



Assignment

Task 1

Web Automation: Selenium

Language: Java

Maven

TestNG

Haseeb Saleem

Contents

LoginTests.....	3
1. validLoginTest.....	3
2. invalidLoginTest.....	3
3. emptyFieldsLoginTest.....	3
4. lockedOutUserLoginTest	3
5. problemUserLoginTest	3
6. performanceGlitchUserLoginTest.....	3
7. errorUserLoginTest.....	4
8. visualUserLoginTest.....	4
InventoryTests	4
1. sortProductsByNameTest	4
2. sortProductsByPriceLowToHighTest.....	4
3. checkProductAmountTest	4
4. checkAddToCartButtonTest	5
DrawerTest	5
1. openLeftDrawerTest	5
2. aboutPageTest.....	5
3. logoutTest.....	5
4. resetAppStateTest	5
CheckoutTests	6
1. addToCartAndCheckoutTest	6
AddToCartTests.....	6
1. addProductAndVerifyOnCartPageTest	6
2. removeProductFromCartTest.....	6
3. addMultipleProductsToCartTest.....	6
4. addAndRemoveMultipleProductsFromCartTest	6

LoginTests

1. validLoginTest

Purpose: This test case verifies the successful login by providing valid username and password.

Functionality:

- Calls the login method on the LoginPage object with valid credentials.
- Verifies the successful login by checking the presence of the inventory container element on the next page.
- Uses Assert.assertTrue to confirm the success.

2. invalidLoginTest

Purpose: This test case verifies the handling of an invalid login attempt.

Functionality:

- Calls the login method on the LoginPage object with invalid username and password.
- Verifies the displayed error message for an invalid login attempt.
- Uses Assert.assertTrue to check if the error message matches the expected pattern.

3. emptyFieldsLoginTest

Purpose: This test case verifies the handling of a login attempt with empty username and password fields.

Functionality:

- Calls the login method on the LoginPage object with empty username and password.
- Verifies the displayed error message for empty fields.
- Uses Assert.assertEquals to check if the error message matches the expected message.

4. lockedOutUserLoginTest

Purpose: This test case verifies the handling of a login attempt with a locked-out user.

Functionality:

- Calls the login method on the LoginPage object with a locked-out user's credentials.
- Verifies the displayed error message for a locked-out user.
- Uses Assert.assertTrue to check if the error message matches the expected pattern.

5. problemUserLoginTest

Purpose: This test case verifies the successful login with a user known to have no issues.

Functionality:

- Calls the login method on the LoginPage object with valid credentials.
- Verifies the successful login by checking the presence of the inventory container element on the next page.
- Uses Assert.assertTrue to confirm the success.

6. performanceGlitchUserLoginTest

Purpose: This test case verifies the successful login with a user known to have performance glitches.

Functionality:

- Calls the login method on the LoginPage object with valid credentials.

- Verifies the successful login by checking the presence of the inventory container element on the next page.
- Uses `Assert.assertTrue` to confirm the success.

7. `errorUserLoginTest`

Purpose: This test case verifies the successful login with a user known to generate a specific product description error.

Functionality:

- Calls the login method on the `LoginPage` object with valid credentials.
- Verifies the product description on the next page, checking for a specific error-related text.
- Uses `Assert.assertTrue` to confirm the presence of the expected error text.

8. `visualUserLoginTest`

Purpose: This test case verifies the handling of a login attempt with a visual user, specifically checking for a visual failure element.

Functionality:

- Calls the login method on the `LoginPage` object with a visual user's credentials.
- Verifies the presence of a visual failure element in the shopping cart container.
- Uses `Assert.assertTrue` to confirm the presence of the visual failure element.

InventoryTests

1. `sortProductsByNameTest`

Purpose: This test case verifies the functionality of sorting products by name on the Inventory Page.

Functionality:

- Clicks on the sort dropdown to choose the option to sort products by name.
- Retrieves the list of product names from the page.
- Iterates through the list to ensure that products are sorted alphabetically by name.
- Uses `Assert.assertTrue` to confirm the sorting order.

2. `sortProductsByPriceLowToHighTest`

Purpose: This test case verifies the functionality of sorting products by price (low to high) on the Inventory Page.

Functionality:

- Clicks on the sort dropdown to choose the option to sort products by name (as a pre-step).
- Clicks on the "Price (low to high)" option to sort products by price.
- Retrieves the list of product prices from the page.
- Iterates through the list to ensure that products are sorted in ascending order of price.
- Uses `Assert.assertTrue` to confirm the sorting order.

3. `checkProductAmountTest`

Purpose: This test case checks the format of the product amount on the Inventory Page.

Functionality:

- Retrieves all product elements containing the product amount.

- Iterates through the list to ensure that each product amount starts with the "\$" symbol.
- Uses Assert.assertTrue to confirm the format.

4. checkAddToCartButtonTest

Purpose: This test case checks the visibility of "Add to Cart" buttons on the Inventory Page.

Functionality:

- Retrieves all "Add to Cart" buttons on the page.
- Iterates through the list to ensure that each "Add to Cart" button is displayed.
- Uses Assert.assertTrue to confirm the visibility of each button.

DrawerTest

1. openLeftDrawerTest

Purpose: This test case verifies the functionality of opening the left drawer and checking its content.

Functionality:

- Clicks on the menu icon to open the left drawer.
- Locates the "All Items" link in the left drawer.
- Asserts that the link contains the text "All Items."

2. aboutPageTest

Purpose: This test case verifies the functionality of navigating to the About page from the left drawer.

Functionality:

- Clicks on the menu icon to open the left drawer.
- Clicks on the "About" button in the left drawer.
- Locates a paragraph element on the About page.
- Asserts that the paragraph contains the expected text.

3. logoutTest

Purpose: This test case verifies the functionality of logging out from the left drawer.

Functionality:

- Clicks on the menu icon to open the left drawer.
- Clicks on the "Logout" button in the left drawer.
- Locates a paragraph element on the login page.
- Asserts that the paragraph contains text indicating accepted usernames.

4. resetAppStateTest

Purpose: This test case verifies the functionality of resetting the application state from the left drawer.

Functionality:

- Clicks on the "Add to Cart" button to add an item to the cart.
- Clicks on the menu icon to open the left drawer.
- Clicks on the "Reset" button in the left drawer.
- Attempts to find an element with the class 'shopping_cart_badge.'
- Asserts that the element does not exist, indicating a reset application state.

CheckoutTests

1. addToCartAndCheckoutTest

Purpose: This test case verifies the functionality of the add to cart and checkout process.

Functionality:

- Clicks on the cart icon to navigate to the cart.
- Clicks on the checkout button within the cart.
- Fills in the checkout information (first name, last name, and zip code) using predefined values.
- Verifies that the checkout summary page is displayed by checking the presence of the title element.
- Clicks on the finish button to complete the checkout process.
- Verifies that the thank you page is displayed by checking the presence of the "complete-header" element.

AddToCartTests

1. addProductAndVerifyOnCartPageTest

Purpose: This test case adds a product to the cart and verifies its presence on the cart page.

Functionality:

- Verifies that the cart icon is updated to show the correct number of items (1 in this case).
- Clicks on the cart icon to navigate to the cart page.
- Verifies that the added product is displayed in the cart.

2. removeProductFromCartTest

Purpose: This test case removes a product from the cart and verifies its absence.

Functionality:

- Navigates to the cart page.
- Clicks on the remove button to remove the product from the cart.
- Verifies that the cart is now either empty or contains the correct number of items.

3. addMultipleProductsToCartTest

Purpose: This test case adds multiple products to the cart and verifies the count.

Functionality:

- Finds and clicks on all "Add to Cart" buttons on the page.
- Verifies the count of items in the cart badge, ensuring it is greater than 1.

4. addAndRemoveMultipleProductsFromCartTest

Purpose: This test case adds and then removes multiple products from the cart.

Functionality:

- Calls the addMultipleProductsToCartTest method to add multiple products to the cart.
- Opens the shopping cart.
- Finds and clicks on all "Remove" buttons in the cart.
- Verifies that the cart is empty after removing the products.