.

NAME:ALVIA SAJI

ROLL NO.:11

COURSE NAME:PROGRAMMING IN C

DATE:17-07-2024

SHOPPING STSYEM

Manage Shopping

INTRODUCTION

*Overview of the project****:***

This project aims to develop a basic shopping system in C, allowing users to browse products, add items to their cart, and checkout.

*Problem statement:*

Create a program that simulate product browsing, cart management, and checkout processes.

*Objective:*

Design and implement a C program that provides a user-friendly shopping interface, accurately handling product selection and checkout processes.

System Requirements

- *Hardware requirements:*

Any computer with a C compiler (e.g GCC)

*- Software requirements:*

C programming language, standard library (stdio.h, string.h)

Design and Development

- *Description of the program logic:*

The program uses structs to define products and cart items, with functions for displaying products, adding to the cart, and checking out. The main function handles user input and switches between program modes.

*Pseudocode:*

1. Initialize constants and global variables:

- MAX\_PRODUCTS = 10

- MAX\_CART\_ITEMS = 10

- Product struct: id, name, price

- CartItem struct: productId, quantity

- products[MAX\_PRODUCTS] array of Product structs

- cart[MAX\_CART\_ITEMS] array of CartItem structs

- cartItemCount = 0

- productCount = 3 (initially)

2. Define functions:

- displayProducts()

- Print "Available Products:"

- Loop through products array:

- Print product id, name, price formatted nicely

- End loop

- addToCart()

- Prompt user for productId and quantity

- Loop through products array:

- If productId matches products[i].id:

- Add productId and quantity to cart[cartItemCount]

- Increment cartItemCount

- Print confirmation message

- Exit function

- If productId not found, print "Product not found!"

- checkout()

- Initialize total = 0

- Print "Checkout:"

- Loop through cart array:

- Nested loop through products array:

- If cart[i].productId matches products[j].id:

- Print product name, quantity, price \* quantity

- Add price \* quantity to total

- Print "Total: $" + total formatted to 2 decimal places

- Reset cartItemCount to 0 (clear cart)

3. Main program loop:

- Display menu options:

- "1. Browse Products"

- "2. Add to Cart"

- "3. Checkout"

- "4. Exit"

- Prompt user for choice

- Switch based on choice:

- Case 1: Call displayProducts()

- Case 2: Call addToCart()

- Case 3: Call checkout()

- Case 4: Print "Thank you for shopping!" and exit loop

- Default: Print "Invalid choice! Please try again."

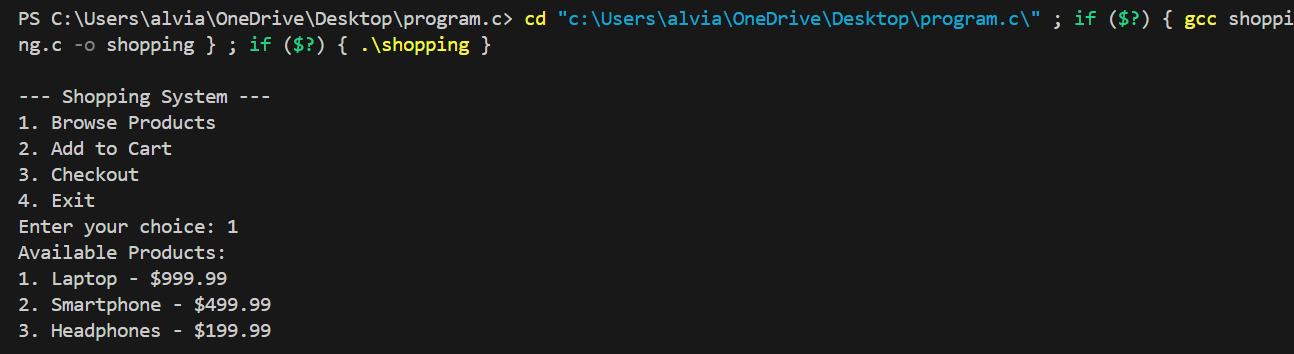
4. End loop when choice is 4

5. Terminate program

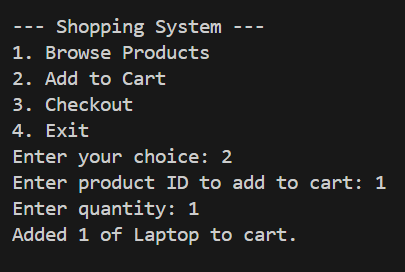
Testing and Results

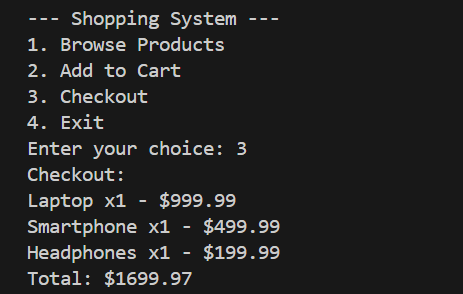
*Test cases:*

*1.*Browse products: Verify all products are displayed correctly

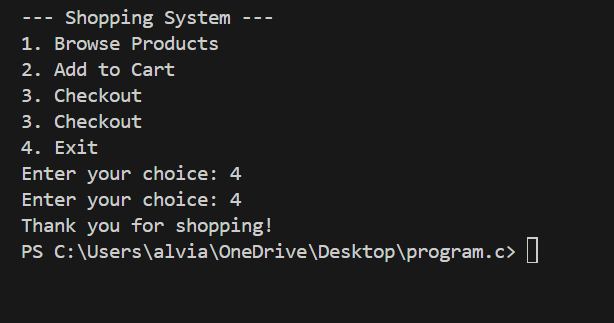
**

*2.*Add to cart: Check if products are successfully added to the cart.

**

*3.Checkout: Confirm total cost calculation*

*4*.Exit the program

**

*Discussion of results:*

*The program successfully implements the shopping system, accurately managing products and cart items.*

*- Future enhancements:*

*Implement additional features like user authentication, product search, and payment processing*

**Conclusion**

*-* ***Summary of the project:***

This project demonstrates a basic shopping system in C, showcasing struct usage, function design, and user input handling.

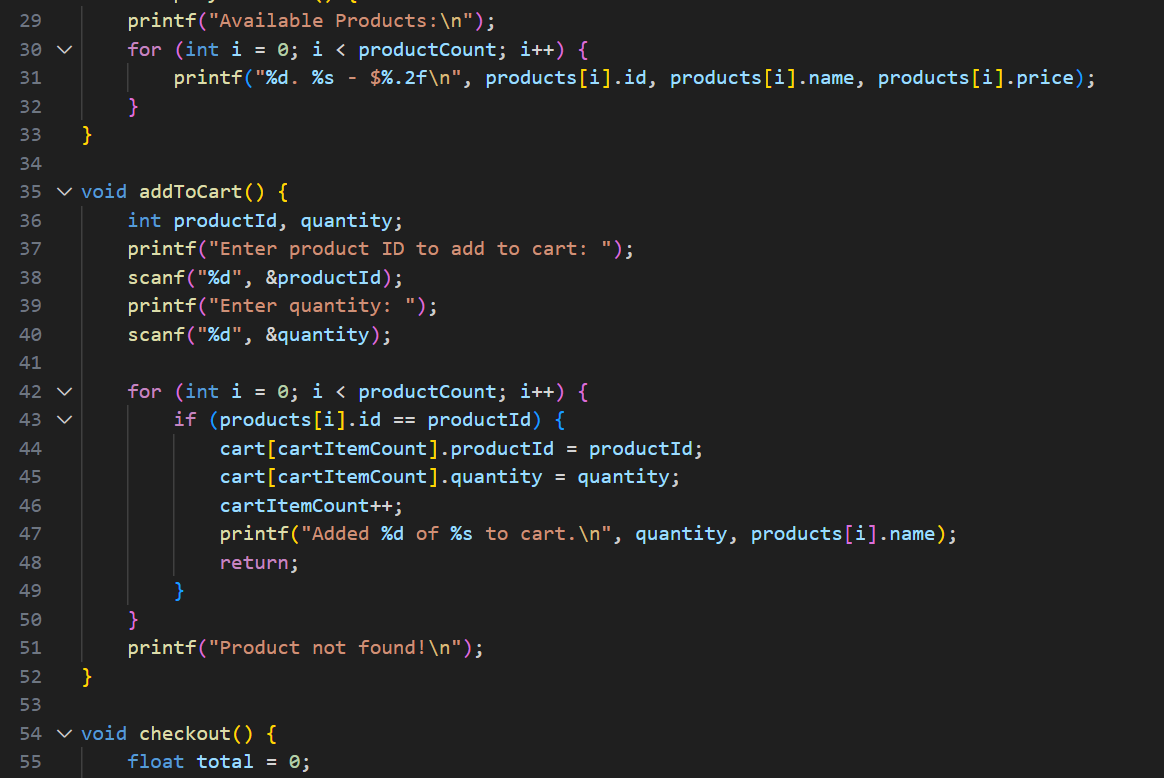
**References**

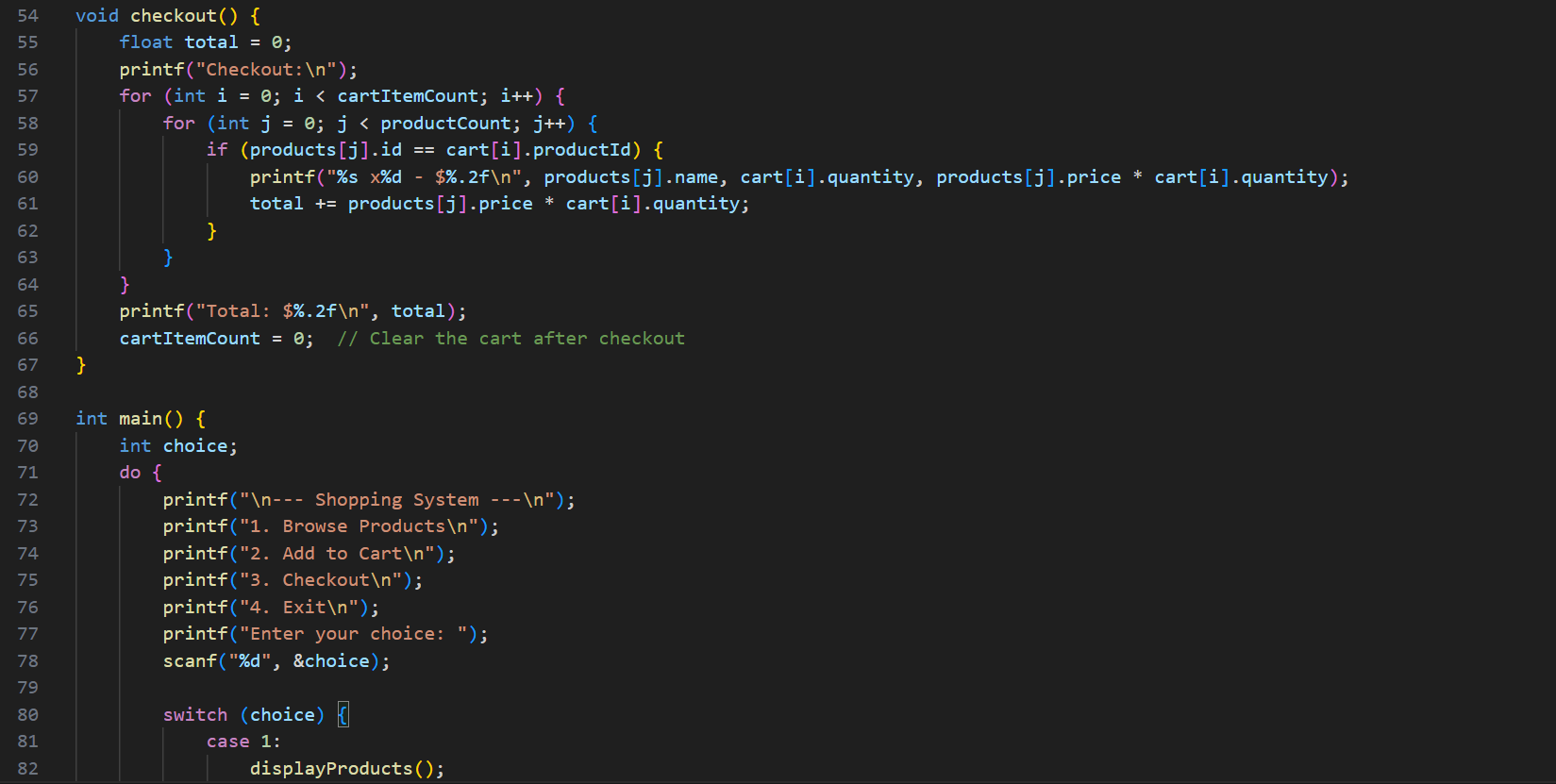
* Online tutorials

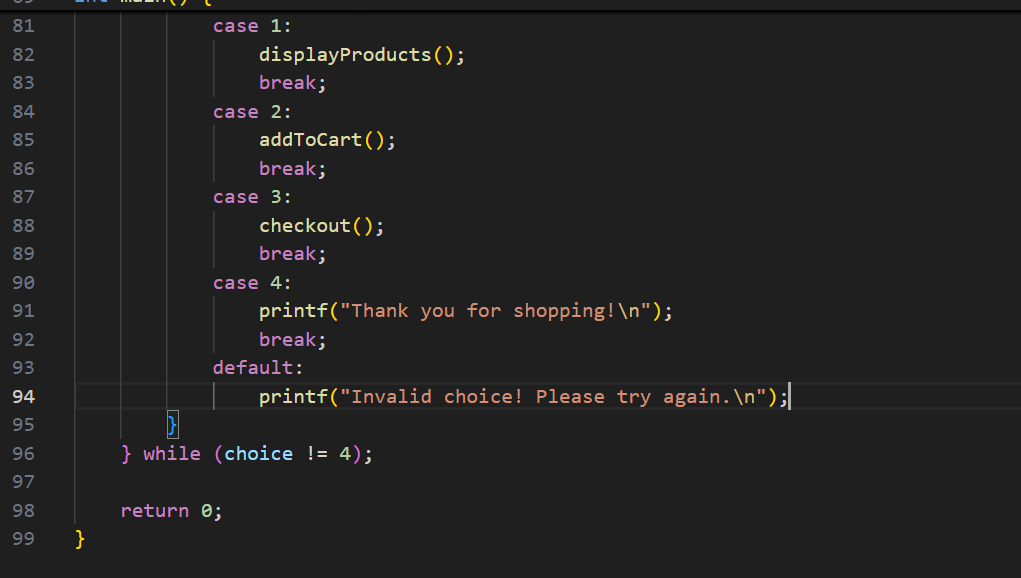
**Appendices**

***Code listing:***

******

******

******

******