**Manual Testing Strategy**

1. Confirm the lowest price in the store:

- Implementation:

- Open the application in a web browser.

- Navigate to the store page.

- Locate and inspect the HTML elements containing item prices.

- Extract the list of prices programmatically using browser developer tools.

- Confirm that the lowest price among the extracted prices is 7.99.

- Documentation:

- Record the steps taken to extract prices.

- Note down the lowest price found.

- Report any issues or discrepancies.

2. Test the checkout process:

- Implementation:

- Begin with an empty cart and navigate to the product catalog.

- Add items to the cart.

- Proceed to checkout.

- For each step in the checkout flow (shipping, payment, confirmation), verify that the page loads correctly and accepts user inputs.

- Ensure that submitted information is processed accurately.

- Documentation:

- Create detailed test cases for each step of the checkout flow.

- Record the results of each test case.

- Document any errors encountered during the checkout process.

3. Ensure the order item's total is showing correctly based on products added to the cart:

- Implementation:

- Add multiple items with different prices and quantities to the cart.

- Proceed to the checkout page.

- Verify that the item total accurately reflects the selected products' prices and quantities.

- Documentation:

- List the items added to the cart.

- Record the expected item total and compare it with the actual total.

- Document any discrepancies.

4. Ensure the order tax:

- Implementation:

- Add items to the cart.

- Proceed to the checkout page.

- Verify that the calculated order tax falls within the range of 5% to 10% of the item price total.

- Documentation:

- Record the order tax amount.

- Calculate the expected tax range based on the item price total.

- Document any deviations from the expected tax range.

5. Reusable login page object method:

- Implementation:

- Create a reusable login method that takes credentials as parameters.

- Verify that users can log in successfully using the method.

- Verify that users can log out from the application.

- Attempt to access /inventory.html without being logged in and assert that the error message is displayed.

- Documentation:

- Document the steps to create and use the reusable login method.

- Record successful login and logout attempts.

- Document any issues related to access control.

6. Ensure the cart shows the correct number when the user removes the inventory from the cart:

- Implementation:

- Add items to the cart and verify that the cart badge updates correctly.

- Test the "Add" and "Remove" buttons to ensure they increment and decrement the cart count accurately.

- Documentation:

- Document the initial cart count.

- Record the changes in cart count after using the "Add" and "Remove" buttons.

- Report any discrepancies.

7. Avoid hard-coded username and password:

- Implementation:

- Move the login information (username and password) into an environment configuration object.

- Update the login process to use the information from the config object.

- Verify that the login process still functions correctly.

- Documentation:

- Document the changes made to move login information to the config object.

- Confirm successful login after the configuration change.

- Report any issues related to configuration.

8. Test the item product view:

- Implementation:

- Navigate to the product view for a specific item.

- Verify that the product information (price, description) is displayed accurately.

- Documentation:

- Record the product details displayed.

- Document any discrepancies in the product view.

9. Confirm the quantity change on checkout:

- Implementation:

- Add a specific product to the cart with a quantity of two.

- Proceed to the checkout page.

- Confirm that the selected quantity is reflected correctly during the checkout process.

- Documentation:

- Document the product added and the specified quantity.

- Record the quantity displayed during checkout.

- Report any issues with quantity validation.

10. Load item data from a fixture file:

- Implementation:

- Create a JSON fixture file containing item data.

- Modify the application to load item data from the fixture file.

- Verify that the item data from the fixture file is displayed accurately on relevant pages.

- Documentation:

- Document the creation of the fixture file.

- Describe the changes made to load item data from the fixture.

- Record any discrepancies between fixture data and actual data.