PROJECT REPORT

SUBMITTED TO:

Sir Bilal Rehman

PROJECT NAME:

Shopping Cart System

COURSE NAME:

CS-121 Programming Fundamentals

DEPARTMENT:

Software Engineering

BATCH:

SE-24 (fall)

SUBMITTED BY (GROUP):

Muhammad Zia Ur Raheem	**(F24608048)**
Muhammad Hassan Khawaja	(F24608030)
Muhammad Hassan	(F24608010)
Muhammad Rehan Muddassir Ranjha	(F24608041)
Muhammad Abdullah Tahir	(F24608011)
Syed Dawar Raza Kazmi	(F24608058)

PROGRAM

```
#include<iostream>
using namespace std;
const double tax_rate=0.07;
const double disc_rate=0.10;
const int max_item=10;
void addItem(string cart[],double prices[],int &itemcount);
void removeItem(string cart[], double prices[], int &itemcount);
void displayCart(string cart[], double prices[], int itemcount);
double calculateTotal(double prices[], int itemCount);
double applyDiscount(double total);
double applyTax(double total);
int main()
{   string cart[max_item];
double prices[max_item];
int itemcount=0;
int choice;
cout<<"\t\t_____"<<endl;
cout<<"\t\t**********************************************************"<<endl;
cout<<"\t\t         SHOPPING CART SYSTEM              "<<endl;
cout<<"\t\t**********************************************************"<<endl;
cout<<"\t\t|_____| "<<endl;
cout<<"\t\t|                 **** MENU****              | "<<endl;
cout<<"\t\t|              1)Add Item                    | "<<endl;
cout<<"\t\t|              2)Remove Item                 | "<<endl;
cout<<"\t\t|              3)Display Cart                | "<<endl;
cout<<"\t\t|              4)Calculate Total and Checkout | "<<endl;
cout<<"\t\t|            5)Exit                          | "<<endl;
```

```
while(choice!=5)
        {
        cout<<"\n\n";
        cout<<"\t\t\      Enter your choice :";
        cin>>choice;
        switch(choice)
        {
        case 1:
                addItem(cart, prices, itemcount);
                break;
        case 2:
        removeItem(cart, prices, itemcount);
break;
        case 3:
                displayCart(cart, prices, itemcount);
break;
       case 4:
             {
double total = calculateTotal(prices, itemcount);
total = applyDiscount(total);
total = applyTax(total);
cout<< "Total price after discount and tax: $" << total << endl;
break;
      }
        case 5:
        cout<< "Thank you for shopping! Goodbye!\n";
        break;
default:
cout<< "Invalid choice. Please try again.\n";
        }
   }
return 0;
}
```

CONTRIBUTIONS

```
void addItem(string cart[],double prices[],int &itemcount)
 {
if (itemcount < max_item) {
string item;
double price;
cout<< "Enter the item name: ";
cin.ignore();
getline(cin, item);
cout<< "Enter the item price: $";
cin>> price;
cart[itemcount] = item;
prices[itemcount] = price;
itemcount++;
cout<< item << " has been added to your cart.\n";
   } else {
cout<< "Your cart is full! Cannot add more items.\n";
```

Muhammad Zia Ur Raheem

```
void removeItem(string cart[], double prices[], int &itemcount)
  {
string itemToRemove;
bool itemFound = false;
cout<< "Enter the name of the item to remove: ";
cin.ignore();
getline(cin, itemToRemove);
for (int i = 0; i < itemcount; i++) {
if (cart[i] == itemToRemove) {
for (int j = i; j < itemcount - 1; j++) {
cart[j] = cart[j + 1];
prices[j] = prices[j + 1];
          }
itemcount--;
itemFound = true;
cout<< itemToRemove << " has been removed from your cart.\n";
break;
       }
   }
if (!itemFound) {
cout<< "Item not found in the cart.\n";
   }
}
```

Muhammad Hassan Khawaja

```
void displayCart(string cart[], double prices[], int itemcount)
 {
        if (itemcount == 0) {
cout<< "Your cart is empty.\n";
   } else {
cout<< "\n--- YOUR SHOPPING CART ---\n";
for (int i = 0; i < itemcount; i++) {
cout<< cart[i] << " - $" << prices[i] << endl;
      }
   }
 }
```

Muhammad Rehan Muddassir Ranjha

```
double calculateTotal(double prices[], int itemcount)
 {
double total = 0.0;
for (int i = 0; i < itemcount; i++) {
total += prices[i];
   }
return total;
}
```

Muhammad Hassan

```
double applyDiscount(double total) {
double discount = total * disc_rate;
double discountedTotal = total - discount;
cout<< "Discount applied: -$" << discount << endl;
return discountedTotal;
}
```

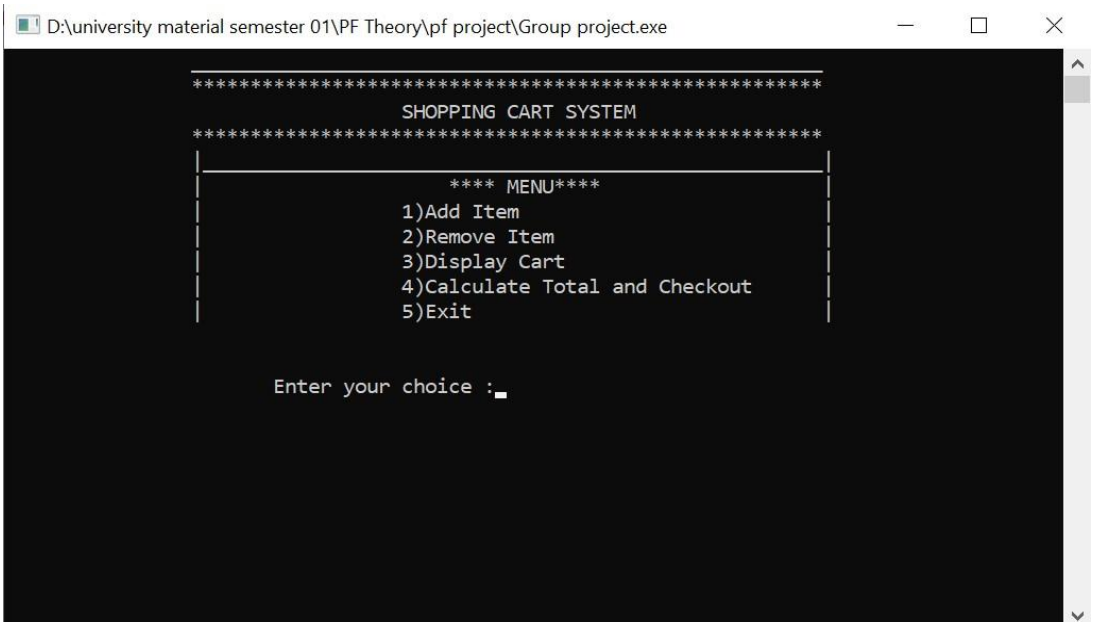
Muhammad Abdullah Tahir

```
double applyTax(double total) {
double tax = total * tax_rate;
double finalTotal = total + tax;
cout<< "Tax applied: +$" << tax << endl;
return finalTotal;
}
```

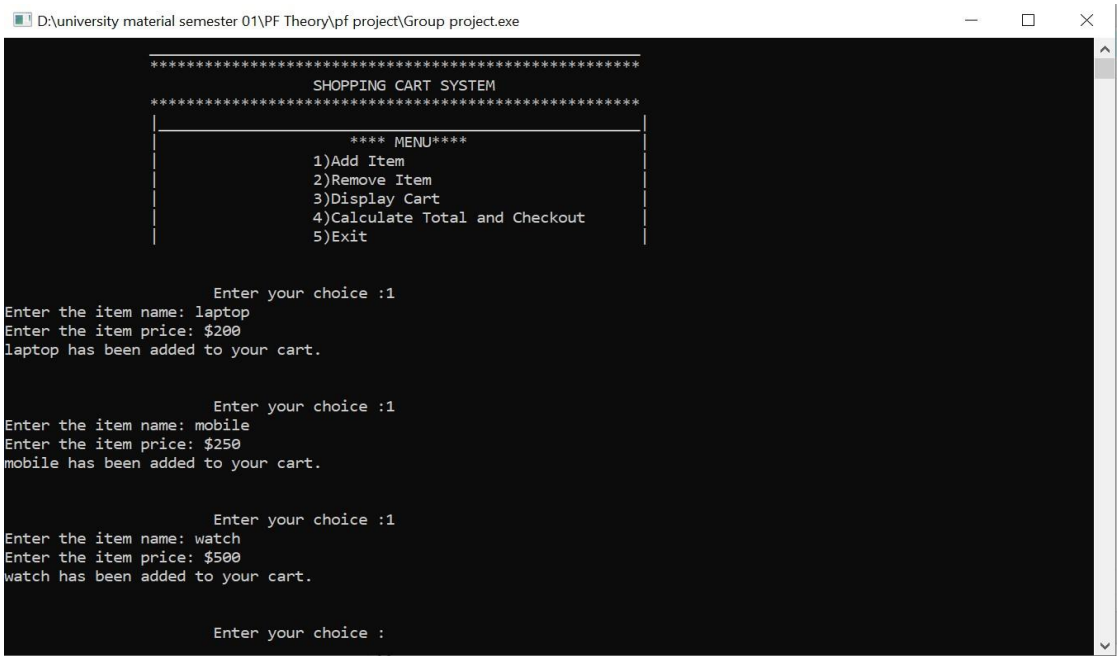
Syed Dawar Raza Kazmi

OUTPUTS

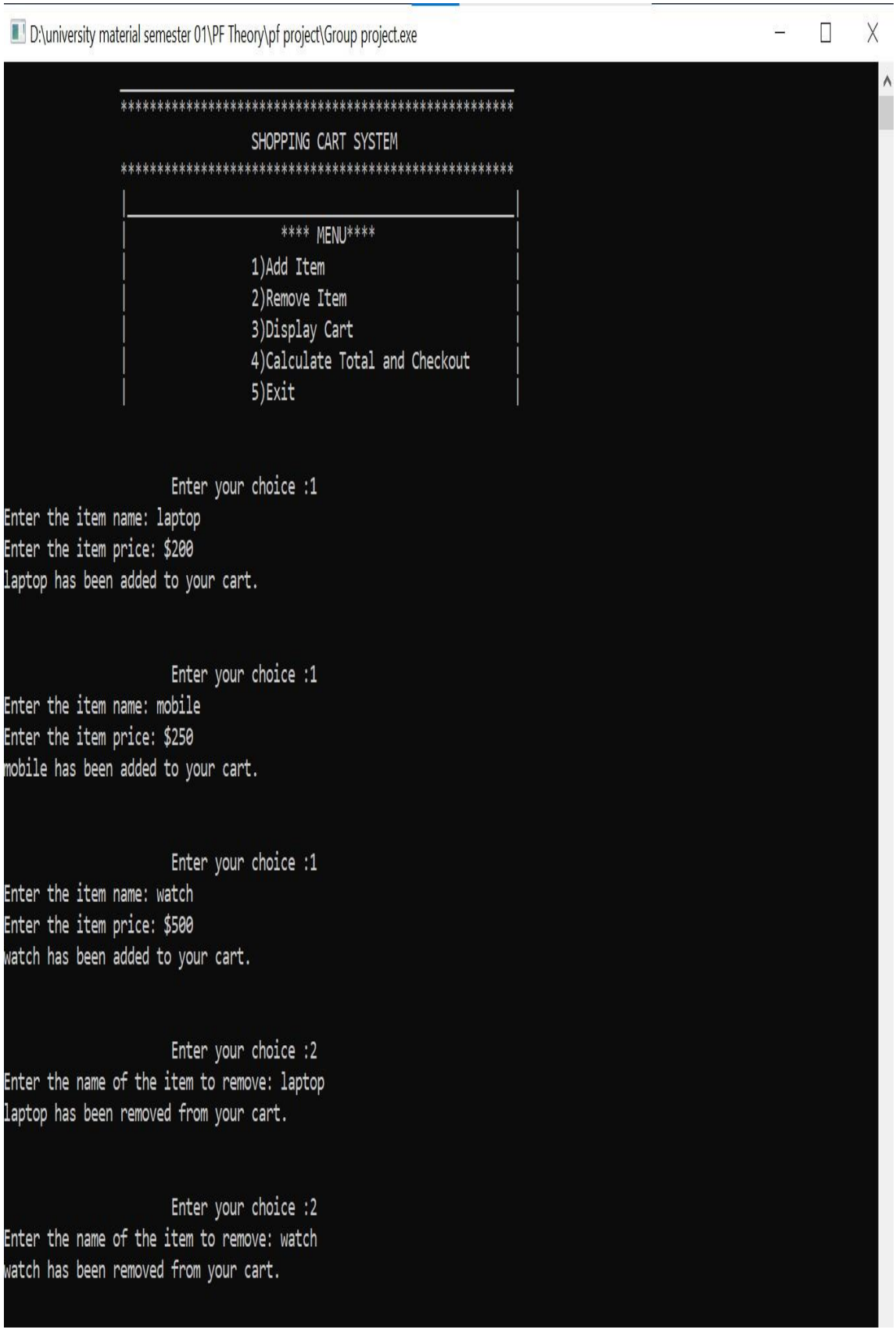
USER INTERFACE:



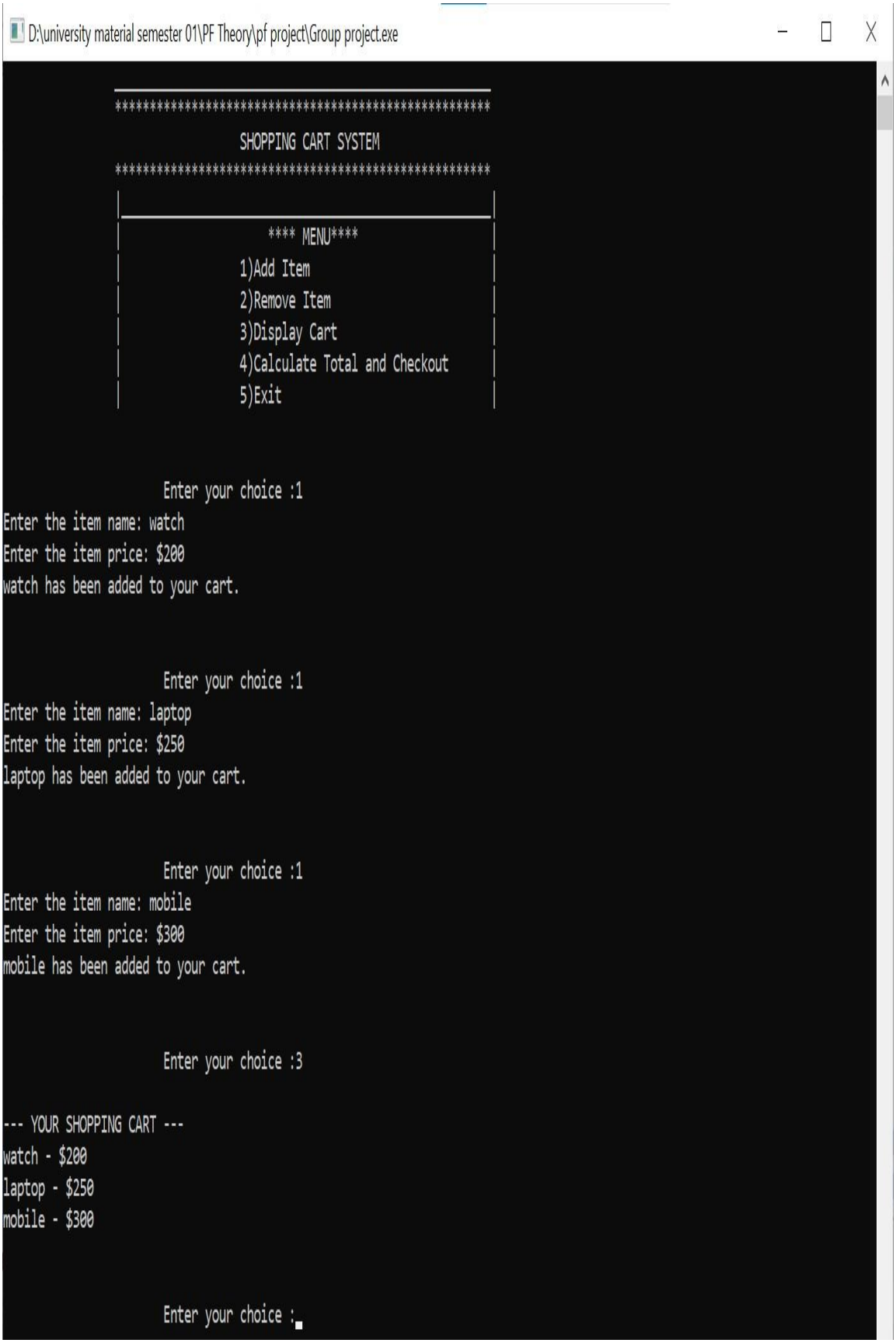
void addItem(string cart[],double prices[],int &itemcount);



void removeItem(string cart[], double prices[], int &itemcount);



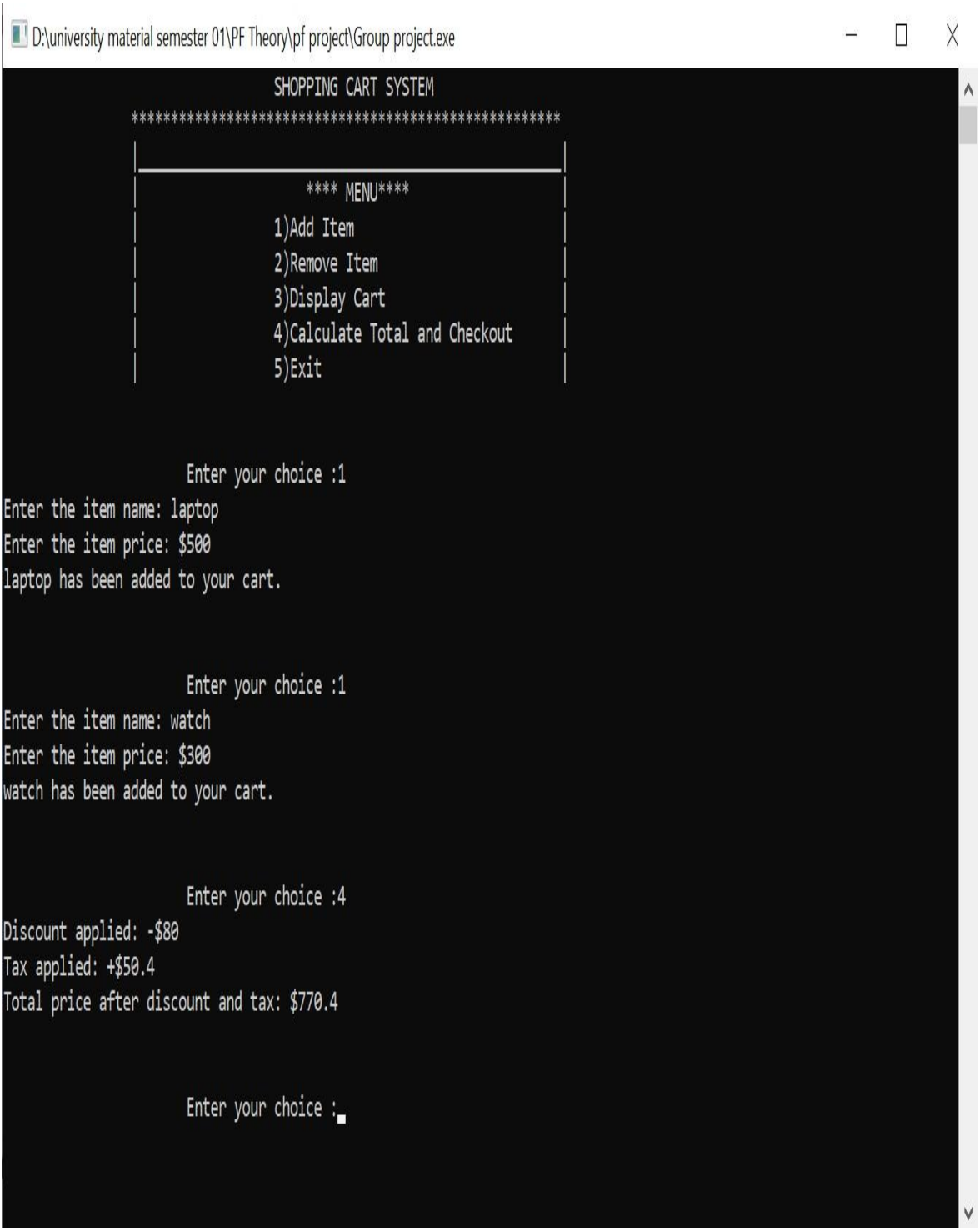
void displayCart(string cart[], double prices[], int itemcount);



double calculateTotal(double prices[], int itemCount);

double applyDiscount(double total);

double applyTax(double total);



DESCRIPTION

MAIN FUNCTIONALITY

Menu Navigation:
A user-friendly menu system allows users to select various options like adding items, removing items, displaying the cart, or calculating the total.

Cart Management:
Users can maintain a shopping cart with items, prices, and update it dynamically.

Checkout:
The program calculates the total, applies a discount, and then calculates the tax to provide the final total cost.

Exit:
Safely exits the program with a farewell message.

COMPONENTS AND THEIR ROLES

Global Constants

- **tax_rate and disc_rate:** Define the tax rate (7%) and discount rate (10%) applied during checkout.
- **max_item:** Restricts the cart to a maximum of 10 items for simplicity.

Functions

(a) addItem

- Accepts an item name and price from the user.
- Adds the item and its price to the respective arrays (cart and prices).
- Increments the itemcount.

(b) removeItem

- Allows the user to remove an item by name.
- If found, shifts the remaining elements in the arrays leftward to maintain order.
- Decrements the itemcount.

displayCart

- Prints the contents of the cart if not empty, showing items with their corresponding prices.
- Displays a message if the cart is empty.

(d) calculateTotal

- Iterates through the prices array to sum up the prices of all items in the cart.
- Returns the total before discount and tax.

(e) applyDiscount

- Calculates a discount based on the total amount (total * disc_rate).
- Prints the discount applied and returns the total after discount.

(f) applyTax

- Calculates tax based on the total amount (total * tax_rate).
- Prints the tax applied and returns the final total after tax.

PROGRAM FLOW

Initialization:	Menu Options:	Interactivity:	Error Handling:
Arrays cart and prices store item names and prices, and itemcount tracks the total items.	Displays options continuously until the user selects option 5 (Exit).	The user can add up to max_item items, remove specific items, view their cart, and compute final totals with a breakdown of discount and tax.	Ensures appropriate messages for invalid input, full cart, or attempting to remove non-existent items.

CHALLENGE AND SOLUTION

Challenge: The system had to dynamically handle adding and removing items while keeping the cart organized. Shifting elements when removing an item could have been prone to errors (like array overflows or data misalignment).

Resolution: Used a **for-loop** to shift the elements after removing an item, ensuring that the array remained contiguous. Thorough testing was done for edge cases like removing the first or last item in the cart.