**Functional Test Scenarios**

**1. Feature: User Authentication Module**

**Scenario:** Successful Login for Standard User

**Description:** This scenario verifies that a standard user can successfully log in to the application, as per US-01.

**Given** the user is on the login page at `https://www.saucedemo.com`

**When** the user enters "standard\_user" as the username

**And** the user enters "secret\_sauce" as the password

**And** the user clicks the "Login" button

**Then** the user should be successfully authenticated

**And** the user should be redirected to the inventory page (`/inventory.html`)

**And** the inventory page should display a list of products

**Notes/Assumptions:** This scenario directly covers the positive flow for `standard\_user` login.

**Test Data:** Username: `standard\_user`, Password: `secret\_sauce`

**Priority:** Critical

**Scenario:** Login Attempt with Locked Out User

**Description:** This scenario verifies that a locked-out user receives an appropriate error message upon login attempt, as per US-02.

**Given** the user is on the login page

**When** the user enters "locked\_out\_user" as the username

**And** the user enters "secret\_sauce" as the password

**And** the user clicks the "Login" button

**Then** the user should remain on the login page

**And** an error message should be displayed with the text: "Epic sadface: Sorry, this user has been locked out."

**Notes/Assumptions:** This scenario covers a negative test case for user authentication.

**Test Data:** Username: `locked\_out\_user`, Password: `secret\_sauce`

**Priority:** High

**Scenario:** Login Attempt with Problem User

**Description:** This scenario verifies the application's behavior when logging in with the `problem\_user` credential.

**Given** the user is on the login page

**When** the user enters "problem\_user" as the username

**And** the user enters "secret\_sauce" as the password

**And** the user clicks the "Login" button

**Then** the user should be successfully authenticated

**And** the user should be redirected to the inventory page (`/inventory.html`)

**Notes/Assumptions:** The Test Plan notes ambiguity regarding the expected behavior for `problem\_user`. This test will help clarify if the "problem" is functional or purely a UI rendering issue. The expected result assumes a successful login, with any issues to be documented on the inventory page.

**Test Data:** Username: `problem\_user`, Password: `secret\_sauce`

**Priority:** Medium

**2. Feature: Product Inventory and Browsing Page**

**Scenario:** Verify Product Display After Login

**Description:** This scenario ensures that after a successful login, the product catalog is correctly displayed on the inventory page, as per PRD Section US-03.

**Given** the user is successfully logged in as `standard\_user`

**When** the user is on the inventory page

**Then** a list of product items should be visible inside the `inventory\_container`

**And** each product should display its name, description, and price

**Notes/Assumptions:** Assumes a successful login as a precondition.

**Scenario:** Verify Product Sorting Functionality

**Description:** This scenario tests the sorting of products on the inventory page, verifying that products can be sorted by various criteria.

**Given** the user is on the inventory page with products displayed

**When** the user selects "Name (Z to A)" from the sort dropdown

**Then** the products should be re-ordered alphabetically from Z to A by their names

**When** the user selects "Price (low to high)" from the sort dropdown

**Then** the products should be re-ordered by price from lowest to highest

**Notes/Assumptions:** Assumes the sorting dropdown is present and functional. This functionality is inferred as a standard feature for an e-commerce inventory page.

**3. Feature: Shopping Cart Functionality**

**Scenario:** Add Item to Cart and Verify Badge Update

**Description:** This scenario verifies that adding an item to the shopping cart correctly updates the cart badge, as per US-04.

**Given** the user is logged in and on the inventory page

**And** the shopping cart is empty

**When** the user clicks "Add to cart" for the "Sauce Labs Backpack"

**Then** the shopping cart badge should appear and display '1'

**Notes/Assumptions:** This covers the initial "add to cart" functionality and badge update.

**Priority:** High

**Scenario:** Remove Item from Cart and Verify Badge Update

**Description:** This scenario verifies that removing an item from the shopping cart correctly updates the cart badge.

**Given** the user has added the "Sauce Labs Backpack" to the cart

**And** the shopping cart badge displays '1'

**When** the user clicks "Remove" for the "Sauce Labs Backpack"

**Then** the shopping cart badge should disappear

**And** the "Remove" button should change back to "Add to cart" for that item

**Notes/Assumptions:** This covers the "remove from cart" functionality and badge update.

**Priority:** High

**Scenario:** View Cart Page and Verify Items

**Description:** This scenario verifies that items added to the cart are correctly displayed on the cart page.

**Given** the user has added one or more items to the shopping cart

**When** the user clicks the shopping cart icon

**Then** the user should be redirected to the cart page (`/cart.html`)

**And** all added items should be listed with their name, description, and price

**Notes/Assumptions:** Assumes a working navigation to the cart page.

**4. Feature: Customer Checkout Process**

**Scenario:** Successful End-to-End Checkout

**Description:** This scenario verifies the complete successful multi-step checkout process from cart to order confirmation, as per US-05.

**Given** the user is logged in as `standard\_user`

**And** the "Sauce Labs Onesie" is in the shopping cart

**When** the user navigates to the cart page and clicks "Checkout"

**And** the user enters "Test" for "First Name", "User" for "Last Name", and "12345" for "Zip/Postal Code"

**And** the user clicks the "Continue" button

**And** on the "Checkout: Overview" page, the user clicks "Finish"

**Then** the user should be redirected to the "Checkout: Complete!" page (`/checkout-complete.html`)

**And** a success message "Thank you for your order!" should be displayed

**And** the shopping cart should be empty

**Notes/Assumptions:** This is a critical path positive test case.

**Test Data:** First Name: Test, Last Name: User, Zip: 12345

**Priority:** Critical

**Scenario:** Checkout with Missing Required Information

**Description:** This scenario verifies that the checkout process handles missing required information (First Name, Last Name, or Zip Code).

**Given** the user is on the "Checkout: Your Information" page

**When** the user clicks "Continue" without entering the "First Name"

**Then** an error message should appear indicating the First Name is required

**When** the user enters a "First Name" but clicks "Continue" without a "Last Name"

**Then** an error message should appear indicating the Last Name is required

**When** the user enters a "First Name" and "Last Name" but clicks "Continue" without a "Zip/Postal Code"

**Then** an error message should appear indicating the Zip/Postal Code is required

**Notes/Assumptions:** This covers negative test cases for the checkout form validation.

**5. Feature: Session Management**

**Scenario:** Logout Functionality

**Description:** This scenario verifies that a logged-in user can successfully log out of the application, as per US-06.

**Given** the user is logged in as `standard\_user`

**When** the user clicks the main menu icon

**And** the user clicks the "Logout" link

**Then** the user should be redirected to the login page

**And** the user's session should be terminated

**Notes/Assumptions:** Ensures proper session termination.

**Scenario:** Reset App State Functionality

**Description:** This scenario verifies the "Reset App State" functionality, ensuring it clears the shopping cart, as per US-06.

**Given** the user is logged in

**And** there are items in the shopping cart

**When** the user clicks the main menu icon

**And** the user clicks the "Reset App State" link

**Then** the shopping cart should be empty and the cart badge should disappear

**Notes/Assumptions:** This confirms the application's ability to return to a clean state without a full logout.