Welcome To My Presentation

Online Shopping Cart System

Presented By	Supervised By
Bhuban Chakma ID: IT-23054 1st Year 2nd Semester Session:2022-23 Dept. Of ICT,MBSTU	Supervised By Dr. Ziaur Rahman Associate professor Dept. Of ICT , MBSTU

Course Title:Project-1

Course Code:ICT1200

Introduction

- **Purpose**:
  - To build a simple console-based shopping cart application for product management.
  - Allows users to view, add, remove items in a cart, and save data.
- **Language**: C programming
- **Key Features**:
  - Product loading from files, cart management, data persistence

Objectives

- **Provide a User-Friendly Interface**
- **Enable Cart Management**: Adding, removing items
- **Support Persistent Cart Storage**: Save cart to file for session continuity
- **Ensure Data Integrity**: File handling, user input validation

Features Overview

Product Loading

Products are loaded from a products.txt products.txt file and displayed with ID, ID, name, and price.

Cart Management

Users can add items, view the cart, and and remove items from the cart.

Data Persistence

Cart data is saved to cart.txt, allowing allowing reloading of saved data.

User Personalization

User Details

Program prompts for user's name, email, and contact number.

Enhanced Experience

Personalization in cart and bill display.

Example

"Dear [User's Name], your total bill is..."

Code Structure

Data Structure

Item structure with id, name, price, and quantity fields.

Key Functions

Product, cart, and file handling functions implemented.

File Operations

Reads products from products.txt and saves/loads cart data

data from cart.txt.

Sample Workflow

User Data Collection

Collect name, email, and contact number.

Main Menu

Show products, add/remove from cart, view cart, exit (save (save cart).

Cart Display

Show personalized message with total amount.

Key Code Snippets

Loading Products from File:

```
void loadProducts(Item products[], int *count) {
    FILE *file = fopen("products.txt", "r");
    // Error handling and reading product data
}
```

Key Code Snippets

Adding Item to Cart:

```
void addToCart(Item cart[], int *cartCount, Item products[], int
count) {
    // Prompts for ID and quantity, adds to cart if valid
}
```

Source Code

Main Function:

```c
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);
```

Main Function

```
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;
}
```

Load From File Function

```
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);
}
```

Add To Cart Function

```c
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");
}
```

Remove From Cart Function

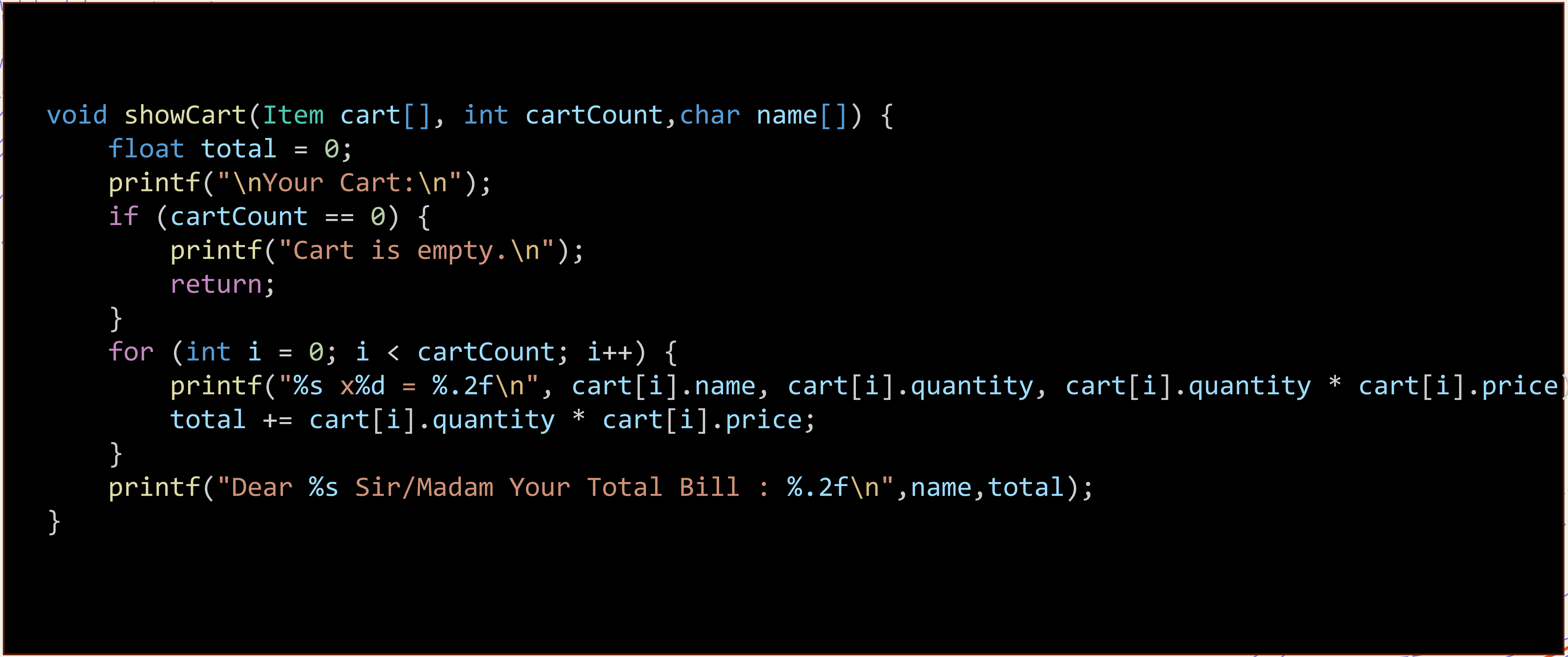
```c
void removeFromCart(Item cart[], int *cartCount) {
    int id, found = 0;
    printf("Enter product ID to remove: ");
    scanf("%d", &id);

    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");
    }
}
```

Show Products Function

```
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);
        }
}
```

Show Cart Function



```c
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);
}
```

Load Cart Function

```c
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);
}
```

Save Cart Function

```
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);
}
```

Limitations & Improvements

Limitations & Improvements

- **Product Name Formatting**: Support for multi-word names
- **Enhanced Input Validation**: More checks for numeric inputs
- **Security**: Potential for encrypting saved data (name, contact)

Conclusion

- **Achievements**: Functional and modular shopping cart application in C
- **Next Steps**: Improve usability, input handling, and interface
- **Final Thought**: This project showcases foundational skills in C programming, file handling, and user input management

THANK YOU