**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**

**Chennamsetty Sai Lakshmi – 24KB1A05A0**

**Chinaswami Sadhvika – 24KB1A05B5**

**Chinthamreddy Gireesha – 24KB1A05B8**

**Devareddy Chaitanya Sri – 24KB1A05D8**

**Under the Guidance of**

**Smt.B.Sruthi**

****

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**N.B.K.R INSTITUTE OF SCIENCE & TECHNOLOGY**

**(AUTONOMOUS)**

VIDYANAGAR-524413(Kota,Tirupati district,Andhara Pradesh)

N.B.K.R INSTITUTE OF SCIENCE &TECHNOLOGY

**(AUTONOMOUS)**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**



###### CERTIFICATE

This is to certify that the project report SIMPLE E-COMMERCE CART SYSTEM being submitted by

**Chennamsetty Sai Lakshmi – 24KB1A05A0**

**Chinaswami Sadhvika – 24KB1A05B5**

**Chinthamreddy Gireesha – 24KB1A05B8**

**Devareddy Chaitanya Sri – 24KB1A05D8**

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

|  |  |
| --- | --- |
| **Mrs.B.Sruthi Designation** | **Dr.A.Raja Sekhar Reddy**  **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 **Chennamsetty Sai Lakshmi – 24KB1A05A0**

Date:05-05-2025 **Chinaswami Sadhvika – 24KB1A05B5**

**Chinthamreddy Gireesha – 24KB1A05B8**

**Devareddy Chaitanya Sri – 24KB1A05D8**

## **Acknowledgement**

I would like to express my sincere gratitude to all those who supported me throughout the development of my project titled **“E-Commerce Cart System.”**

First and foremost, I extend my heartfelt thanks to my project guide and teacher, **Mrs.B.Sruthi** ,whose expert guidance, encouragement, and constructive feedback were instrumental in the successful completion of this project. Their valuable suggestions helped me improve the functionality and structure of the system at every stage.

I am also grateful to the faculty members of the **CSE**, for providing the knowledge and skills that laid the foundation for this project. Their teachings and support created a strong academic environment that inspired me to take on this work.

A special word of thanks to my friends, for their cooperation, feedback, and constant encouragement throughout the development process. Their inputs, testing assistance, and motivation helped me refine the project and stay on track.

This project helped me understand the practical implementation of concepts like arrays, linked lists, and menu-driven interfaces in C programming. It enhanced my logical thinking and problem-solving skills. I also learned the importance of user experience and interface design in software development. Collaborating with peers and discussing challenges made the learning process even more engaging. The entire journey has been a valuable learning experience that will help me in future technical endeavors.

Finally, I would like to thank everyone who directly or indirectly contributed to the successful development and completion of this project.

**Abstract**

This document presents the development of a simple **E-commerce Cart System** to demonstrate fundamental data structures and procedural programming concepts using the **C language**. The system simulates an online shopping experience, enabling users to interact with a list of available products and manage a virtual shopping cart. It highlights how basic programming concepts such as arrays and linked lists can be applied to real-world applications in e-commerce.

Key functionalities include adding products to the cart, updating item quantities, and removing products, all managed using a **linked list**. Each product in the cart is represented as a node, containing product details (ID, name, price) and quantity. The system also calculates the **total cost** based on the selected items and their quantities, providing the user with a final amount for the purchase.

While the system offers a basic implementation, it provides a foundation for understanding core mechanisms involved in online shopping carts. It demonstrates practical applications of **arrays**, **linked lists**, and **menu-driven user interfaces** in a console-based environment. However, the project also acknowledges limitations such as scalability, advanced filtering, and integration with real payment systems, which are beyond its scope.

Overall, this project serves as an educational tool for learning how to design simple e-commerce systems while offering a glimpse into more complex features that can be added in future versions, such as multi-user support and product search functionality.

TABLE OF CONTENTS

S.NO TOPIC PAGE

1. Introduction 1

2. Literature Survey 2

3. Software Requirement Analysis 3

4. Software Design 4

5. Proposed system 4-5

6. Coding 6-11

7. Output 11-12

8. Conclusion 13

1. **Introduction**

This project focuses on the development of a basic e-commerce cart system implemented in the C programming language. The primary objective is to simulate the functionality of a real-world online shopping cart while applying fundamental programming principles. The system is designed to utilize arrays for storing product information and a linked list to manage the shopping cart. This combination allows for clear organization and efficient operations, such as adding, removing, and displaying items in the user's cart. Arrays provide fast access to a predefined list of available products, while linked lists offer dynamic memory allocation, making it easy to modify the contents of the cart during runtime without the limitations of fixed-size structures.

Using C for this project reinforces key procedural programming concepts, such as pointer manipulation, dynamic memory management, and the use of user-defined data types (e.g., structs The system features a simple, text-based user interface, allowing users to interact with the cart via a command-line menu. This interface, although minimal, helps users navigate through the options easily and supports core shopping actions such as viewing the catalog, selecting products, updating quantities, and reviewing the final bill. While the project does not include advanced features such as user authentication, persistent storage, or graphical user interfaces, it lays a strong foundation for understanding how a typical e-commerce platform operates at the data level. The project can be further enhanced by incorporating file handling for data persistence, implementing error checking for more robust input handling, and possibly extending the interface into a basic GUI using tools like C-based graphics libraries.

1

**2.Literature Survey**

The development of e-commerce cart systems has been extensively studied as a crucial component of online shopping platforms. A shopping cart allows users to select, review, and manage products before finalizing their purchase, making it a core feature in enhancing user experience and boosting conversion rates. Researchers like Li and Zhang (2002) have emphasized the impact of shopping cart design on customer satisfaction and purchase decisions. Modern implementations often include features such as real-time item updates, persistent cart sessions, and guest checkouts to minimize friction during the shopping process. The integration of secure payment gateways, such as PayPal or Stripe, is also a common focus in literature, with studies stressing the importance of SSL encryption and tokenized transactions for protecting user data (Huang & Benyoucef, 2013). Additionally, the rise of mobile commerce has driven the need for responsive cart systems that function seamlessly across devices, with Progressive Web Apps (PWAs) emerging as a popular solution.

Technologically, most shopping cart systems utilize a combination of frontend frameworks like React or Angular, backend technologies such as Node.js, Django, or PHP, and databases like MySQL or MongoDB to manage product and user information. The use of RESTful APIs enables efficient communication between the client and server, improving performance and scalability. Literature also highlights the importance of good user interface (UI) and user experience (UX) design principles, including simplified checkout processes and clear navigation paths, which help reduce cart abandonment rates (Norman, 2013; Kohavi et al., 2009). Comparative studies of open-source platforms such as WooCommerce

2

Magento offer insights into the effectiveness of different implementation strategies and suggest that combining security, usability, and mobile compatibility leads to a more successful e-commerce cart system.

**3.** **Software Requirement Specification**

**3.1. Functional Requirements**

 \* Adding Items to the Cart: Users can select products and add them to their virtual shopping cart. The system identifies the product and records the desired quantity.

 \* Viewing the Cart: Users can view a summary of the items in their cart, including the item name, quantity, unit price, and subtotal for each item.

 \* Updating Quantities: Users can modify the quantity of items already in their cart. The system automatically recalculates the subtotals and the total.

 \* Removing Items from the Cart: Users can remove items from their shopping cart. The cart display updates upon removal.

 \* Calculating the Total: The system calculates the total cost of all items in the shopping cart by summing their subtotals.

**3.2. Non-Functional Requirements**

The primary non-functional requirements for this basic system are simplicity and basic usability. The system is intended to be easy to understand and demonstrate the core concepts effectively.

3

**4. Software Design**

**4.1. Data Flow**

The system involves the flow of product data from an array (representing the inventory) to a linked list (representing the user's cart) when a user adds an item. Information is retrieved from the linked list to display the cart contents and calculate the total. User input drives the actions of adding, updating, and removing items.

**4.2. Design of Flow Algorithms**

 \* Adding: Retrieve product details from the array based on user selection, create a node in the linked list with this information and the specified quantity.

 \* Viewing: Traverse the linked list and display the details (name, quantity, price, subtotal) of each node.

 \* Updating: Locate the node corresponding to the item being updated, modify its quantity, and recalculate the subtotal.

 \* Removing: Locate the node to be removed in the linked list and adjust the links to exclude it.

 \* Calculating Total: Traverse the linked list and sum the subtotal of each node.

**5.Proposed System**

**5.1. Modules and Functionalities**

The proposed system consists of the following conceptual modules:

 \* Product Inventory: Stores the list of available products (name, ID, price) using an array. 4

4 \* Shopping Cart: Manages the user's selected items (product, quantity) using a singly linked list.

 \* Add to Cart Functionality: Allows users to add products from the inventory to their shopping cart.

 \* View Cart Functionality: Enables users to see the items currently in their shopping cart with their details.

 \* Update Quantity Functionality: Permits users to change the quantity of items in their cart.

 \* Remove from Cart Functionality: Allows users to delete items from their cart.

 \* Calculate Total Functionality: Computes the total cost of all items in the cart

.

**6. Code Logic**

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

struct Product {

int id;

char name[50];

float price;

}; 5

struct CartNode {

int productId;

int quantity;

struct CartNode\* next;

};

#define PRODUCT\_COUNT 5

struct Product products[PRODUCT\_COUNT] = {

{1, "Laptop", 75000.00},

{2, "Headphones", 1500.00},

{3, "Keyboard", 700.00},

{4, "Monitor", 12000.00},

{5, "Mouse", 500.00}

};

struct CartNode\* cartHead = NULL;

struct Product\* findProduct(int id) {

for (int i = 0; i < PRODUCT\_COUNT; i++) {

if (products[i].id == id) return &products[i];

}

return NULL; 6

}

void addToCart(int productId, int quantity) {

struct CartNode\* current = cartHead;

while (current != NULL) {

if (current->productId == productId) {

current->quantity += quantity;

printf("Updated quantity for Product ID %d.\n", productId);

return;

}

current = current->next;

}

struct CartNode\* newNode = (struct CartNode\*)malloc(sizeof(struct CartNode));

newNode->productId = productId;

newNode->quantity = quantity;

newNode->next = cartHead;

cartHead = newNode;

printf("Product ID %d added to cart.\n", productId);

}

void showProducts() {

printf("ID Name Price\n");

for (int i = 0; i < PRODUCT\_COUNT; i++) { 7

printf("%-3d %-12s ₹%.2f\n", products[i].id, products[i].name, products[i].price);

}

}

void showCart() {

struct CartNode\* current = cartHead;

if (current == NULL) {

printf("Cart is empty.\n");

return;

}

float total = 0.0;

printf("ID Name Qty Price\n");

while (current != NULL) {

struct Product\* p = findProduct(current->productId);

if (p != NULL) {

float cost = p->price \* current->quantity;

printf("%-3d %-12s %-3d ₹%.2f\n", p->id, p->name, current->quantity, cost);

total += cost;

} 8

current = current->next;

}

printf("Total: ₹%.2f\n", total);

}

void checkout() {

showCart();

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

struct CartNode\* current = cartHead;

while (current != NULL) {

struct CartNode\* temp = current;

current = current->next;

free(temp);

}

cartHead = NULL;

}

int main() {

int choice, productId, quantity;

do {

printf("\n1.Show Products 2.Add to Cart 3.View Cart 4.Checkout 0.Exit\nEnter choice: ");

scanf("%d", &choice); 9

switch (choice) {

case 1:

showProducts();

break;

case 2:

printf("Enter Product ID: ");

scanf("%d", &productId);

printf("Enter Quantity: ");

scanf("%d", &quantity);

if (findProduct(productId)) addToCart(productId, quantity);

else printf("Invalid Product ID.\n");

break;

case 3:

showCart();

break;

case 4:

checkout();

break;

case 0:

printf("Goodbye!\n"); 10

break;

default:

printf("Invalid choice.\n");

}

} while (choice != 0);

return 0;

}

**7.Output result**

1.Show Products

2.Add to Cart

3.View Cart

4.Checkout 0.Exit

Enter choice: 1

ID Name Price

1 Laptop ₹75000.00

2 Headphones ₹1500.00

3 Keyboard ₹700.00

4 Monitor ₹12000.00

5 Mouse ₹500.00 11

Enter choice: 2

Enter Product ID: 3

Enter Quantity: 2

Product ID 3 added to cart.

Enter choice: 3

ID Name Qty Price

3 Keyboard 2 ₹1400.00

Total: ₹1400.00

Enter choice: 4

ID Name Qty Price

3 Keyboard 2 ₹1400.00

Total: ₹1400.00

Thank you for shopping! 12

**8.Conclusion**

This E-commerce Cart System, built using C, demonstrates core programming concepts through the use of arrays and linked lists. It simulates an online shopping cart where users can add, view, update, and remove items, with automatic total cost calculation. Arrays store product details for quick access, while linked lists manage the cart dynamically. The simple text-based interface offers a practical way to understand data structures and memory handling in a procedural programming context.

Though functional, the system has limitations—it lacks user authentication, persistent storage, error handling, and a graphical interface. Despite this, it serves as a strong foundation for further development. Future improvements could include login features, file or database integration, a GUI, and advanced functionalities like search or payment systems, making it more aligned with real-world e-commerce applications.

13