Test Cases

Unit Test Cases

Unit Testing 1: Verify adding a product.

Testing Objective: To ensure that product is added successfully.

Test Case Id: TC_001

No.	Test case / Test script	Attribute and value	Expected result	Actual Result	Result
1	Verify that a product is added successfully.	Product Name = 'Bleu De Chanel' , Description= ' Eau de Parfum', Price = 90	Product should be added successfully.	Product should be added successfully.	Pass
2	Verify enabling/disabling "Add Product" button.	Input fields filled/not filled.	"Add Product" button enabled/disabled accordingly.	"Add Product" button enabled/disabled accordingly.	Pass
3	Verify alert message on successful addition.	Single Tap on Add Product button.	Alert message displayed on success	Alert message displayed on success	Pass
4	Verify clearing input fields after adding a product	Product added successfully.	Product name, description, and price fields cleared	Product name, description, and price fields cleared	Pass

Unit Testing 2: View Products

Testing Objective: To ensure that products are displayed.

Test Case Id: TC_002

No.	Test case / Test script	Attribute and value	Expected result	Actual Result	Result
1	Verify that products are displayed.	[{ Product Name: "Bleu De Chanel", Price: $90 }, { Product Name: "Coco Mademoiselle", Price: $120 }]	Products are being displayed.	Products are being displayed.	Pass

Unit Testing 3: View Product Details

Testing Objective: To ensure that product details are displayed.

Test Case Id: TC_003

No.	Test case / Test script	Attribute and value	Expected result	Actual Result	Result
1	Verify navigation to product details screen	Product is pressed.	Navigation to product details screen should occur.	Navigation to product details screen will occur.	Pass
2	Verify that product details are displayed.	Product Name: "Bleu De Chanel", Description: "Eau de Parfum", Price: $90	Product details are being displayed.	Product details are being displayed.	Pass
3	Verify navigation to previous screen	"Go Back" button is pressed.	Navigation to previous screen should occur.	Navigation to previous screen will occur.	Pass

Unit Testing 4: Add to Cart

Testing Objective: To verify that product is added to cart.

Test Case Id: TC_004

No.	Test case / Test script	Attribute and value	Expected result	Actual Result	Result
1	Verify adding product to cart successfully.	'Add To Cart' button is pressed.	Product should be added to the cart successfully.	Product should be added to the cart successfully.	Pass

2	Verify handling duplicate product in cart.	'Add To Cart' button is pressed.	Alert message should be displayed indicating item is already in cart.	Alert message should be displayed indicating item is already in cart.	Pass

Unit Testing 5: Delete Product

Testing Objective: To ensure that a product is deleted.

Test Case Id: TC_005

No.	Test case / Test script	Attribute and value	Expected result	Actual Result	Result
1	The user navigates to View Product	Single Tap	View product screen is displayed.	View product screen is displayed.	Pass
2	The user clicks on the delete icon for the desired product to be deleted.	Single Tap	Product is deleted successfully.	Product is deleted successfully.	Pass

Unit Testing 6: View Cart

Testing Objective: To ensure items added to cart are fetched.

Test Case Id: TC_006

No.	Test case / Test script	Attribute and value	Expected result	Actual Result	Result
1	Verify fetching cart items successfully	Product Name: "Bleu De Chanel", Price: $90, Quantity: 1	Cart items should be fetched and displayed	Cart items are fetched and displayed.	Pass
2	Verify calculating total price of products in cart.	Total: $250	Total price of all items in the cart should be calculated.	Total price of all items in the cart will be calculated.	Pass
3	Verify increasing quantity of an item in the cart	'+' icon is pressed.	Quantity of the item in the cart should increase.	Quantity of the item in the cart will increase.	Pass
4	Verify price is increased when quantity is increased.	'+' icon is pressed.	Price shall increase when quantity increases.	Price will increase when quantity increases.	Pass

5	Verify decreasing quantity of an item in the cart	'-' icon is pressed.	Quantity of the item in the cart should decrease.	Quantity of the item in the cart will decrease.	Pass
6	Verify price is decreased when quantity is decreased.	'-' icon is pressed.	Price shall decrease when quantity decreases.	Price will decrease when quantity decreases.	Pass

Unit Testing 7: Delete Order in Cart

Testing Objective: To ensure that all items in the cart are deleted.

Test Case Id: TC_007

No.	Test case / Test script	Attribute and value	Expected result	Actual Result	Result
1	Verify deleting all items from the cart	"Delete Order" button pressed.	All items in the cart should be deleted.	All items in the cart will be deleted.	Pass

Unit Testing 8: Checkout in Cart

Testing Objective: To ensure that order is placed successfully.

Test Case Id: TC_008

No.	Test case / Test script	Attribute and value	Expected result	Actual Result	Result
1	Verify checkout process.	"Checkout" button pressed.	Alert message should be displayed with order placed message.	Alert message will be displayed with order placed message.	Pass
2.	Verify cart is cleared after checkout process.	"Checkout" button pressed.	All items in the cart shall be cleared after checkout.	All items in the cart are cleared after checkout.	Pass

Integration Testing

No.	Test case / Test script	Attribute and value	Expected result	Actual Result	Result
1	To verify integration of ‘Add Product’ with Products database.	“Add Product’’ button pressed.	Product details are sent to the backend, a new entry is created in the database and the ‘View Product’ list is updated to include the new product.	Product details are sent to back end, a new entry is created in the database and the ‘View Product’ list is updated to include the new product.	Pass
2	To verify products are being fetched from the back end in the product list.	Navigate to ‘View Product’	After adding a product, it should be fetched and displayed in the product list.	After adding a product, it will be fetched and displayed in the product list.	Pass
3	To verify that product is added to cart.	Product Name: "Bleu De Chanel", Description: "Eau de Parfum", Price: $90 Button: ‘Add to Cart’	The product should be added to the cart.	The product will be added to the cart.	Pass

4	To verify the calculation of the total price in the cart.	[{Product Name: "Bleu De Chanel", Price: $90, Quantity: 2 }, { Product Name: "Coco Mademoiselle", Price: $120, Quantity: 1 }] Total Price: $300	The total price in the cart should be calculated correctly based on the quantity of each item.	The total price in the cart is calculated correctly.	Pass
5	To verify the total price updates based on quantity changes.	Product Name: "Bleu De Chanel", Price: $90, Quantity: 2, Quantity decreased:1 Product Name: "Coco Mademoiselle", Price: $120, Quantity: 1, Quantity increased: 3 Total Price: $450	The total price in the cart should reflect the updated quantity of the product, recalculating the total price accordingly.	The total price in the cart is recalculated to reflect the change.	Pass
6	To verify removal of products from cart.	“Delete Order” button pressed.	The front end sends a delete request to backend which removes all the items from the cart in the database and ‘view cart’ is updated.	The front end sends a delete request to backend which removes all the items from the cart in the	Pass

				database and 'view cart' is updated.	
7	To verify cart persistence integration after user navigates away from cart.	Add products to cart, navigate away and then return to cart.	The front-end requests cart state upon returning, the back end provides the current cart state, and the cart displays the previously added items.	The front-end requests cart state upon returning, the back end provides the current cart state, and the cart displays the previously added items.	Pass
8	To verify error handling integration when adding a duplicate product.	Attempt to add a product with a name that already exists.	The backend checks for duplicates and sends an error response, which the front end displays to the user.	The backend checks for duplicates and sends an error response, which the front end displays to the user.	Pass