Test Plan

Test objectives:

* To verify that the title appears when the application is running.
* To verify that the user is able to view values from the drop-down menu.
* To verify that the user is able to add items to the cart using the “add to cart” button.
* To verify that the user is able to apply a discount code in the text box.
* To verify that the discount button applies the discount code.
* To verify that the user is able to remove an item from the cart.
* To verify that an error message is generated for invalid item removal.

Test Strategy:

* Functional Testing: Verify that all functional requirements are met, including making sure the title appears when the application is running, testing drop-down menus, adding and removing items from the cart, checking to ensure discounts are applied, and generating error messages.

Test Scope: The scope of testing includes the following aspects of the application:

* Title Validation
* Drop-down testing
* Adding items to cart
* Adding discount code
* Applying discount code
* Removing an item
* Error message testing

Features to be tested: key features to be tested include:

* Selecting an option from a drop-down menu
* Adding an item to the cart
* Removing an item from the cart
* Entering the discount code
* Applying a discount code
* Error messaging when trying to remove an invalid item

Test Cases:

Test Case ID: TC1

Description: Test if title appears when the application is running

Input: run application and check to ensure that the title appears

Expected result: a title should appear while the web application is running.

Level of testing: System

Module(s) to be tested: Application Title

Test Case Technique Used: Positive Testing

Test Case ID: TC2

Description: Test that the user is able to view values from a drop-down menu.

Input: User should click on a drop-down menu.

Expected Output: none

Expected Result: A valid list of cart items should be displayed when the user selects the drop-down menu.

Level of testing: System

Module(s) to be tested: Drop Down menu

Test Case Technique used: Use Case Scenarios

Test Case ID: TC3

Description: Test if the user is able to add selected item to the cart via the ‘add to cart’ button

Input: have a user click the ‘add to cart' button

Expected Output: a list of products. User should than select Banana

Expected Result: Banana is added to the cart.

Level of testing: intergration

Module(s) to be tested: Add to cart button

Test Case Technique used: Functional testing

Test Case ID: TC4

Description: Test if the user is able to enter a discount code in text box

Input: enter “DISCOUNT20”(Case matters) discount code into discount code text box.

Expected Output:

Expected Result: Discount code should be accepted.

Level of testing: Integration

Module(s) to be tested: Discount code text box

Test Case Technique used: Functional testing

Test Case ID: TC5

Description: Verify the discount code was applied

Input: After user enters “DISCOUNT20” into the text box, user should select the apply discount button.

Expected Output: new price at 20% lower

Expected Result: discount code gets applied and is reflected in the price

Level of testing: Integration and Validation

Module(s) to be tested: discount code button

Test Case Technique used: Positive testing

Test Case ID: TC6

Description: Verify that the user is able to remove an item

Input: user selects Banana from the drop-down menu and clicks on remove button.

Expected Output: cart items minus removed item

Expected Result: item is removed from the cart

Level of testing: Integration and Validation

Module(s) to be tested: Remove Item button

Test Case Technique used: Positive testing

Test Case ID: TC7

Description: Verify that an error message is generated for trying to remove an invalid item.

Input: User selects item to remove from drop down and clicks remove button

Expected Output: Error message saying that this item does not exist.

Expected Result: error message generated.

Level of testing: Validation

Module(s) to be tested: remove item button

Test Case Technique used: Negative testing

Assumptions and Dependencies:

* The SRS, HLD, and LLD accurately represent the software requirements and it’s design.
* Access to text entry fields and drop-down menus.
* Adequate test data provided.
* The Low-Level Design (LLD) implementation aligns with the design plans.

This test plan provides test cases for validating that a title appears, drop-down values are available, ability to add and delete items from cart, and testing of discount button as well as testing invalid item removal.