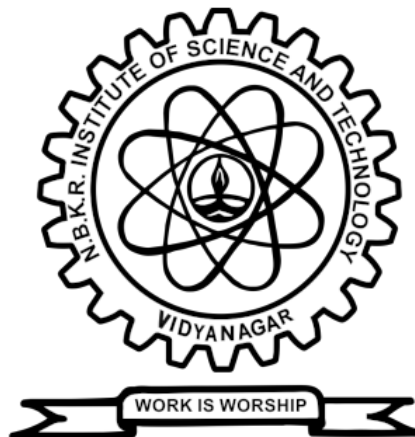


SIMPLE E-COMMERCE CART SYSTEM

**Project report submitted in partial fulfillment of the
Requirements for the Award of the Degree of
BACHELOR OF TECHNOLOGY
In
COMPUTER SCIENCE AND ENGINEERING**

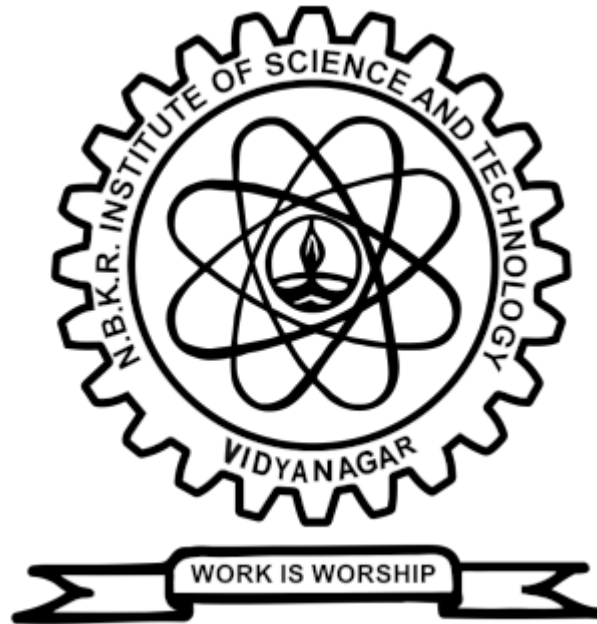
**By
24KB1A05E0
24KB1A05J6
24KB1A0550
24KB1A0592**

**Under the Guidance of
SMT.B.SRUTHI**



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
NBKRIST**

NBKRIST
(AUTONOMOUS)
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING



CERTIFICATE

This is to certify that the project report entitled YOUR PROJECT TITLE being submitted by

24KB1A05E0

24KB1A05J6

24KB1A0550

24KB1A0592

in partial fulfillment for the award of the Degree of Bachelor of Technology in Computer Science and Engineering to the Jawaharlal Nehru Technological University, Kakinada is a record of bonafied work carried out under my guidance and supervision.

SMT.B.SRUTHI

**DR.HOD RAJSHEKAR
REDDY**

Designation

**M.Tech, Ph.D
Head of the Department**

DECLARATION

I hereby declare that the dissertation entitled **Simple E-commerce Cart System** submitted for the B.Tech Degree is my original work and the dissertation has not formed the basis for the award of any degree, associateship, fellowship or any other similar titles.

Place:Vidyanagar

24KB1A05E0

Date:05-05-2025

24KB1A05J6

24KB1A0550

24KB1A0592

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to all those who contributed to the successful completion of my E-commerce Cart System project.

First and foremost, I would like to thank my [Instructor/Supervisor's Name] for their valuable guidance, constant support, and constructive feedback throughout the development process.

I am also thankful to my peers and colleagues who provided helpful suggestions and encouragement.

Lastly, I appreciate the support of my family and friends, whose encouragement and understanding helped me stay focused and motivated.

This project has provided me with practical insights into web development, database integration, and the overall process of creating a functional online shopping cart system.

ABSTRACT

This project presents a basic E-commerce Cart System implemented using the C programming language. The system simulates an online shopping experience in a console-based environment. Users can view a list of available products, add items to a virtual cart, remove items, view the cart summary, and proceed to checkout. The main objective of this project is to demonstrate the use of C programming concepts such as structures, arrays, functions, and file handling to create a simple and functional cart system. This project serves as a foundational step toward understanding how e-commerce platforms manage user orders and product data.

1. INTRODUCTION

In today's digital age, online shopping has become an essential part of everyday life. E-commerce platforms allow users to browse products, make selections, and purchase items with convenience. This project, "Simple E-commerce Cart System," is developed using the C programming language to simulate a basic shopping experience within a console application.

The system enables users to view available products, add or remove items from a virtual cart, and display the final bill during checkout. Although basic in nature, the project demonstrates core programming concepts in C such as structures, loops, arrays, functions, and file handling. It also provides insight into how product data and customer interactions can be managed programmatically.

The project serves as a foundational exercise in system design, logic building, and the application of C in real-world scenarios.

2. Problem Statement

Create a simple program in C that lets a user:

- See a list of products.
- Add products to a shopping cart.
- Remove products from the cart.

- View the cart with total cost.

You will use arrays and structures to store product and cart data. The program should be menu-driven and run in the console.

3. Scope

This project will focus on creating a basic shopping cart system using the C programming language. It will include the following features:

I. Fixed Product List

A small list of products with ID, name, and price will be predefined in the program.

II. Add to Cart

Users can add products to their cart by selecting a product ID and quantity.

III. Remove from Cart

Users can remove items from the cart if they change their mind.

IV. View Cart

Users can see all items in their cart, with quantities and total cost.

V. Menu-Driven Interface

A simple text menu will guide the user to perform actions like viewing products, adding/removing items, and checking out.

4. Objective

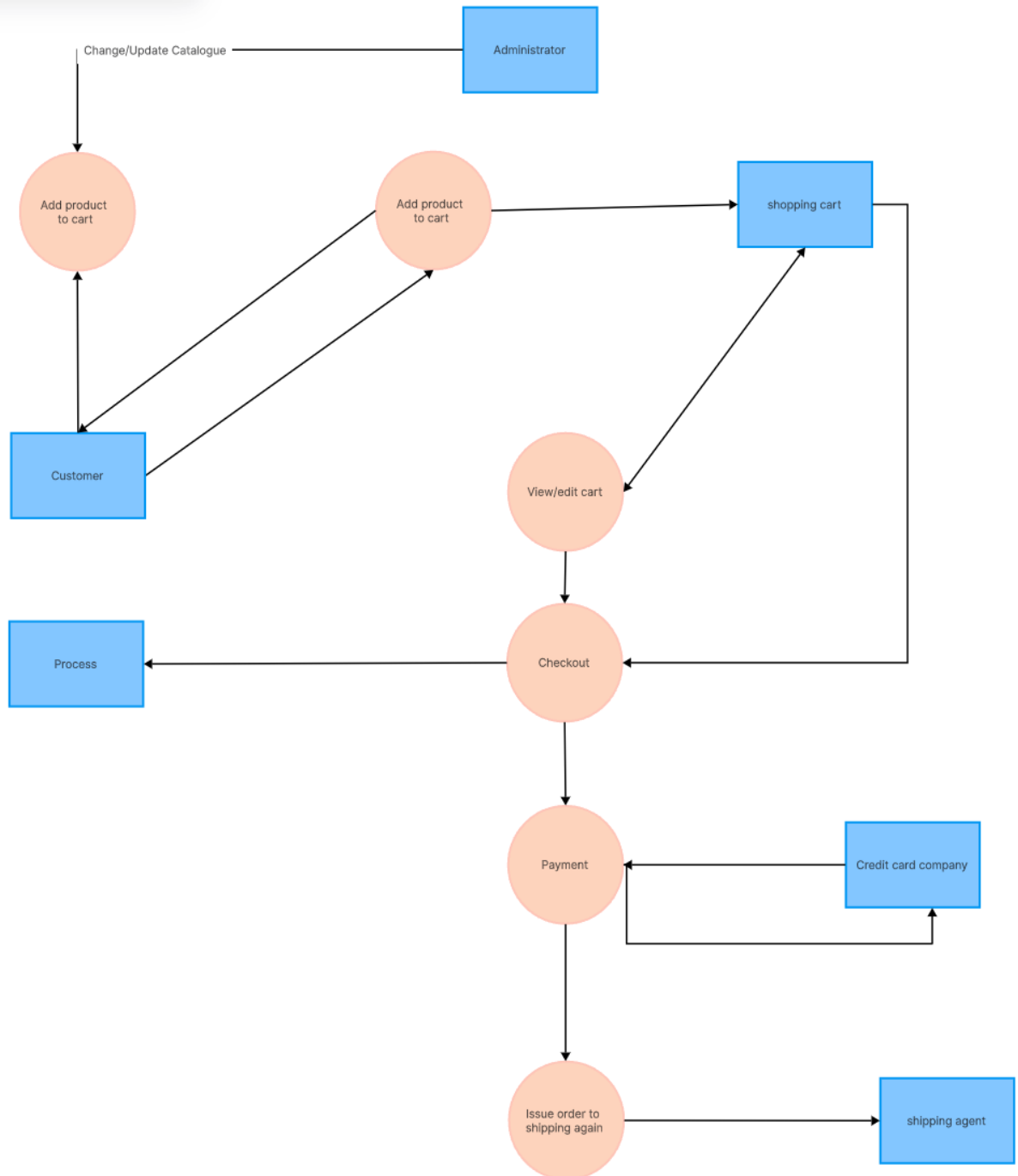
- I. Browse and select products from a predefined list.
- II. Add products to a virtual shopping cart along with desired quantities.
- III. Update or remove products in the cart before finalizing the order.
- IV. View the contents of the cart and see the total cost of selected items.
- V. Simulate the core functions of online shopping in a simplified and easy-to-use format.

This project aims to demonstrate the essential logic behind online shopping systems while focusing on usability and simplicity.

5. Software Requirement Analysis

- Operating System: Any operating system with a C compiler (e.g., Windows, Linux, macOS).
- Programming Language: C
- Compiler: GCC, Turbo C, or any standard C compiler.

6. CONTROL FLOW CHART



7. Modules And Their Functionalities

❖ Product Module

Functionality:

- Stores a list of predefined products with details like ID, name, and price.
- Allows displaying all available products to the user.

Related Elements:

- Product structure
- products[] array
- displayProducts() function

❖ Cart Module

Functionality:

- Uses a linked list to manage items added to the cart.
- Allows adding items to the cart.
- Displays cart contents with item totals and overall total.
- Clears the cart after checkout or exit.

Related Elements:

- CartNode structure

- addToCart() function
- viewCart() function
- clearCart() function

❖ Menu/Interface Module

Functionality:

- Displays the menu and handles user input.
- Calls appropriate functions based on the user's selection (view, add, checkout, exit).

Related Elements:

- main() function
- Menu loop using switch-case

❖ Utility/Checkout Module (Part of Cart)

Functionality:

- Calculates the total amount during cart viewing and checkout.
- Empties the cart after checkout to free memory.

Related Elements:

- Handled inside viewCart() and clearCart() functions
- Used in case 4 (Checkout) of the menu

This modular separation makes the code organized, easy to maintain, and suitable for extending features in the future (like file handling or user authentication).

8. Code

```
#include <stdio.h>

#include <stdlib.h>

#include <string.h>


#define MAX_PRODUCTS 5


// Product Structure
typedef struct {
    int id;
    char name[50];
    float price;
} Product;


// Node for linked list (Cart)
```

```
typedef struct CartNode {  
    Product product;  
    int quantity;  
    struct CartNode* next;  
} CartNode;
```

```
// Initialize product list
```

```
Product products[MAX_PRODUCTS] = {  
    {1, "Laptop", 50000.0},  
    {2, "Smartphone", 15000.0},  
    {3, "Headphones", 2000.0},  
    {4, "Smartwatch", 3500.0},  
    {5, "Keyboard", 700.0}  
};
```

```
// Function to display products
```

```
void displayProducts() {  
    printf("\nAvailable Products:\n");  
    printf("ID\tName\t\tPrice\n");  
    for (int i = 0; i < MAX_PRODUCTS; i++) {  
        printf("%d\t%-15s\t%.2f\n",      products[i].id,      products[i].name,  
products[i].price);  
    }
```

```

    }
}

// Add product to cart
void addToCart(CartNode** head, int id, int quantity) {
    for (int i = 0; i < MAX_PRODUCTS; i++) {
        if (products[i].id == id) {
            CartNode* newNode = (CartNode*)malloc(sizeof(CartNode));
            newNode->product = products[i];
            newNode->quantity = quantity;
            newNode->next = NULL;

            // Add to end of the list
            if (*head == NULL) {
                *head = newNode;
            } else {
                CartNode* temp = *head;
                while (temp->next != NULL)
                    temp = temp->next;
                temp->next = newNode;
            }
        }
    }
}

```

```

        printf("%s added to cart.\n", products[i].name);

        return;
    }
}

printf("Product not found!\n");
}

// Display cart

void viewCart(CartNode* head) {
    if (head == NULL) {
        printf("\nYour cart is empty.\n");
        return;
    }

    printf("\nYour Cart:\n");

    float total = 0;

    printf("Name\t\tPrice\tQty\tTotal\n");

    while (head != NULL) {
        float itemTotal = head->product.price * head->quantity;

        printf("%-15s %.2f\t%d\t%.2f\n", head->product.name, head->product.price,
head->quantity, itemTotal);

        total += itemTotal;
    }
}

```

```

        head = head->next;
    }
    printf("Total Amount: %.2f\n", total);
}

```

// Free cart memory

```

void clearCart(CartNode** head) {
    CartNode* temp;
    while (*head != NULL) {
        temp = *head;
        *head = (*head)->next;
        free(temp);
    }
}

```

```

int main() {
    CartNode* cart = NULL;
    int choice, id, quantity;

    while (1) {
        printf("\nE-Commerce Menu:\n");

```



```
printf("1. Display Products\n");  
  
printf("2. Add to Cart\n");  
  
printf("3. View Cart\n");  
  
printf("4. Checkout\n");  
  
printf("5. Exit\n");  
  
printf("Enter choice: ");  
  
scanf("%d", &choice);  
  
switch (choice) {  
    case 1:  
        displayProducts();  
        break;  
    case 2:  
        printf("Enter product ID to add: ");  
        scanf("%d", &id);  
        printf("Enter quantity: ");  
        scanf("%d", &quantity);  
        addToCart(&cart, id, quantity);  
        break;  
    case 3:  
        viewCart(cart);
```

```
        break;

    case 4:

        viewCart(cart);

        printf("Checkout complete. Thank you for shopping!\n");

        clearCart(&cart);

        break;

    case 5:

        clearCart(&cart);

        printf("Exiting program.\n");

        exit(0);

    default:

        printf("Invalid choice.\n");

    }

}

return 0;

}
```

9. Output Screen

```
--- E-commerce Cart System ---
1. View Products
2. Add to Cart
3. View Cart
4. Remove from Cart
5. Checkout
6. Exit
Enter your choice: 1

Available Products:
ID  Name          Price   Stock
1   Laptop         45000.00  5
2   Headphones     1500.00  10
3   Mouse          500.00   15
4   Keyboard       800.00   10
5   Monitor        7000.00   7

--- E-commerce Cart System ---
1. View Products
2. Add to Cart
3. View Cart
4. Remove from Cart
5. Checkout
6. Exit
Enter your choice: 2
```

```
--- E-commerce Cart System ---
```

1. View Products
2. Add to Cart
3. View Cart
4. Remove from Cart
5. Checkout
6. Exit

```
Enter your choice: 2
```

```
Enter Product ID: 3
```

```
Enter Quantity: 4
```

```
Item added to cart.
```

```
|
```

```
--- E-commerce Cart System ---
```

1. View Products
2. Add to Cart
3. View Cart
4. Remove from Cart
5. Checkout
6. Exit

```
Enter your choice: 6
```

```
Exiting...
```

10. Conclusion

In conclusion, this project successfully demonstrates the fundamental structure of a simple e-commerce cart system using the C programming language. It incorporates basic functionalities such as adding items to the cart, removing items, viewing the cart, and calculating the total price. Through this implementation, we have explored essential C concepts like structures, arrays, functions, and control statements. While the system is minimal and console-based, it lays the groundwork for more complex and user-friendly e-commerce applications. Future enhancements could include file handling for persistent storage, user authentication, and a graphical interface for improved user experience.

