# Report on the Online Shopping Program

## 1. Introduction

The program simulates a simple online shopping system, allowing users to view available products, add products to their cart, view the contents of the cart, and save the order details to a file. The system is designed using C programming, employing basic file operations and structures to represent products and the shopping cart.

## 2. Program Structure

The program utilizes various C constructs including functions, loops, conditionals, structures, and file handling. Below are the key components of the program:

Structures:

Product: Holds the details of a product, including id, name, and price.

CartItem: Holds information about an item in the shopping cart, including the associated product and quantity.

Global Variables:

products[]: An array holding the available products in the store.

cart[]: An array holding the items added to the shopping cart.

cartSize: A variable to keep track of the number of items in the cart.

Functions:

initializeProducts(): Initializes the products[] array with predefined product information.

displayProducts(): Displays the list of available products with their respective prices.

addToCart(): Adds a selected product to the cart, updating the cart with the product and the desired quantity.

viewCart(): Displays the contents of the cart, showing the products, their prices, quantities, and total cost.

saveOrderToFile(): Saves the order details (cart items and total cost) to a text file named order.txt.

menu(): Provides the user with a menu-driven interface to interact with the program, allowing them to view products, add items to the cart, view the cart, or save the order.

## 3. How It Works

The program operates in a loop, continuously presenting the user with a menu of options. The user can choose from the following actions:

View Products: Displays a list of products available for purchase, including their ID, name, and price.

Add Product to Cart: Prompts the user to input the product ID and the quantity of the product they wish to add to their cart. The system verifies the product ID and updates the cart accordingly.

View Cart: Displays the contents of the cart, showing each product, its price, quantity, and the total cost of each item. It also calculates and displays the total cost of all items in the cart.

Save Order: Saves the cart's details to a text file, including the product names, prices, quantities, and the total cost. The file is saved as order.txt.

Exit: Exits the program with a thank-you message.

## 4. Functionality in Detail

Product Initialization: The initializeProducts() function populates the products[] array with five predefined products, each having a unique ID, name, and price.

Adding Products to the Cart: When the user selects the "Add Product to Cart" option, the program prompts the user for a product ID and quantity. If the product exists, it is added to the cart, and the user is notified that the product has been successfully added. If the product ID is invalid, an error message is displayed.

Viewing the Cart: The cart's contents are displayed in a formatted table, showing the product name, price, quantity, and total cost for each item. The total cost of all items is also calculated and displayed.

Saving the Order: The saveOrderToFile() function creates or overwrites a text file named order.txt, saving the details of the current cart. This includes the name, price, quantity, and total cost of each product in the cart, along with the total price of the entire order.

## 5. Program Flow

The program starts by initializing the product list, then enters a loop where it presents a menu of options. Based on user input, the program performs the corresponding action:

1. View Products: Lists all available products.

2. Add Product to Cart: Prompts for the product ID and quantity, and adds the item to the cart.

3. View Cart: Displays the cart with details about the items and the total price.

4. Save Order: Saves the cart contents to a file.

5. Exit: Ends the program.

## 6. Potential Improvements

While the program serves its basic purpose, there are several areas where it can be enhanced:

Error Handling: More robust error handling can be implemented, especially for invalid inputs (e.g., non-numeric entries for product ID or quantity).

Cart Management: The cart currently does not support removing items or modifying quantities once they are added. Implementing features like updating or removing cart items could be a useful addition.

Dynamic Product List: Instead of hardcoding the products in the initializeProducts() function, products could be loaded from an external file, allowing the store's inventory to be easily updated.

User Interface: The text-based interface could be improved with better formatting, more descriptive messages, and input validation to provide a more user-friendly experience.

## 7. Conclusion

This program demonstrates a simple but functional implementation of an online shopping system in C. It allows users to view products, manage a shopping cart, and save their orders to a file. The program showcases basic C features such as structures, file handling, and array manipulation, and provides a solid foundation for building more complex e-commerce systems. Future improvements could include more interactive features, better error handling, and dynamic product management.