Project Report

On

Online Shopping Cart System

Submitted by	Supervised by
Bhuban Chakma **IT-23054**	**Dr. Ziaur Rahman** **Associate professor** **Department of ICT, MBSTU**

DEPARTMENT OF INFORMATION AND COMMUNICATION TECHNOLOGY

MAWLANA BHASHANI SCIENCE AND TECHNOLOGY UNIVERSITY

SANTOSH, TANGAIL-1902, DHAKA, BANGLADESH.

Acknowledgements

I would like to express my deep sense of gratitude and convey thanks to everyone who helped me and supported during the completion of this project. First, I would like to express a deep sense of gratitude to my supervisor Dr. Ziaur Rahman, Associate professor, Dept. of Information and Communication Technology, MBSTU for helping, guiding, and supporting me throughout the project. I acknowledge my department for providing the courses and a great atmosphere that helped complete different chapters of this paper. I would also like to thank my friends, without whom this project would have been impossible.

Table of contents

Page no

Chapter 1

1.1 -- Introduction of the project

This report details the design and implementation of a basic console-based shopping cart application developed in C. The purpose of the application is to provide users with a simple system for managing shopping transactions, allowing them to view a list of products, add items to a cart, remove items, and view the total cost. The application also supports persistent cart storage, enabling continuity between sessions.

Chapter 2

2.1 -- Objectives of Project

The primary goals of this project are:

- To design a user-friendly shopping cart interface.
- To implement features for adding, viewing, and removing items in a cart.
- To allow users to save and reload cart data across sessions.
- To ensure data security and integrity through proper file handling and input validation.

Chapter 3: Features of the project

3.1- Loading and Displaying Products

- **loadProducts Function**: Reads product details (ID, name, and price) from an external products.txt file.
- **showProducts Function**: Displays the list of available products in a user-friendly format, showing each product's ID, name, and price.

3.2- Cart Management

- **addToCart Function**: Allows users to add products to their cart by entering the product ID and desired quantity.
- **removeFromCart Function**: Users can remove items from their cart by entering the product ID.
- **showCart Function**: Lists all items currently in the cart with quantities, individual costs, and a calculated total cost. Additionally, it includes a personalized message with the user's name.

Chapter 4: Source Code

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>



typedef struct Item {
    int id;
    char name[50];
    float price;

    int quantity;
} Item;

void loadProducts(Item products[], int *count) {
    FILE *file = fopen("products.txt", "r");
    if (file==NULL) {
        printf("Error opening products file.\n");
        return;
    }
    while (fscanf(file, "%d %s %f", &products[*count].id,
products[*count].name, &products[*count].price) != EOF) {
        products[*count].quantity = 0;
        (*count)++;
    }
    fclose(file);
```

```
}

void showProducts(Item products[], int count) {
    printf("\nAvailable Products:\n");
    for (int i = 0; i < count; i++) {
        printf("ID:%d- %s - %.2f\n", products[i].id, products[i].name,
products[i].price);
    }
}

void addToCart(Item cart[], int *cartCount, Item products[], int count) {
    int id, quantity;
    printf("Enter product ID: ");
    scanf("%d", &id);
    printf("Enter quantity: ");
    scanf("%d", &quantity);

    for (int i = 0; i < count; i++) {
        if (products[i].id == id) {
            cart[*cartCount] = products[i];
            cart[*cartCount].quantity = quantity;
            (*cartCount)++;
            printf("Product added to cart.\n");
            return;
        }
    }
    printf("Product not found.\n");
}
```

```
void removeFromCart(Item cart[], int *cartCount) {
    int id, found = 0;
    printf("Enter product ID to remove: ");
    scanf("%d", &id);

    for (int i = 0; i < *cartCount; i++) {
        if (cart[i].id == id) {
            found = 1;
            for (int j = i; j < *cartCount - 1; j++) {
                cart[j] = cart[j + 1];
            }
            (*cartCount)--;
            printf("Product removed from cart.\n");
            break;
        }
    }
    if (!found) {
        printf("Product not found in cart.\n");
    }
}

void showCart(Item cart[], int cartCount,char name[]) {
    float total = 0;
    printf("\nYour Cart:\n");
    if (cartCount == 0) {
        printf("Cart is empty.\n");
```

```
        return;
    }
    for (int i = 0; i < cartCount; i++) {
        printf("%s x%d = %.2f\n", cart[i].name, cart[i].quantity, cart[i].quantity *
cart[i].price);
        total += cart[i].quantity * cart[i].price;
    }
    printf("Dear %s Sir/Madam Your Total Bill : %.2f\n",name,total);
}

void saveCart(Item cart[], int cartCount) {
    FILE *file = fopen("cart.txt", "w");
    for (int i = 0; i < cartCount; i++) {
        fprintf(file, "%d %s %.2f %d\n", cart[i].id, cart[i].name, cart[i].price,
cart[i].quantity);
    }
    fclose(file);
}

void loadCart(Item cart[], int *cartCount) {
    FILE *file = fopen("cart.txt", "r");
    if (file==NULL) {
        return;
    }
    while (fscanf(file, "%d %s %f %d", &cart[*cartCount].id,
cart[*cartCount].name, &cart[*cartCount].price, &cart[*cartCount].quantity) !=
EOF) {
        (*cartCount)++;
    }
```

```
    fclose(file);
}

int main() {
    Item products[100];
    Item cart[100];
    char name[30],email[100],contact[20];
    int productCount = 0, cartCount = 0;
    int choice;
    printf("Please Enter Your Name:\n");
    fgets(name,30,stdin);
    printf("Please Enter Your Email Address:\n");
    gets(email);
    printf("Please Enter Your Contact No. :\n");
    gets(contact);

    loadProducts(products, &productCount);
    loadCart(cart, &cartCount);

    do {
        printf("\nMenu:\n");
        printf("1. Show Products\n2. Add to Cart\n3. Remove from Cart\n4. Show
Cart\n5. Exit\n");
        printf("Enter your choice: ");
        scanf("%d", &choice);

        switch (choice) {
            case 1:
```

```
            showProducts(products, productCount);
            break;
        case 2:
            addToCart(cart, &cartCount, products, productCount);
            break;
        case 3:
            removeFromCart(cart, &cartCount);
            break;
        case 4:
            showCart(cart, cartCount,name);
            break;
        case 5:
            saveCart(cart, cartCount);
            printf("Cart saved. Exiting...\n");
            break;
        default:
            printf("Invalid choice. Please try again.\n");
    }
  } while (choice != 5);

  return 0;
}
```

Chapter 5

5.1 -- Functions

The code is organized into the following core functions:

- Product Functions (loadProducts, showProducts): Handle loading and displaying available products.
- Cart Functions (addToCart, removeFromCart, showCart): Provide cart management features.
- File Handling Functions (saveCart, loadCart): Enable saving and reloading of the cart data.
- User Input: Collects and stores user data (name, email, contact number) for personalization.

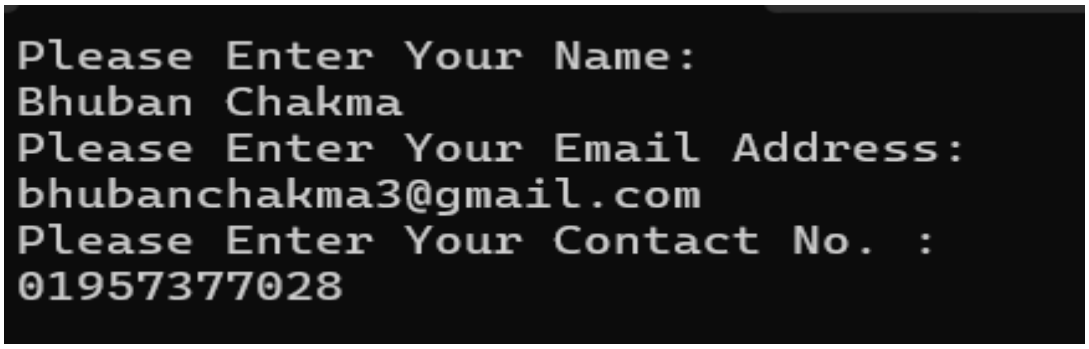
5.2 -- Sample Workflow

Upon starting, the user is prompted to enter their name, email, and contact number. The main menu displays options to view products, add to cart, remove from cart, view the cart, or exit the program. The program workflow is as follows:

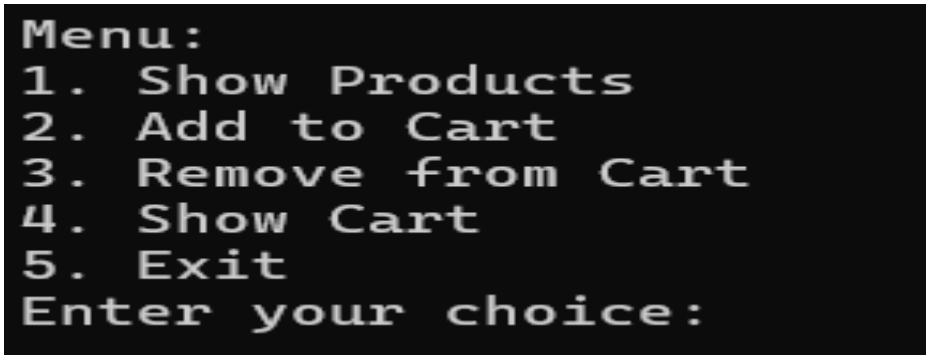
1. Show Products: Lists all available products from products.txt.
2. Add to Cart: Prompts for the product ID and quantity, adding the selected item to the cart.
3. Remove from Cart: Prompts for the product ID, removes the item if found in the cart.
4. Show Cart: Displays all items in the cart, the total price, and a personalized greeting.
5. Exit: Saves the cart to cart.txt and exits the program.

Chapter 6: Output

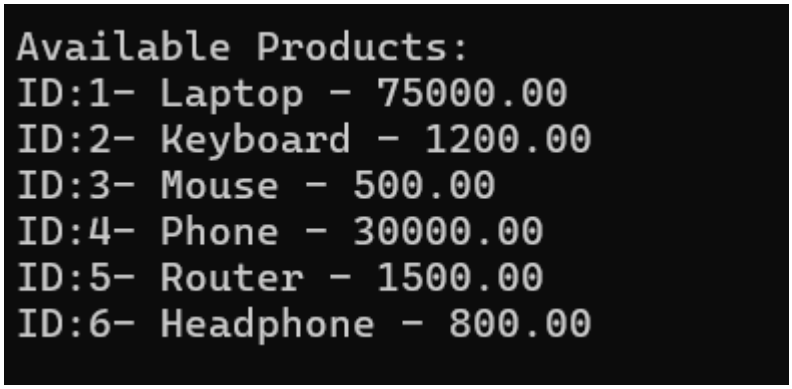
6.1: Screenshot of user input data



6.2: Screenshot of menu:



6.3: Screenshot of available products:



6.4: Screenshot of add to cart menu:

```
Enter your choice: 2
Enter product ID: 1
Enter quantity: 2
Product added to cart.
```

6.5: Screenshot of remove from cart menu:

```
Enter your choice: 3
Enter product ID to remove: 1
Product removed from cart.
```

6.6: Screenshot of show cart menu:

```
Your Cart:
Phone x5 = 150000.00
Laptop x2 = 150000.00
Dear Bhuban Chakma
 Sir/Madam Your Total Bill : 300000.00
```

Chapter 7

7.1 -- Limitations and areas for improvement

- Product Name Formatting: Current code assumes product names are single words (no spaces). This could be improved to handle multi-word product names.

- Input Validation: Additional validation could be added for numeric inputs to prevent entry errors.

- Data Encryption: Encrypting sensitive data such as the user's name and contact information before saving to files could enhance security**.**

Chapter 8 : Conclusion

This shopping cart program meets its primary objectives of providing a basic but functional shopping cart system with options to view products, manage a cart, and persist data across sessions. The code is structured in a modular manner, enhancing readability and maintainability. With further refinements, including enhanced input validation and user interface improvements, this program could be developed into a more robust solution.