Comsats University, Islamabad, Attock Campus

Department Of Computer Science

Assignment: 01

Subject:	**MAD**
Program:	**BSSE-VI**
Submitted By:	**Maryum**
Submitted To:	**Sir Kamran**
Reg N0:	**SP22-BSE-051**
Date:	**26 September,2024**

“JAVASCRIPT SHOPPING CART FEATURES USING ES6 FUNCTIONS AND ARRAY METHODS”

Introduction:

The objective of this assignment was to implement a JavaScript-based shopping cart feature that allows for the addition, removal, and updating of items in the cart. It also calculates the total cost of the items, displays a detailed cart summary, and applies discount codes. The code utilizes ES6 features such as arrow functions, array methods (map, filter, reduce), and object manipulate objects in the cart.

Code Explanation:

1. Shopping Cart Array:
   The **cart** array is initialized as an empty array that will hold the items added to the cart. Each item in the cart is represented as an object with properties like **productId**, **productName**, **quantity**, **price, category**, **brand,** and **description**.

2. Add Items to the Cart:
   The **addItemToCart** function is responsible for adding a new product to the cart. It takes in product details as arguments and pushes an object containing those details into the **cart** array using the **push** method.

   addItemToCart(1, "Foundation", 1, 1200, "Cosmetics", "Missrose", "Matte waterproof high coverage foundation for fair complexion");

3. Remove Items from the Cart:
   The **removeItemFromCart** function allows removal of items by **productId**. The **findIndex** method is used to locate the product in the **cart** array, and if found, it is removed using the splice method.

4. Update Item Quantity:
   The **updateItemQuantity** function is used to modify the quantity of an existing product in the cart. It uses the **map** method to check for the product with the matching **productId** and updates its **quantity**.

   updateItemQuantity(2, 3); // Updates the quantity of the item with productId 2

5. Calculate Total Cost:

The **calculateTotalCost** function calculates the total price of all items in the cart. It uses the **reduce** method to accumulate the total cost by multiplying the price by the quantity for each item and summing them up.

6. Display Cart Summary:
7. The **displayCartSummary** function creates a summary of the cart, displaying key details like product name, quantity, total price, category, brand, and description for each item. It uses the **map** method to iterate over the **cart** array and display the details for each product. The **filter** method is used to remove items with zero quantity.

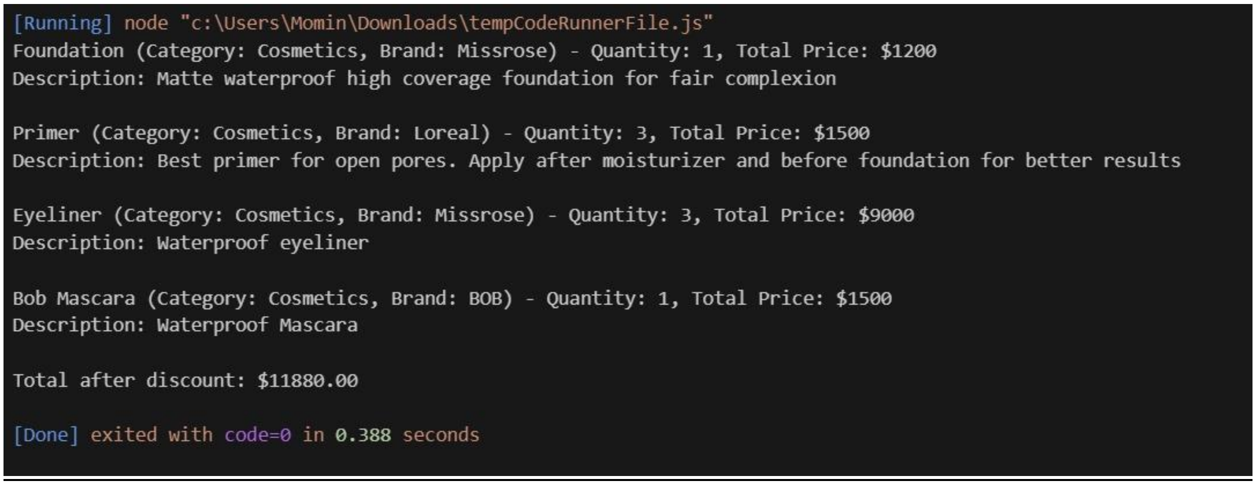
displayCartSummary();

8. Apply Discount Code:

The **applyDiscoun**t function checks if a discount code has been applied (e.g., "DISCOUNT10") and calculates the discounted total cost. It uses a simple condition to apply a 10% discount and then displays the total after discount.

applyDiscount("DISCOUNT10"); // Applies a 10% discount to the total cost

OUTPUT:



```
[Running] node "c:\Users\Momin\Downloads\tempCodeRunnerFile.js"
Foundation (Category: Cosmetics, Brand: Missrose) - Quantity: 1, Total Price: $1200
Description: Matte waterproof high coverage foundation for fair complexion

Primer (Category: Cosmetics, Brand: Loreal) - Quantity: 3, Total Price: $1500
Description: Best primer for open pores. Apply after moisturizer and before foundation for better results

Eyeliner (Category: Cosmetics, Brand: Missrose) - Quantity: 3, Total Price: $9000
Description: Waterproof eyeliner

Bob Mascara (Category: Cosmetics, Brand: BOB) - Quantity: 1, Total Price: $1500
Description: Waterproof Mascara

Total after discount: $11880.00

[Done] exited with code=0 in 0.388 seconds
```

Conclusion:

Through this assignment, I learned how to manipulate arrays and objects in JavaScript to manage the functionality of a shopping cart. I got hands-on experience with using ES6 arrow functions, array methods like **map**, **reduce**, **filter**, and object manipulation. One of the challenges I faced was understanding how to update an item in the cart without modifying the entire array, which was solved using the **map** method. Additionally, implementing the discount feature allowed me to apply conditions effectively in the code. Overall, this assignment enhanced my understanding of JavaScript's modern features and their practical applications in developing interactive components for a shopping cart.