

*Green University of Bangladesh*

#### Department of Computer Science and Engineering (CSE) Semester: (Spring, Year: 2024), B.Sc. in CSE (Day)

**E-commerce Platform Management System**

#### Course Title: Data Structure Lab Course Code: CSE 206

#### Section: 232\_D8

### Students Details

|  |  |
| --- | --- |
| **Name** | **ID** |
| Taslim Ahmed Tamim | 232002105 |
| - | - |

#### Submission Date: 08/12/2024

#### Course Teacher’s Name: Md. Parvez Hossain

### [For teachers use only: Don’t write anything inside this box]

**Date:**

**Comments:**

**Signature:**

**Marks:**

**Lab Project Status**

**Contents**

1. [Introduction](#_bookmark0) 2
   1. [Overview](#_bookmark1) 2
   2. [Motivation](#_bookmark2) 2
   3. [Problem Definition](#_bookmark3) 3
      1. [Problem Statement](#_bookmark4) 3
   4. [Design Goals/Objectives](#_bookmark5) 3
   5. [Application](#_bookmark6) 4
2. [Design/Development/Implementation of the Project](#_bookmark7) 5
   1. [Introduction](#_bookmark8) 5
   2. [Project Details](#_bookmark9) 5
      1. [Techonologies used](#_bookmark10) 6
   3. [Implementation](#_bookmark11) 6
   4. [Source Code](#_bookmark12) 7
3. [Conclusion](#_bookmark13) 8
   1. [Discussion](#_bookmark14) 8
   2. [Limitations](#_bookmark15) 8
   3. [Scope of Future Work](#_bookmark16) 9

**Chapter 1 Introduction**

# Overview

The E-Commerce System is a console-based application developed in C that simulates the core features of an online shopping platform. It allows users to view products, add items to their cart, remove them, and complete purchases through a checkout process. Admins can manage the product catalog by adding, restocking, and removing products. The system uses structures to represent products and cart items, implements binary search for efficient product lookup, and employs file handling to load and save prod- uct data. This project showcases basic data structures, memory management, and file operations, providing a simple yet functional e-commerce simulation.

# Motivation

**Practical Application of Programming Skills:** The project allowed for the practical application of core concepts in computer science, including data structures, memory management, and algorithms, in a real-world e-commerce scenario.

**Simulating E-Commerce Functionality:** The goal was to build a simple, func- tional console-based e-commerce platform with essential features like product brows- ing, cart management, and checkout.

**Learning Through Problem Solving:** Developing the project involved solving challenges such as managing inventories, handling user inputs, and implementing al- gorithms like binary search to efficiently manage product data.

**Software Engineering Practices:** The project focused on simulating typical e- commerce system workflows, thus enhancing understanding of real-world system de- sign and transaction processing.

**Improving Practical Coding Skills:** Through this project, I aimed to refine problem- solving and coding skills, focusing on dynamic memory allocation, file handling, and building modular code structures.

# Problem Definition

The project aims to develop a simple, console-based e-commerce system that allows users to browse products, manage their shopping cart, and complete purchases. It in- cludes essential features like adding/removing products from the cart, viewing product details, and a checkout process. Additionally, the system provides admin functionality to manage products by adding, restocking, or removing them. The data is stored persis- tently in files, ensuring the system maintains product information across sessions. This project seeks to simulate the fundamental operations of an e-commerce platform while focusing on simplicity and user experience.

## Problem Statement

The problem at hand is to create a basic e-commerce system that allows users to interact with products, manage a shopping cart, and complete purchases, all within a console- based application. The system should support essential operations such as viewing products, adding or removing items from the cart, checking out, and managing product inventory. It should also include an admin interface to add, remove, or restock products. The goal is to design a simple and efficient solution to demonstrate the core function- ality of an e-commerce platform while managing product data and user actions in an organized manner.

# Design Goals/Objectives

The main objectives of the project are to create a simple yet functional e-commerce sys- tem that offers core shopping features with an easy-to-use interface and robust backend management.

* User-Friendly Interface: The application aims to provide an intuitive text-based interface where users can easily navigate between product listings, their cart, and checkout. The focus is on making it simple for anyone to use without requiring technical knowledge.
* Product Management: The project should allow an admin to efficiently manage the product catalog. Admin functionalities include adding new products, updat- ing product details (such as stock and prices), and removing products from the inventory when they are no longer available.
* Cart Functionality: The cart system should allow users to add products, set quan- tities, and remove items as needed. The system will also ensure that product stock is updated in real-time when items are added or removed, maintaining data integrity.
* Efficient Product Search: To enable quick access to a product from a large in- ventory, the system implements a binary search algorithm to search for products by their names. This improves the search efficiency compared to linear searches, especially as the number of products grows.
* Order Checkout: Once a user has finished selecting items, they can proceed to the checkout, where the system displays the total cost of their cart, confirms their purchase, and clears the cart afterward. This simulates the final steps of an e- commerce purchase process.
* Scalability: The design will allow for easy scalability, so additional features can be added in the future. These features might include a payment system, customer profiles, or a graphical user interface (GUI) to enhance the user experience.
* Robust Error Handling: The system will incorporate proper error-handling mech- anisms to deal with issues such as invalid input, insufficient stock, or unavail- able products. This ensures that users always receive clear feedback, preventing crashes or confusion.

# Application

This e-commerce system project serves as a foundational application for simulating a basic online shopping platform. While its primary purpose is to demonstrate funda- mental features of an e-commerce site, it can be expanded or integrated into real-world scenarios with a few modifications. Some potential applications of this project include:

* Small and Medium Retail Businesses: Small businesses looking to transition to online sales can use this system as a simple, cost-effective e-commerce platform. It can help them manage product catalogs, track inventory, and facilitate the shop- ping experience, allowing them to reach a broader audience without significant investment in complex systems.
* Personalized Online Shopping Platforms: Businesses can adapt this system for personalized shopping experiences by integrating features like product recommendations, discounts, and promotions. It could be the base for an online store that uses customer behavior and preferences to suggest items, improving the shop- ping experience and boosting sales.
* Inventory Management for Retail Chains: Retail chains can utilize this system as a foundational tool for managing and tracking product inventories across multiple locations. By maintaining product data, including stock levels, prices, and restocking needs, the system can optimize inventory control and reduce stockouts or overstocking situations.
* Test Environment for Payment Gateway Integration: The system could be used in testing environments where developers integrate and test payment gateways, such as PayPal or Stripe. It provides a simple interface to simulate transactions, helping developers ensure that payment processing works correctly before deployment to a live environment.
* Time saving: By automation system for billing and ordering it consume less time than manual and also more efficient

**Chapter 2**

**Design/Development/Implementation of the Project**

# Introduction

The e-commerce system project is a console-based application that simulates an online shopping platform. It allows users to view products, add them to a shopping cart, and proceed with checkout. Administrators can manage the product inventory, including adding, restocking, and removing items. The system incorporates basic functionali- ties of a real-world e-commerce platform, offering a simple, user-friendly interface for both customers and admins. It serves as a foundation for learning and developing e- commerce applications. [[1](#_bookmark17)] [[2](#_bookmark18)] [[3](#_bookmark19)].

# Project Details

The project is a console-based e-commerce platform built using the C programming language. It simulates an online shopping environment, allowing users to browse avail- able products, add items to their shopping cart, and proceed with checkout. The system is structured into several core functionalities:

**Product Management:** Admin users have the ability to manage the product catalog. They can add new products, restock existing ones, or remove products that are no longer available. This is done by maintaining an array of product structures, each containing attributes like product ID, name, price, and stock quantity.

**Shopping Cart:** Users can add products to their cart by selecting a product and specifying the quantity. The cart itself is implemented using a linked list, where each node represents a product in the cart, storing its product details and the quantity the user wants to purchase. The system checks product availability before adding it to the cart, ensuring that there is sufficient stock.

**Checkout Process:** After browsing and adding items to their cart, users can view the contents of their cart and proceed to checkout. Upon checkout, the total price is dis- played, and the cart is cleared, simulating a purchase.

**Persistence of Data:** The project incorporates file handling to persist product data. This ensures that the product catalog is stored across program runs. When the program starts, it attempts to load existing product data from a file, and when the user exits, any changes to product stock or new products are saved back to the file.

**Search and Binary Search:** The project implements a binary search algorithm to efficiently locate products by their name. This enhances the performance when searching for items, especially as the product catalog grows.

## Technologies used

To develop this project I used C programming language. To run the project I used visual studio code editor.



Figure 2.1: Visual studio code

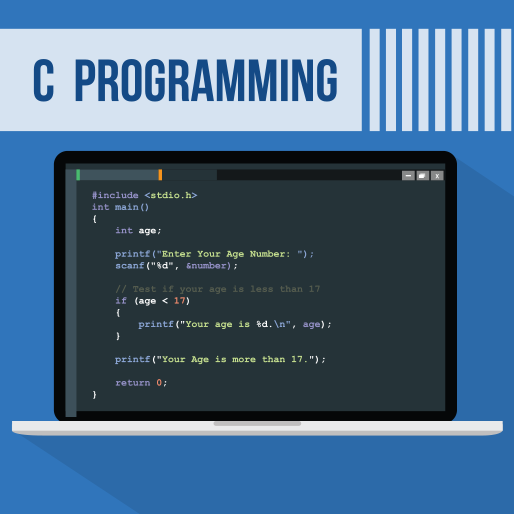


Figure 2.2: C Programing language

# Implementation

##### Data Structures

Product Structure: The project uses a structure to represent a product, which holds essential information such as product ID, name, price, and stock quantity. CartItem Structure: A linked list is used to represent the shopping cart. Each node in the list represents an item in the cart, which stores a product and its quantity.

##### Functionality

Displaying Products: A function is implemented to list all available products, displaying their details such as ID, name, price, and stock.

**Product Search:** A binary search method is used to efficiently locate products by their name in the product list, making the search operation fast even as the product list grows.

**Add to Cart:** A function allows users to add products to the cart by checking product availability and stock levels before adding the item. It also updates the stock of the product accordingly.

**View Cart:** Users can view the contents of their shopping cart, including the names, prices, quantities, and total price of all products in the cart.

**Remove from Cart:** If a user wants to remove a product from the cart, the system searches the cart for the specified product and removes it, updating the stock of the product.

**Checkout:** This function handles the finalization of the purchase, displaying the cart contents, calculating the total amount, and clearing the cart once the purchase is completed.

**Admin Operations:** Admin users have special privileges to add, restock, or remove products from the catalog. This is done through admin-specific functions.

##### File Handling

The system supports data persistence through file handling. Product data (such as ID, name, price, and stock) is saved to and loaded from a file (products.txt). This ensures that product information is retained between program executions.

##### User Interface

The program operates through a console-based menu system, where users can interact with the system by entering their choices. Options include browsing products, manageing the shopping cart, and performing admin tasks. The menu is continually displayed until the user chooses to exit.

##### Memory Management

The program dynamically allocates memory for storing products and cart items using malloc and realloc. This allows the system to handle varying numbers of products and cart items efficiently.

# Project Screenshots:

# 

Figure 1: Menu

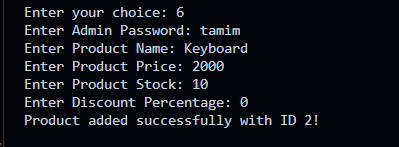
****

Figure 2: admin product add

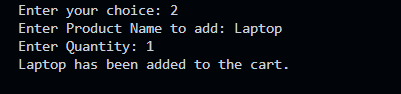


Figure 3: User Product add in the cart

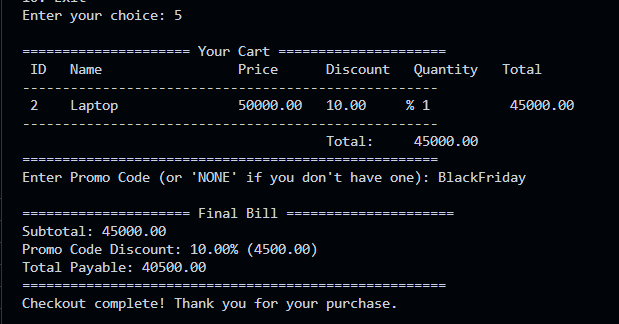


Figure 4: Product checkout

**Chapter 3 Conclusion**

In conclusion, the e-commerce console-based project demonstrates the fundamental functionality of an online shopping system, such as viewing products, adding/removing items from the cart, and managing product data. However, it faces challenges such as limited user interaction, memory management, error handling, and scalability. The lack of a graphical user interface (GUI) restricts the user experience, while the reliance on file-based storage for product data introduces potential data loss risks. Despite these limitations, the project provides a solid foundation for understanding core e-commerce concepts, and with further improvements, it could evolve into a more advanced, user- friendly, and scalable platform.

# Discussion

The implementation of this e-commerce console-based project demonstrated the core functionalities of an online shopping system, focusing on user interaction, data management, and inventory control. The system effectively allows users to browse products, add them to a cart, and perform checkout, while administrators can manage products and stock. The use of linked lists for the cart and binary search for product lookup ensures efficient data handling. However, while the console interface is functional, a graphical user interface (GUI) would improve user experience. Additionally, performance optimization can be considered for larger datasets or more complex operations.

# Limitations

The e-commerce project, while functional, has several limitations. It lacks a graphical user interface (GUI), making it less visually appealing compared to modern applica- tions. Additionally, there is no user authentication, meaning personalized features such as individual carts are not supported. The product search functionality relies on binary search with exact string matching, which can be cumbersome due to case sensitivity and lack of support for partial matches. The system uses file-based storage for product data, which is less efficient and scalable than database systems. Furthermore, the checkout process lacks real payment gateway integration, making it a simulation rather than a

fully realized transaction system.

# Scope of Future Work

To enhance the e-commerce project, several improvements can be made. First, transitioning from a console-based interface to a graphical user interface (GUI) or a web- based platform using technologies like GTK or web frameworks such as HTML, CSS, JavaScript, and backend solutions like Node.js or Django would significantly improve the user experience. Additionally, replacing file-based storage with a more robust database system like MySQL, SQLite, or MongoDB would streamline the management of product inventory, user data, and cart information, ensuring better efficiency and security. Implementing user authentication and account features would enable personalized shopping experiences, allowing users to save preferences, wish lists, and order history. Integration with payment gateways such as PayPal, Stripe, or Razor pay would allow for secure online transactions, making the checkout process complete. Finally, incorporating an AI-powered chatbot would assist users with product inquiries, track orders, and offer recommendations, further improving customer service and satisfaction.

**References**

1. Google: www.google.com
2. YouTube tutorial: https://www.youtube.com/watch?v=Zzs6kLlkAUQ