

Test Plan v1.0

SauceDemo Web Application

Üllar Kappak, QA Engineer

Objective

The goal of this plan is to check that SauceDemo works correctly for a typical shopping flow: login, browsing products, adding items to the cart, checkout and logout. Testing also covers basic usability, security and accessibility checks.

Scope

Inscope

- Login with different user types (valid, invalid, locked).
- Products page: product list, sorting, product detail.
- Cart: add, remove, update, persist after refresh.
- Checkout: personal info form, overview, order completion.
- Menu actions: All items, Reset App State, Logout.
- Footer links (Twitter, Facebook, LinkedIn).
- Non-functional checks, page responsiveness, error messages, simple performances and accessibility checks.

Out of scope

- Load/stress testing with large numbers of users.
- Payment gateway or backend integration (not included in demo site).
- Database/API level testing.
- Mobile app testing (only web browser tested).

Test Approach

- Manual testing will be done for all the features.
- Positive and negative scenarios will be covered.
- Exploratory testing will be used to find out usability issues.
- Defects will be reported with screenshots and linked to test case ID's.
- Later, automation (Cypress/Playwright) can reuse these manual cases.

Test Items

- Login page
- Product catalog and sorting
- Cart page and icon
- Checkout workflow (Your Information, Overview, Complete)
- Menu options
- Footer links

Entry and Exit Criteria

Entry Criteria

- Application is available at <https://www.saucedemo.com>
- Test accounts are provided
- Test environment is stable

Exit Criteria

- All planned test cases executed
- No open critical bugs.
- Test results documented in summary report

Deliverables

- Test Plan (this document)
- Test Case suite (55 manual test cases)
- Test Execution report (Pass/Fail cases)
- Defect Report (list of failed cases)
- Final Test Summary Report

Environment and Tools

- Browsers: Chrome, Firefox, Edge
- Tools: Browser DevTools, Google Docs/Excel for test cases, Screenshots for bug evidence

Risks

- Demo site may reset data unexpectedly (re-run tests if needed)
- Know demo limits (fake checkout)
- Security not production level (Document issues but treat as demo environment)

Schedule and Resources

- Tester: 1 QA Engineer
- Time: 2-3 days for full run
- Cycle: 1 full cycle + retest for failed or flaky tests

Conclusion

This test plan provides a simple but complete approach for testing SauceDemo. It ensures that the most important user flows are validated, issues are reported and the results are ready to share in a client report or portfolio.

Test Cases

Login and Session

Test Case ID	SD_LOGIN_01
Feature	Login
Description	Verify login with valid standard_user username
Precondition	User is on Saucedemo login page (http://www.saucedemo.com)
Test data	Username: standard_user Password: secret_sauce
Test steps	<ol style="list-style-type: none">1. Enter username2. Enter password3. Click Login button
Expected result	The user is redirected to the Product page with 6 products visible.
Actual result	✅ User redirected to Product page with 6 products visible.

Test Case ID	SD_LOGIN_02
Feature	Login
Description	Verify locked_out_user can't log in and shows error
Precondition	User is on login page
Test data	Username: locked_out_user Password: secret_sauce
Test steps	<ol style="list-style-type: none">1. Enter username2. Enter password3. Click Login button
Expected result	Error message should be displayed: "Epic sadface: Sorry, this user has been locked out."
Actual result	✅ Error message "Epic sadface: Sorry, this user has been locked out" is displayed:

Test Case ID	SD_LOGIN_03
Feature	Login
Description	Verify login fails with an invalid password and the correct error message is shown
Precondition	User is on login page
Test data	Username: standard_user Password: wrongpassword
Test steps	<ol style="list-style-type: none">1. Enter username2. Enter password3. Click Login button
Expected result	Error message should be displayed: "Epic sadface: Username and password do not match any user in this service"
Actual result	✅ Error message "Epic sadface: Username and password do not match any user in this service" is displayed

Test Case ID	SD_LOGIN_04
Feature	Login
Description	Verify error for blank username and password
Precondition	User is on login page
Test data	-
Test steps	<ol style="list-style-type: none"> 1. Leave both fields empty 2. Click Login button
Expected result	Error message should be displayed: "Epic sadface: Username is required"
Actual result	✅ Error message "Epic sadface: Username is required" is displayed

Test Case ID	SD_LOGIN_05
Feature	Login
Description	Verify error for blank password only
Precondition	User is on login page
Test data	Username: standard_user
Test steps	<ol style="list-style-type: none"> 1. Enter username 2. Leave password field empty 3. Click Login button
Expected result	Error message should be displayed: "Epic sadface: Password is required"
Actual result	✅ Error message "Epic sadface: Password is required" is displayed

Test Case ID	SD_LOGIN_06
Feature	Security
Description	Verify password field masks characters
Precondition	User is on login page
Test data	Password: secret_sauce
Test steps	<ol style="list-style-type: none"> 1. Type into password field
Expected result	Password characters are masked, input type="password"
Actual result	✅ Password hidden while typing


Test Case ID	SD_LOGIN_07
Feature	Session
Description	Verify Logout clears session and unable to access "Products" page
Precondition	User is logged in as standard_user on Products page
Test data	-
Test steps	<ol style="list-style-type: none"> 1. Open Menu top left corner 2. Click Logout 3. Go to https://www.saucedemo.com/inventory.html
Expected result	User is redirected to login page
Actual result	✅ Redirected to login page when opening Products page


Test Case ID	SD_LOGIN_08
Feature	Security
Description	Verify protected pages are blocked when logged out
Precondition	User is logged out on a fresh session
Test data	-
Test steps	1. Go to https://www.saucedemo.com/inventory.html
Expected result	User is redirected to login page and error message should be displayed
Actual result	✅ Redirected to login page and error message is displayed


Test Case ID	SD_LOGIN_09
Feature	Accessibility
Description	Verify Enter key submits login form
Precondition	User is on login page
Test data	Username: standard_user Password: secret_sauce
Test steps	1. Enter username 2. Enter password 3. Press Enter key
Expected result	Form submits and user is redirected to Products page
Actual result	✅ Login successful via Enter key

Test Case ID	SD_LOGIN_10
Feature	UX
Description	Verify error banner can be dismissed and reappears on next invalid submit
Precondition	User is on login page with error already visible
Test data	-
Test steps	1. Click X to dismiss the error 2. Submit invalid credentials again
Expected result	Error hides after dismissal, shows again on next invalid submit.
Actual result	✅ Dismissal works and error shows again on next submit

Inventory and sorting

Test Case ID	SD_INV_01
Feature	Inventory
Description	Verify Products page shows 6 items with name, price, image and Add to cart button
Precondition	User is on login page
Test data	Username: standard_user Password: secret_sauce
Test steps	<ol style="list-style-type: none">1. Enter valid credentials2. Count product tiles3. Inspect product fields on 6 tiles
Expected result	<ul style="list-style-type: none">• Exactly 6 tiles are present.• Product names are:<ol style="list-style-type: none">1. Sauce Labs Backpack (\$29.99)2. Sauce Labs Bike Light (\$9.99)3. Sauce Labs Bolt T-Shirt (\$15.99)4. Sauce Labs Fleece Jacket (\$49.99)5. Sauce Labs Onesie (\$7.99)6. Sauce Labs Red T-Shirt (\$15.99)• Each tile shows: name, description, price, product image, and Add to cart button.
Actual result	 All 6 tiles are displayed correctly: description, price, product image and Add to card button. Displayed name is "Test.allTheThings() T-Shirt (Red)" instead of "Sauce Labs Red T-Shirt"

Test Case ID	SD_INV_02
Feature	Sorting
Description	Verify sorting by Name (A to Z) orders items alphabetically ascending
Precondition	User is logged in and on Products page
Test data	-
Test steps	<ol style="list-style-type: none">1. Click and open sort dropdown menu2. Select Name (A to Z)
Expected result	Items are ordered from A to Z by product name
Actual result	 Order was A to Z


Test Case ID	SD_INV_03
Feature	Sorting
Description	Verify sorting by Name (Z to A) orders items alphabetically descending
Precondition	User is logged in and on Products page
Test data	-
Test steps	<ol style="list-style-type: none">1. Click and open sort dropdown menu2. Select Name (Z to A)
Expected result	Items are ordered from Z to A by product name
Actual result	 Order was Z to A


Test Case ID	SD_INV_04
Feature	Sorting
Description	Verify sorting by Price (low to high) uses numeric ascending order
Precondition	User is logged in and on Products page
Test data	-
Test steps	<ol style="list-style-type: none"> 1. Click and open sort dropdown menu 2. Select Price (low to high)
Expected result	Items appear in ascending price order (numeric)
Actual result	✅ Prices sorted low to high


Test Case ID	SD_INV_05
Feature	Sorting
Description	Verify sorting by Price (high to low) uses numeric descending order
Precondition	User is logged in and on Products page
Test data	-
Test steps	<ol style="list-style-type: none"> 1. Click and open sort dropdown menu 2. Select Price (high to low)
Expected result	Items appear in descending price order (numeric)
Actual result	✅ Prices sorted high to low

Test Case ID	SD_INV_06
Feature	Navigation
Description	Verify clicking product title opens detail page
Precondition	User is logged in and on Products page
Test data	Product: Sauce Labs Backpack
Test steps	<ol style="list-style-type: none"> 1. Click product title
Expected result	User lands on product detail page
Actual result	✅ Detail opened correctly via product title

Test Case ID	SD_INV_07
Feature	Navigation
Description	Verify clicking product image opens same detail page
Precondition	User is logged in and on Products page
Test data	Product: Sauce Labs Backpack
Test steps	<ol style="list-style-type: none"> 1. Click product image
Expected result	User lands on product detail page
Actual result	✅ Detail opened correctly via product image

Test Case ID	SD_INV_08
Feature	Accessibility
Description	Verify that each product image has alt attribute that matches the product name in the DOM (used by screen readers and when images fail to load, not shown as a tooltip)
Precondition	User is logged in and on Products page
Test data	-
Test steps	<ol style="list-style-type: none"> 1. Inspect image elements via DevTools 2. Check the value of the alt attribute on the tag 3. Check the alt attribute values for all 6 product images
Expected result	<ul style="list-style-type: none"> • Every product element has a non-empty alt attribute • Alt text should match the product's display name (e.g. "Sauce Labs Backpack", "Sauce Labs Bike Light", etc.)
Actual result	 All 6 images contain alt attributes, but one product has incorrect alt text: - Test.allTheThings() T-Shirt (Red) instead of the expected Sauce Labs Red T-Shirt.

Test Case ID	SD_INV_09
Feature	State
Description	Verify sort selection persists after visiting a detail page and navigation back
Precondition	User is logged in and on Products page with sort set to Price (high to low)
Test data	-
Test steps	<ol style="list-style-type: none"> 1. Open any product detail 2. Click Back to products
Expected result	Returns to Product with the same sort selected and order maintained
Actual result	 Sort is not persisted When navigating back, Products page reverts to default sort (Name A-Z)

Test Case ID	SD_INV_10
Feature	Reset
Description	Verify Reset App State clears cart and returns default sorting
Precondition	User is logged in and on Products page after adding items and changing order
Test data	-
Test steps	<ol style="list-style-type: none"> 1. Click Menu top left corner 2. Click Reset App State 3. Refresh the page
Expected result	Cart is empty and sort returns to default (A to Z)
Actual result	 Cart is cleared and sort returns to default sorting

Cart

Test Case ID	SD_CART_01
Feature	Cart
Description	Verify adding an item to cart from inventory tile updates button and cart icon
Precondition	User is logged in and on Products page
Test data	Product: Sauce Labs Backpack
Test steps	1. Click Add to cart button on the tile
Expected result	Button changes to Remove and cart icon shows 1
Actual result	✅ Button switches to Remove and cart icon shows 1

Test Case ID	SD_CART_02
Feature	Cart
Description	Verify removing an item from inventory tile reverts button and cart icon
Precondition	User is logged in and on Products page
Test data	Product: Sauce Labs Backpack
Test steps	1. Click Remove button on the tile
Expected result	Button returns to Add to cart and cart icon goes to 0
Actual result	✅ Button returned to Add to cart and icon went to 0

Test Case ID	SD_CART_03
Feature	Cart
Description	Verify adding an item to cart from products detail page updates button and cart icon
Precondition	User is logged in and on Products page
Test data	Any product
Test steps	1. Click Add to cart 2. Click Back to Products
Expected result	Selected product tile button shows Remove and cart icon increases by 1
Actual result	✅ Selected product tile button showed Remove and cart icon increased by 1

Test Case ID	SD_CART_04
Feature	Cart
Description	Verify Cart page lists selected items with correct quantity and prices
Precondition	User has at least two different items in cart and opens Cart page
Test data	Two different items added to cart
Test steps	1. Click cart icon top right corner 2. Review added product items
Expected result	Two items have correct quantity and prices
Actual result	✅ Two items listed have correct quantity and prices

Test Case ID	SD_CART_05
Feature	Cart
Description	Verify removing an item in Cart updates totals and icon number
Precondition	User is on Cart page
Test data	Two different items present
Test steps	1. Click Remove for 1 item
Expected result	Item row disappears and cart icon number decreases accordingly
Actual result	✅ Row removed and cart icon updated

Test Case ID	SD_CART_06
Feature	Cart
Description	Verify Continue Shopping returns to Products page and preserves cart
Precondition	User is on Cart page
Test data	-
Test steps	1. Click Continue Shopping
Expected result	User returns to Products page with cart unchanged
Actual result	✅ Returned to products page and cart is unchanged

Test Case ID	SD_CART_07
Feature	Cart
Description	Verify cart icon count number shows correct number by adding multiple different items
Precondition	User is on Products page
Test data	Add 3 different items
Test steps	<ol style="list-style-type: none"> 1. Add item 1 2. Add item 2 3. Add item 3 4. Review cart icon count
Expected result	Cart icon shows 3 items
Actual result	✅ Cart icon shows 3 items

Test Case ID	SD_CART_08
Feature	State
Description	Verify state sync when adding item in product detail page and removing from products page list
Precondition	User has added an item to cart on its detail page
Test data	Same item
Test steps	<ol style="list-style-type: none"> 1. Click Back to Products 2. Click Remove on tile
Expected result	Details page and list stay in sync and cart icon shows correct number
Actual result	✅ States synced and correct cart icon number

Test Case ID	SD_CART_09
Feature	Reliability
Description	Verify cart contents stay after page refresh
Precondition	User has item in cart and is on Products page
Test data	-
Test steps	1. Press browser refresh
Expected result	Cart contents and icon stays during the session
Actual result	✅ Cart stays after page refresh

Test Case ID	SD_CART_10
Feature	Reliability
Description	Verify add/remove toggle works repeatedly without desync
Precondition	User is on Products page
Test data	1 random product
Test steps	<ol style="list-style-type: none"> 1. Click Add to cart 2. Click Remove 3. Add again same item
Expected result	Tile state and cart icon always show current state
Actual result	✅ Toggle remained consistent

Checkout

Test Case ID	SD_CHECKOUT_01
Feature	Checkout
Description	Verify clicking “Checkout” on Cart navigates to “Checkout: Your information” page
Precondition	User has 1 or more items in the cart and on Cart page
Test data	-
Test steps	1. Click Checkout
Expected result	User navigates to Checkout: Your information page
Actual result	✅ Navigation “Checkout: Your information page” was successful

Test Case ID	SD_CHECKOUT_02
Feature	Validation
Description	Verify First Name is mandatory on Checkout: Your information page
Precondition	User is on Checkout: Your information page page
Test data	Leave all blanks empty
Test steps	1. Click Continue
Expected result	Error message “Error: First Name is required” should be displayed
Actual result	✅ Error message is displayed

Test Case ID	SD_CHECKOUT_03
Feature	Validation
Description	Verify Last Name is mandatory on Checkout: Your information page
Precondition	User is on Checkout: Your information page page
Test data	First name: Mari
Test steps	<ol style="list-style-type: none"> 1. Enter first name 2. Click Continue
Expected result	Error message "Error: Last Name is required" should be displayed
Actual result	✔ Error message is displayed

Test Case ID	SD_CHECKOUT_04
Feature	Validation
Description	Verify Postal Code is mandatory on Checkout: Your information page
Precondition	User is on Checkout: Your information page page
Test data	First name: Mari Last name: Tamm
Test steps	<ol style="list-style-type: none"> 1. Enter First and Last name 2. Click Continue
Expected result	Error message "Error: Postal Code is required" should be displayed
Actual result	✔ Error message is displayed

Test Case ID	SD_CHECKOUT_05
Feature	Usability
Description	Verify that mandatory field (First Name, Last Name, Postal Code) are visually indicated as required on Checkout: Your Information
Precondition	User is on Checkout: Your information page page
Test data	-
Test steps	<ol style="list-style-type: none"> 1. Inspect the form fields for First Name, Last Name, Postal Code 2. Look for any required field indicators (such as red *, red border, helper text "required")
Expected result	Each mandatory field should have a visual indicator that is required such as red *, red border, helper text "required"). This should be clear without submitting the form
Actual result	✘ Fields are mandatory but have no visual indicators. User only discovers the requirement after submitting and receiving error messages.

Test Case ID	SD_CHECKOUT_06
Feature	Checkout
Description	Verify valid info proceeds to Checkout Overview
Precondition	User is on Checkout: Your information page page
Test data	First name: Mari Last name: Tamm Postal code: 12345
Test steps	<ol style="list-style-type: none"> 1. Fill the fields 2. Click Continue
Expected result	User navigates to Checkout Overview with items and totals
Actual result	✔ Navigation is successful

Test Case ID	SD_CHECKOUT_07
Feature	Totals
Description	Verify item totals equals the sum of item prices on Checkout: Overview
Precondition	User is on Checkout: Overview page with known added items
Test data	Record item prices
Test steps	<ol style="list-style-type: none"> 1. Sum item prices 2. Compare with displayed Subtotal
Expected result	Displayed item total equals the sum of item prices
Actual result	✅ Item total matched manual sum

Test Case ID	SD_CHECKOUT_08
Feature	Totals
Description	Verify Total equals Item totals plus Tax on Checkout: Overview
Precondition	User is on Checkout: Overview page
Test data	Calculate Item Total and Tax
Test steps	<ol style="list-style-type: none"> 1. Manually calculate Total = Item Total+Tax 2. Compare with displayed Total
Expected result	Displayed Total equals calculated Total.
Actual result	✅ Total equaled Item Total + Tax

Test Case ID	SD_CHECKOUT_09
Feature	Price
Description	Verify item total and tax values are displayed with correct decimal formatting
Precondition	User is on Checkout: Overview page with multiple items in cart
Test data	-
Test steps	<ol style="list-style-type: none"> 1. Observe "Item total"
Expected result	Item total should display with 2 decimal places.
Actual result	❌ Displayed Item total have more than 2 decimal places.

Test Case ID	SD_CHECKOUT_10
Feature	Workflow
Description	Verify Finish completes the order and clears the cart
Precondition	User is on Checkout: Overview page with multiple items in cart
Test data	-
Test steps	<ol style="list-style-type: none"> 1. Click Finish
Expected result	User lands on Checkout Complete page
Actual result	✅ Order completed and Cart icon cleared

Test Case ID	SD_CHECKOUT_11
Feature	Workflow
Description	Verify Cancel from Overview returns to Products and preserves cart
Precondition	User is on Checkout: Overview page with multiple items in cart
Test data	-
Test steps	1. Click Cancel
Expected result	User returns to Products page and items remain in the cart
Actual result	✅ Returned to Products page and items are still in cart

Test Case ID	SD_CHECKOUT_12
Feature	Workflow
Description	Verify Back Home button on Checkout: Complete returns user to the Products page
Precondition	User has completed the checkout and is on the Checkout: Overview page with multiple items previously added to the cart
Test data	-
Test steps	1. Click Back Home button
Expected result	User is redirected to the Products page and cart is cleared because checkout was completed
Actual result	✅ Returned to Products page and cart is empty

Menu

Test Case ID	SD_MENU_01
Feature	Menu
Description	Verify About link opens Sauce Labs website
Precondition	User is on Products page
Test data	-
Test steps	1. Open Menu 2. Click About
Expected result	Browser opens Sauce labs website in same tab
Actual result	✅ Opened Sauce Labs site

Test Case ID	SD_MENU_02
Feature	Menu
Description	Verify Reset App State from Menu clears cart and sort
Precondition	User has items in cart and non-default sort on Products
Test data	-
Test steps	1. Open Menu 2. Click Reset App State 3. Refresh Products page 4. Click Cart icon
Expected result	All product buttons show Add to cart, sort is back to default and cart is empty
Actual result	✅ State is fully reset

Test Case ID	SD_MENU_03
Feature	Footer
Description	Verify Twitter footer link opens correct profile
Precondition	User is on Products page
Test data	-
Test steps	1. Click Twitter icon in footer
Expected result	Twitter profile opens in new tab
Actual result	✅ Twitter opened

Test Case ID	SD_MENU_04
Feature	Footer
Description	Verify Facebook footer link opens correct profile
Precondition	User is on Products page
Test data	-
Test steps	1. Click Facebook icon in footer
Expected result	Facebook profile opens in new tab
Actual result	✅ Facebook opened


Test Case ID	SD_MENU_05
Feature	Footer
Description	Verify LinkedIn footer link opens correct profile
Precondition	User is on Products page
Test data	-
Test steps	1. Click LinkedIn icon in footer
Expected result	LinkedIn profile opens in new tab
Actual result	✅ LinkedIn opened


Non-functional


Test Case ID	SD_NONFUNC_01
Feature	Performance
Description	Measure time to Products page after login while having basic conditions
Precondition	User is on Login page
Test data	Username: standard_user Password: secret_sauce
Test steps	<ol style="list-style-type: none">1. Enter credentials2. Open DevTools - Network3. Click Login button
Expected result	Inventory page loads within acceptable time ≤ 3 seconds (Google suggests that pages should load in under 2-3 seconds for optimal user engagement.)
Actual result	✅ Products page loaded under ≤ 3 seconds


Test Case ID	SD_NONFUNC_02
Feature	UI
Description	Verify price format consistency
Precondition	User is on Products page
Test data	-
Test steps	<ol style="list-style-type: none">1. Review all visible prices
Expected result	Each price shows \$ and two decimal places
Actual result	✅ All prices formatted with \$ and two decimal places

Test Case ID	SD_NONFUNC_03
Feature	Responsive
Description	Verify main flows on mobile viewport 412x915 (Samsung Galaxy S20 Ultra)
Precondition	User is logged in and on Products page with DevTools mobile emulation 412x915 (Samsung Galaxy S20 Ultra)
Test data	Username: standard_user Password: secret_sauce
Test steps	<ol style="list-style-type: none">1. Enter credentials and click Login2. Pick random item and click Add to cart3. Click cart icon4. Click Checkout
Expected result	Layout should be usable, no overlapping or hidden controls
Actual result	✅ Layout is usable, no overlapping or hidden controls

Test Case ID	SD_NONFUNC_04
Feature	Accessibility
Description	Verify visible keyboard focus on interactive controls
Precondition	User is on any Sauce Demo page
Test data	-
Test steps	1. Press Tab repeatedly to move through all interactive elements (buttons, links, inputs, menu, cart)
Expected result	Each interactive element should have visible focus indicator (outline or highlight) when selected via keyboard
Actual result	 No visible focus indicator appears when tabbing through elements.

Test Case ID	SD_NONFUNC_05
Feature	Security
Description	Verify credentials are not stored in URL or plaintext storage
Precondition	User is logged in and on Products page
Test data	-
Test steps	1. Inspect URL parameters 2. Open DevTools - Application - Local Storage and Session Storage
Expected result	No username or password present in URL, Local Storage or Session Storage
Actual result	 No username or password found

Test Case ID	SD_NONFUNC_06
Feature	Security
Description	Verify sensitive data (username/password) is not stored in cookies
Precondition	User is logged in and on Products page
Test data	Username: standard_user Password: secret_sauce
Test steps	1. Open DevTools - Application - Cookies 2. Review cookie name values and flags
Expected result	No username or password should be visible. Only a random session ID/token
Actual result	 Cookie session-username=standard_user is stored in clear text

Test Case ID	SD_NONFUNC_07
Feature	Stability
Description	Stress sort toggling without errors
Precondition	User is logged in and on Products page
Test data	-
Test steps	1. Rapidly switch between all sort options for 30 seconds
Expected result	There should be no crashes and list updates correctly each time
Actual result	 No errors

Test Case ID	SD_NONFUNC_08
Feature	Reliability
Description	Verify refreshing Check Overview preserves state
Precondition	User is on Checkout Overview page
Test data	-
Test steps	1. Refresh page
Expected result	Items, totals, tax should remain visible after reload
Actual result	✅ State stayed after refresh

Test Case ID	SD_NONFUNC_09
Feature	Navigation
Description	Verify browser Back from Overview returns to step one with data intact
Precondition	User reached Overview after filling step one
Test data	-
Test steps	1. Click browser Back
Expected result	User should be returned to Checkout: Your information and fields retain values
Actual result	❌ Returned to step one with data intact

Test Case ID	SD_NONFUNC_10
Feature	UX
Description	Verify error messages are clear and actionable on Login and Checkout: Your information
Precondition	User is on Login page or Checkout: Your information with at least 1 item in the cart
Test data	Login: <ul style="list-style-type: none"> Username: standard_user Password: wrongpassword Checkout: <ul style="list-style-type: none"> First Name = Leave field empty
Test steps	1. On login page attempt login with invalid password 2. On checkout: Your information, leave First Name field empty and click Continue
Expected result	Login error: Epic sadface: Username and password do not match any user in this service Checkout error: Error: First Name is required
Actual result	✅ Both errors are displayed exactly as expected

Test Case ID	SD_NONFUNC_11
Feature	Performance
Description	Verify images load reasonably under throttled network
Precondition	User is logged in and on Products page with network throttled using 3G in DevTools
Test data	-
Test steps	1. DevTools - Network - Throttle (3G) 2. Observe image loading behavior
Expected result	Images should load eventually and no broken images or icons
Actual result	✅ Images load eventually and no broken images or icons

Edge

Test Case ID	SD_EDGE_01
Feature	Edge
Description	Validate username case sensitivity on Login
Precondition	User is on login page
Test data	Username: Standard_user Password: secret_sauce
Test steps	<ol style="list-style-type: none">1. Enter credentials with wrong case2. Click login
Expected result	Login should fail if username does not exactly match the stored account
Actual result	✅ Error message "Epic sadface: Username and password do not match any user in this service" is displayed

Test Case ID	SD_EDGE_02
Feature	Edge
Description	Validate cart state across multiple tabs
Precondition	User is logged in on two tabs on Product page
Test data	Add an item in Tab A
Test steps	<ol style="list-style-type: none">1. Add item in Tab A2. Switch to Tab B and refresh
Expected result	Cart state should match after refresh (Cart icon and Cart items)
Actual result	✅ State matched in both tabs after refresh

Test Case ID	SD_EDGE_03
Feature	Edge
Description	Validate direct deep link to an item detail works within session
Precondition	User is logged in and on Products page
Test data	Add an item in Tab A
Test steps	<ol style="list-style-type: none">3. Open deep link URL4. Click Back to products
Expected result	Product detail should open and clicking Back to products returns to Products
Actual result	✅ Deep link worked and clicking Back to products returned to Products

Completed Test cases and Report:

Test Case ID	Description	Status	Actual results Date of testing: 23.09.2023 Browser: Chrome Version140.0.7339.127 (Official Build) (64-bit)
SD_LOGIN_01	Verify login with valid standard_user username	✓	User redirected to Product page with 6 products visible.
SD_LOGIN_02	Verify locked_out_user can't log in and shows error	✓	Error message "Epic sadface: Sorry, this user has been locked out" is displayed
SD_LOGIN_03	Verify login fails with an invalid password and the correct error message is shown	✓	Error message "Epic sadface: Username and password do not match any user in this service" is displayed
SD_LOGIN_04	Verify error for blank username and password	✓	Error message "Epic sadface: Username is required" is displayed
SD_LOGIN_05	Verify error for blank password only	✓	Error message "Epic sadface: Password is required" is displayed
SD_LOGIN_06	Verify password field masks characters	✓	Password hidden while typing
SD_LOGIN_07	Verify Logout clears session and unable to access "Products" page	✓	Redirected to login page when opening Products page
SD_LOGIN_08	Verify protected pages are blocked when logged out	✓	Redirected to login page and error message is displayed
SD_LOGIN_09	Verify Enter key submits login form	✓	Login successful via Enter key
SD_LOGIN_10	Verify error banner can be dismissed and reappears on next invalid submit	✓	Dismissal works and error shows again on next submit
SD_INV_01	Verify Products page shows 6 items with name, price, image and Add to cart button	✗	All 6 tiles are displayed correctly: description, price, product image and Add to card button. Displayed name is "Test.allTheThings() T-Shirt (Red)" instead of "Sauce Labs Red T-Shirt"
SD_INV_02	Verify sorting by Name (A to Z) orders items alphabetically ascending	✓	Order was A to Z
SD_INV_03	Verify sorting by Name (Z to A) orders items alphabetically descending	✓	Order was Z to A
SD_INV_04	Verify sorting by Price (low to high) uses numeric ascending order	✓	Prices sorted low to high
SD_INV_05	Verify sorting by Price (high to low) uses numeric descending order	✓	Prices sorted high to low
SD_INV_06	Verify clicking product title opens detail page	✓	Detail opened correctly via product title
SD_INV_07	Verify clicking product image opens same detail page	✓	Detail opened correctly via product image
SD_INV_08	Verify product images have correct alt texts	✗	All 6 images contain alt attributes, but one product has incorrect alt text: - Test.allTheThings() T-Shirt (Red) instead of the expected Sauce Labs Red T-Shirt.
SD_INV_09	Verify sort selection persists after visiting a detail page and navigation back	✗	Sort is not persisted When navigating back, Products page reverts to default sort (Name A-Z)
SD_INV_10	Verify Reset App State clears cart and returns default sorting	✓	Cart is cleared and sort returns to default sorting

SD_CART_01	Verify adding an item to cart from inventory tile updates button and cart icon	✓	Button switches to Remove and cart icon shows 1
SD_CART_02	Verify removing an item from inventory tile reverts button and cart icon	✓	Button returned to Add to cart and icon went to 0
SD_CART_03	Verify adding an item to cart from products detail page updates button and cart icon	✓	Selected product tile button showed Remove and cart icon increased by 1
SD_CART_04	Verify Cart page lists selected items with correct quantity and prices	✓	Two items listed have correct quantity and prices
SD_CART_05	Verify removing an item in Cart updates totals and icon number	✓	Row removed and cart icon updated
SD_CART_06	Verify Continue Shopping returns to Products page and preserves cart	✓	Returned to products page and cart is unchanged
SD_CART_07	Verify cart icon count number shows correct number by adding multiple different items	✓	Cart icon shows 3 items
SD_CART_08	Verify state sync when adding item in product detail page and removing from products page list	✓	States synced and correct cart icon number
SD_CART_09	Verify cart contents stay after page refresh	✓	Cart stays after page refresh
SD_CART_10	Verify add/remove toggle works repeatedly without desync	✓	Toggle remained consistent
SD_CHECKOUT_01	Verify clicking "Checkout" on Cart navigates to "Checkout: Your information" page	✓	Navigation "Checkout: Your information page" was successful
SD_CHECKOUT_02	Verify First Name is mandatory on Checkout: Your information page	✓	Error message is displayed
SD_CHECKOUT_03	Verify Last Name is mandatory on Checkout: Your information page	✓	Error message is displayed
SD_CHECKOUT_04	Verify Postal Code is mandatory on Checkout: Your information page	✓	Error message is displayed
SD_CHECKOUT_05	Verify that mandatory field (First Name, Last Name, Postal Code) are visually indicated as required on Checkout: Your Information	✗	Fields are mandatory but have no visual indicators. User only discovers the requirement after submitting and receiving error messages.
SD_CHECKOUT_06	Verify valid info proceeds to Checkout Overview	✓	Navigation is successful
SD_CHECKOUT_07	Verify item totals equals the sum of item prices on Checkout: Overview	✓	Item total matched manual sum
SD_CHECKOUT_08	Verify Total equals Item totals plus Tax on Checkout: Overview	✓	Total equaled Item Total + Tax
SD_CHECKOUT_09	Verify item total and tax values are displayed with correct decimal formatting	✗	Displayed Item total have more than 2 decimal places.
SD_CHECKOUT_10	Verify Finish completes the order and clears the cart	✓	Order completed and Cart icon cleared
SD_CHECKOUT_11	Verify Cancel from Overview returns to Products and preserves cart	✓	Returned to Products page and items are still in cart
SD_CHECKOUT_12	Verify Back Home button on Checkout: Complete returns user to the Products page	✓	Verify Back Home button on Checkout: Complete returns user to the Products page
SD_MENU_01	Verify About link opens Sauce Labs website	✓	Opened Sauce Labs site

SD_MENU_02	Verify Reset App State from Menu clears cart and sort	✓	State is fully reset
SD_MENU_03	Verify Twitter footer link opens correct profile	✓	Twitter opened
SD_MENU_04	Verify Facebook footer link opens correct profile	✓	Facebook opened
SD_MENU_05	Verify LinkedIn footer link opens correct profile	✓	LinkedIn opened
SD_NONFUNC_01	Measure time to Products page after login while having basic conditions	✓	Products page loaded under ≤ 3 seconds
SD_NONFUNC_02	Verify price format consistency	✓	All prices formatted with \$ and two decimal places
SD_NONFUNC_03	Verify main flows on mobile viewport 412x915 (Samsung Galaxy S20 Ultra)	✓	Layout is usable, no overlapping or hidden controls
SD_NONFUNC_04	Verify visible keyboard focus on interactive controls	✗	No visible focus indicator appears when tabbing through elements.
SD_NONFUNC_05	Verify credentials are not stored in URL or plaintext storage	✓	No username or password found
SD_NONFUNC_06	Verify sensitive data (username/password) is not stored in cookies	✗	Cookie "session-username=standard_user" is stored in clear text
SD_NONFUNC_07	Stress sort toggling without errors	✓	There should be no crashes and list updates correctly each time
SD_NONFUNC_08	Verify refreshing Check Overview preserves state	✓	State stayed after refresh
SD_NONFUNC_09	Verify browser Back from Overview returns to step one with data intact	✗	Returned to step one with data intact
SD_NONFUNC_10	Verify error messages are clear and actionable on Login and Checkout: Your information	✓	Both errors are displayed exactly as expected
SD_NONFUNC_11	Verify images load reasonably under throttled network	✗	Images load eventually and no broken images or icons
SD_EDGE_01	Validate username case sensitivity on Login	✓	Error message "Epic sadface: Username and password do not match any user in this service" is displayed
SD_EDGE_02	Validate cart state across multiple tabs	✓	State matched in both tabs after refresh
SD_EDGE_03	Validate direct deep link to an item detail works within session	✓	Deep link worked and clicking Back to products returned to Products

Test Summary

Executed 55 manual test cases on the SauceDemo web application covering login, product inventory, cart, checkout, menu, footer, non-functional and edge scenarios. Out of 55 cases 48 passed and 7 failed.

Web application demonstrates strong functional stability:

- Login and session handling work correctly across all tested scenarios.
- Basic shopping flows (adding items, cart updates, checkout navigation) performs as expected.
- Sorting, navigation and workflow consistency are reliable.

However, several issues were identified:

- Usability gaps such as missing required field indicators on checkout
- Accessibility issues including missing focus indicators for keyboard navigation.
- Data quality problems such as mislabeled product name and incorrect alt text.
- Formatting defect where price totals display excessive decimal places.
- Security concern where the username is stored in cookies in plain text

Overall Assessment

The web application is suitable for demo and basic e-commerce workflows. To improve quality and align with best practices we recommend addressing the identified defects with priority on security and accessibility.