Sauce Labs Bike Light." Verify the number displayed on the cart badge. Click the "Remove" button next to the "Sauce Labs Bike Light" product tile. Verify the number displayed on the cart badge.
Expected Results	<p>Steps 1 & 2: The cart badge should immediately display the number "2".</p> <p>Steps 4 & 5: The badge should immediately decrement and display the number "1".</p> <p>The state changes should be instant without any page refresh.</p>
Actual Results	
Pass/Fail	