**Project Report on Shopping Cart System: Emphasizing Linked Lists and Stack**

**Team Members:**

* 22K4623 Areeb Ahmed
* 22K4279 Farrukh Ali
* 22K4503 Pranjal

**Overview**

This project introduces a Shopping Cart System developed in C++, emphasizing the use of linked lists and stack data structures. The system is designed to simulate a shopping environment, managing product inventory and customer transactions. Its core functionality revolves around efficient data handling and user interaction within a console-based interface.

**Key Components**

**1. Data Structures**

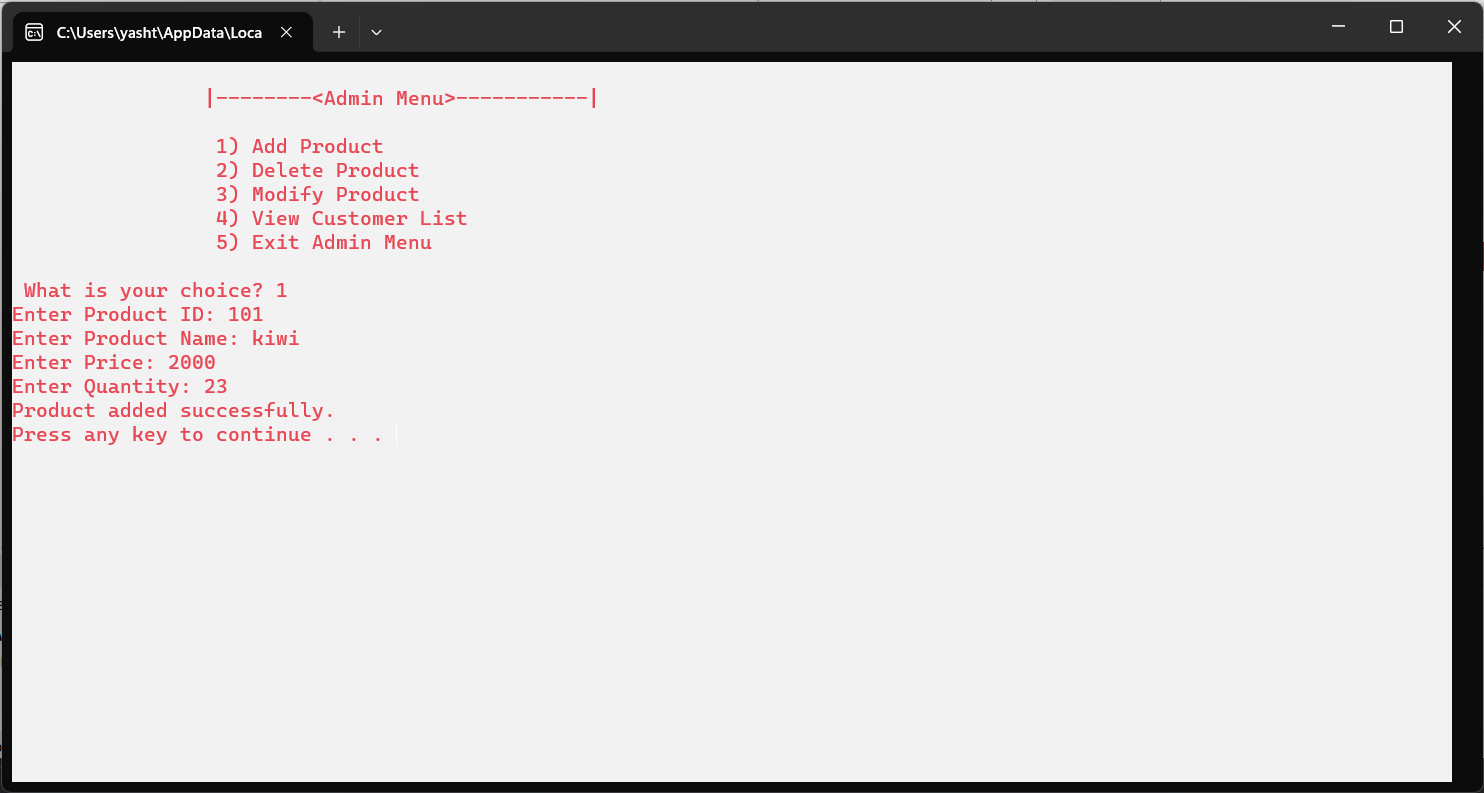
* **Linked Lists**: Used in the CustomerList class to manage dynamic customer data and in stackProduct for handling products.
* **Stack**: Implemented in the stack class to represent products with attributes like ID, name, price, and quantity.

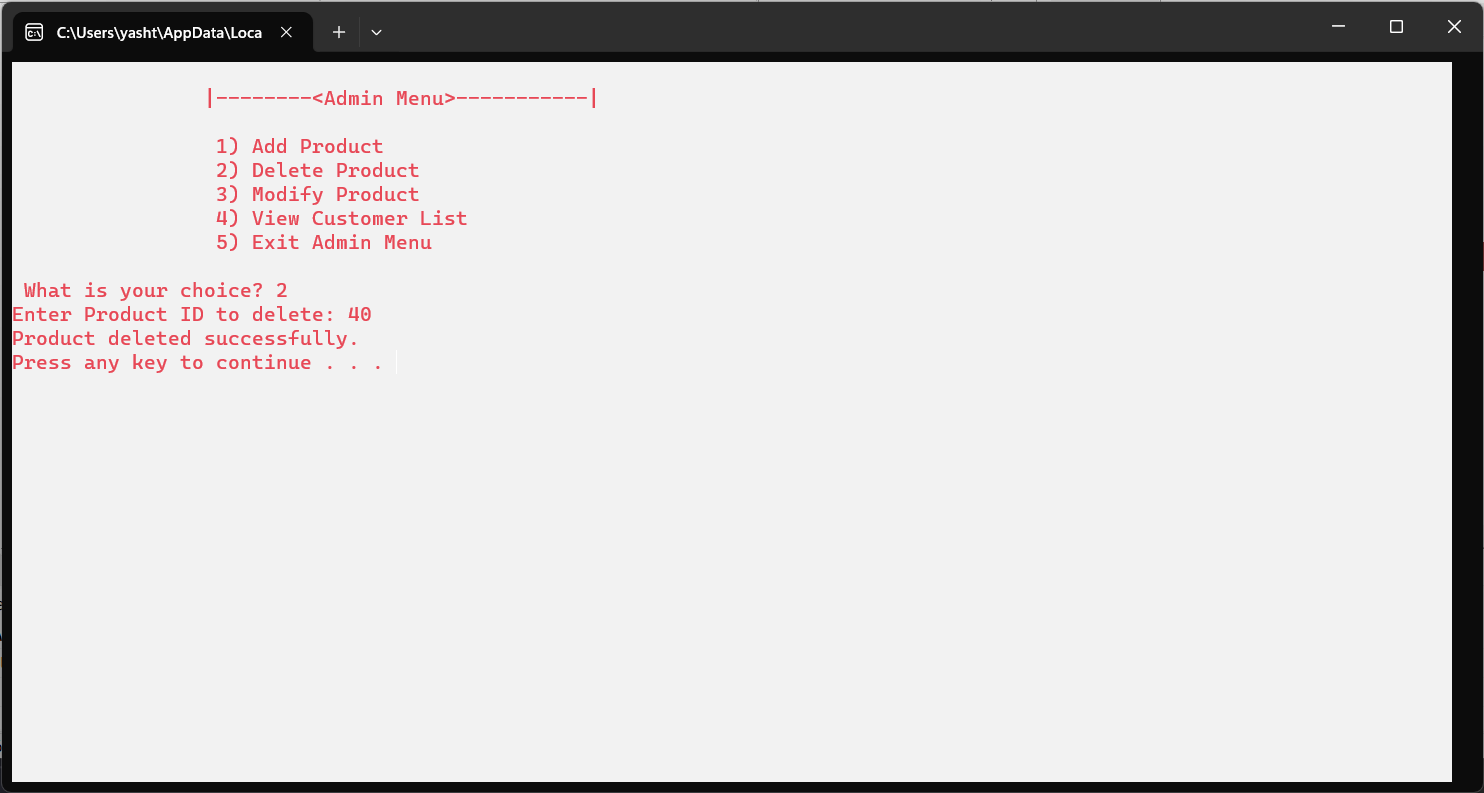
**2. Class Design**

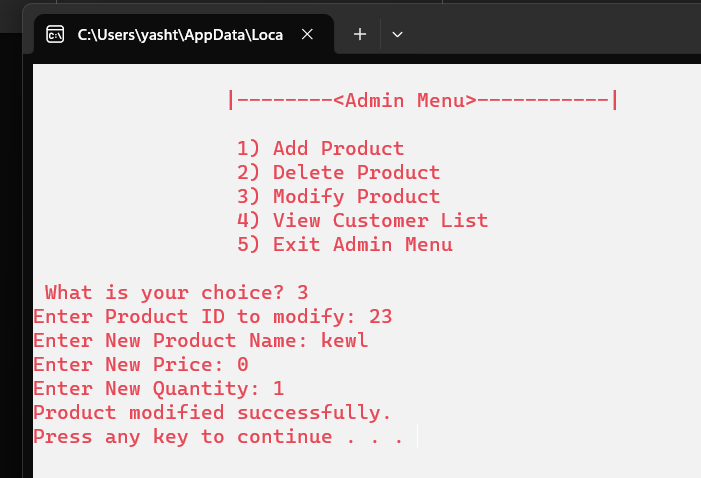
* **stack**: Manages product details and serves as a node in the product linked list.
* **customer**: Holds customer information including name and total payment.
* **CustomerList**: Manages a list of customer objects using linked lists, facilitating add and display operations.
* **stackProduct**: Functions as the main class for product management, utilizing linked lists for storing and manipulating product data.

**3. System Functionalities**

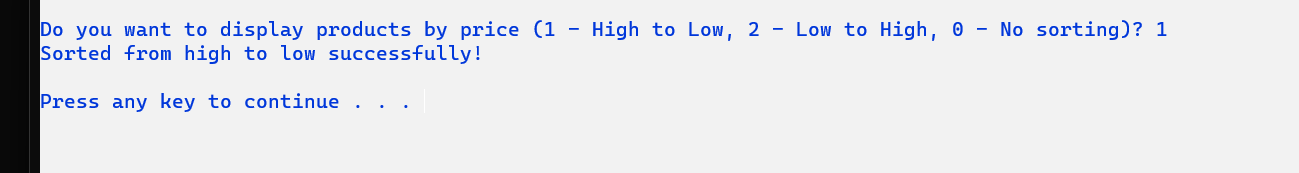
* **Product Management**: Facilitates adding, modifying, and deleting products using stack and linked list operations.





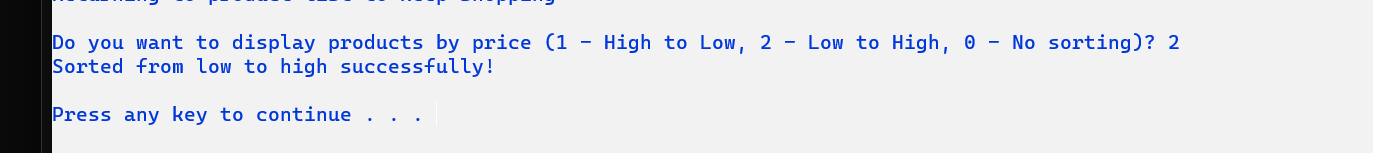


* **Customer Shopping Flow**: Allows customers to browse, sort products, and manage their shopping carts effectively.

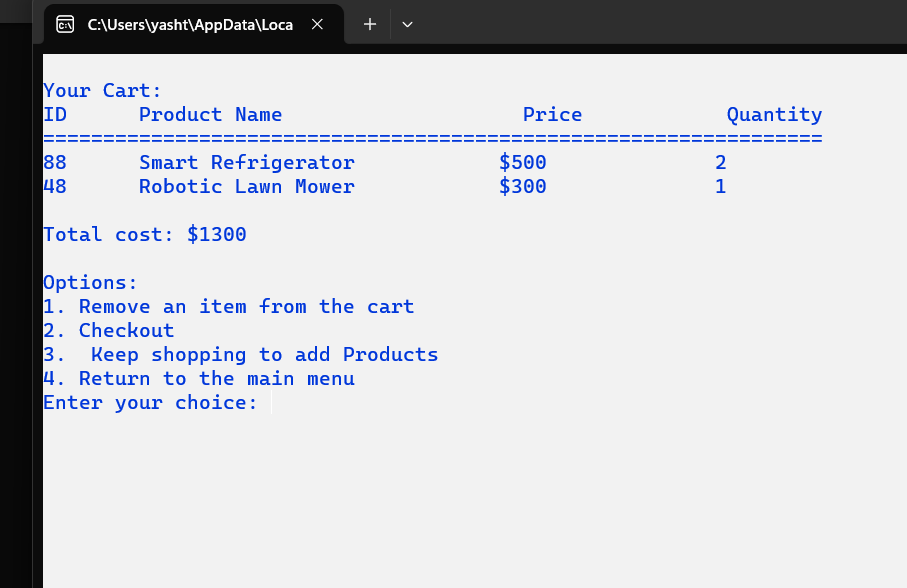


High to low- is displayed below:-

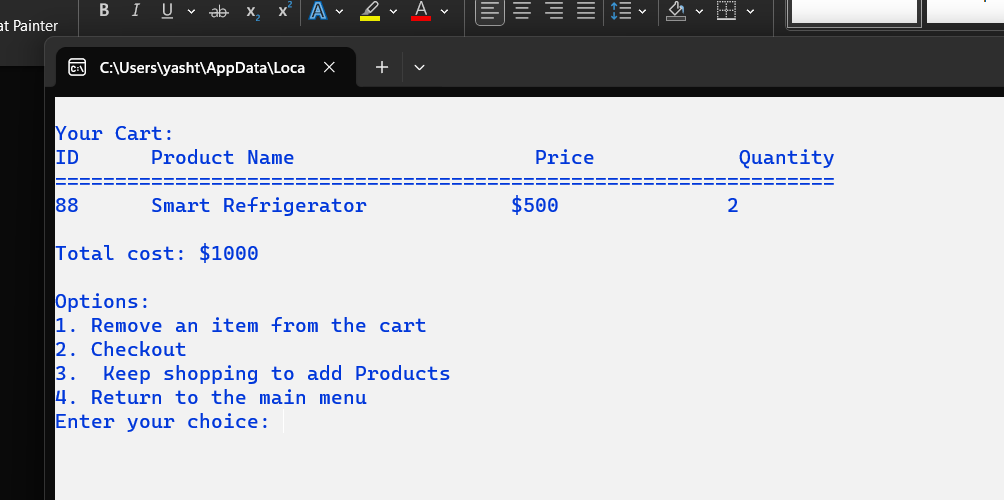




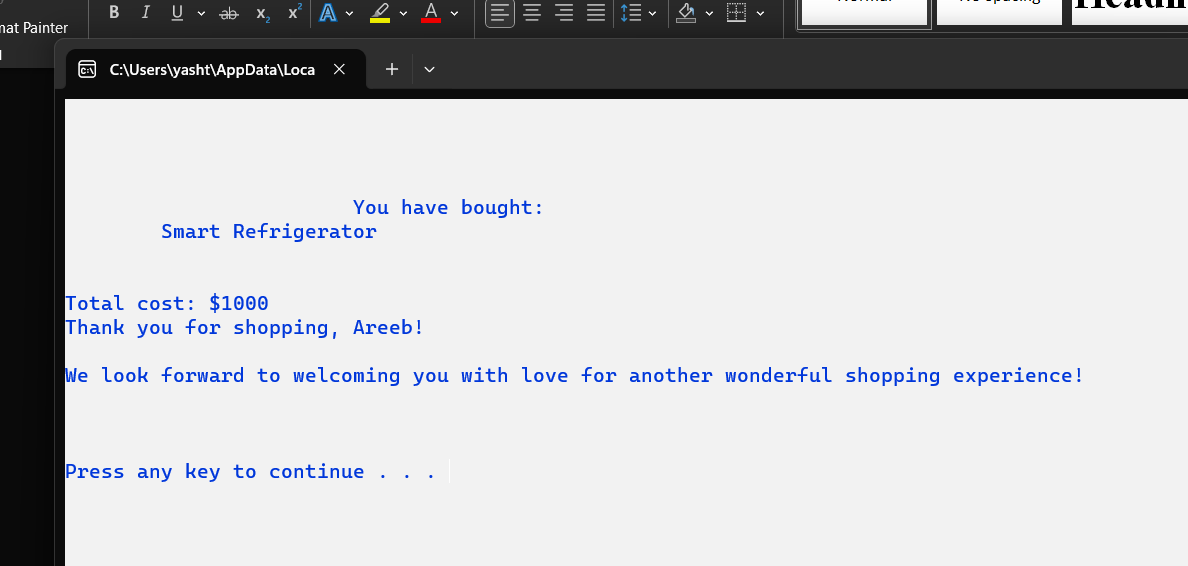




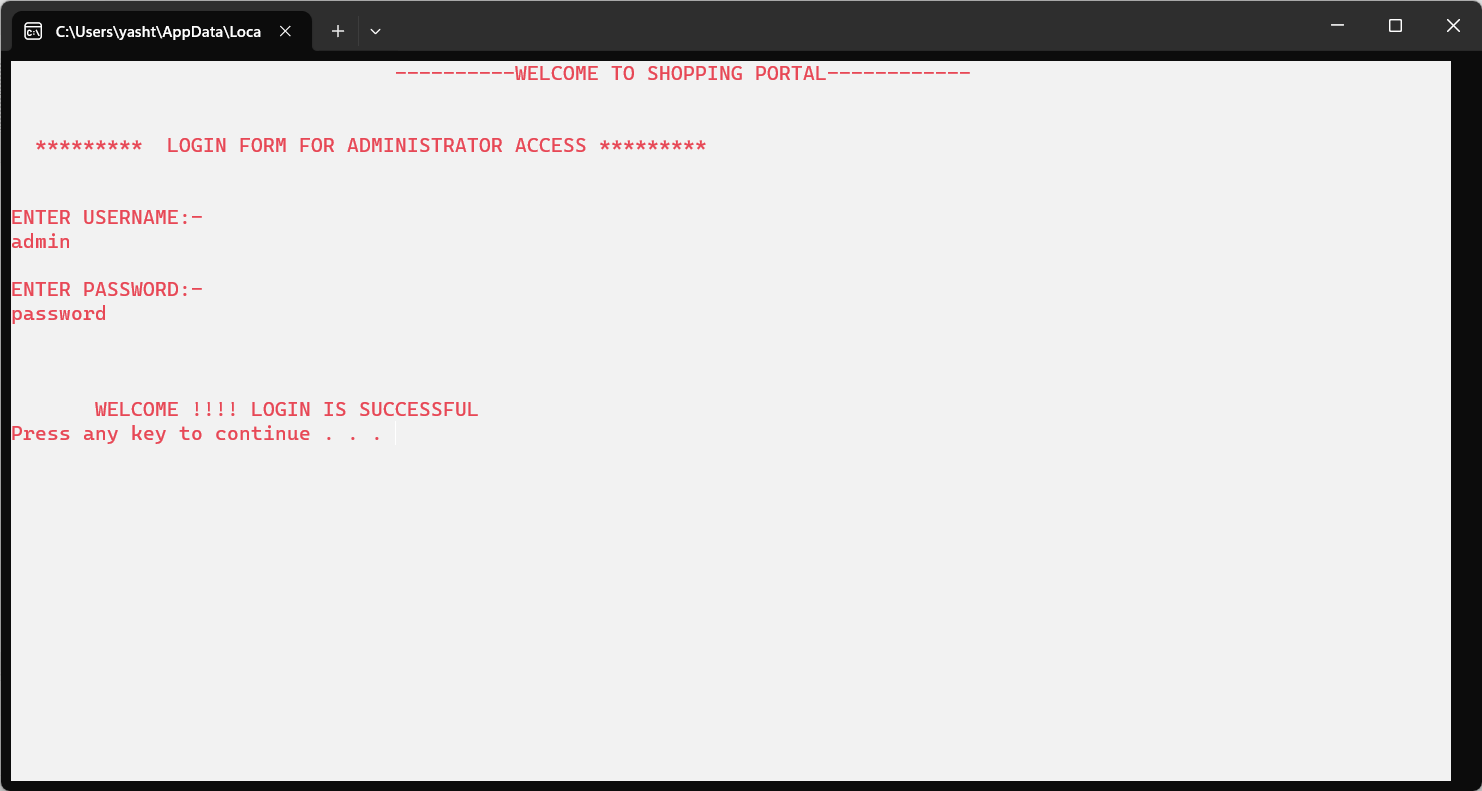
After removing an item:

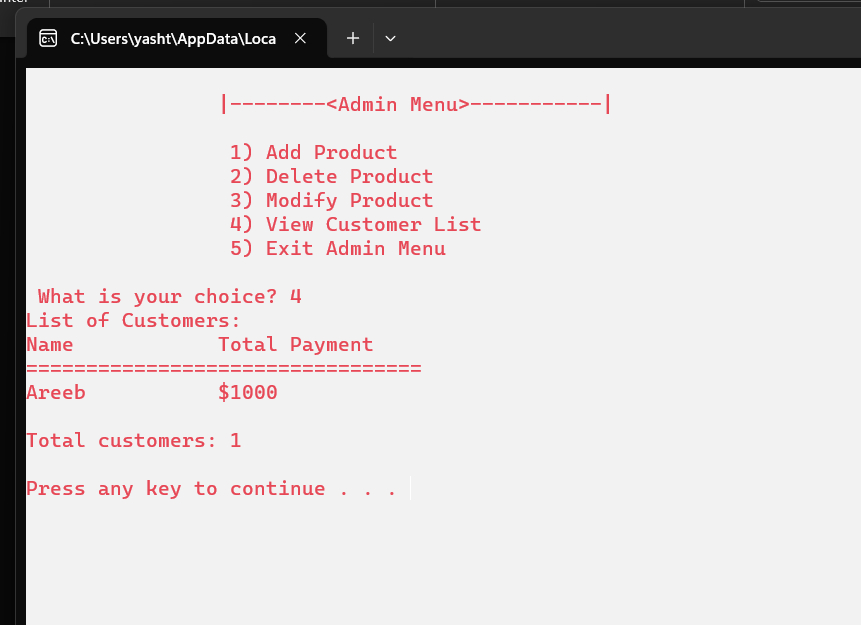


* **Checkout Process**: Computes the total payment and concludes the shopping session.



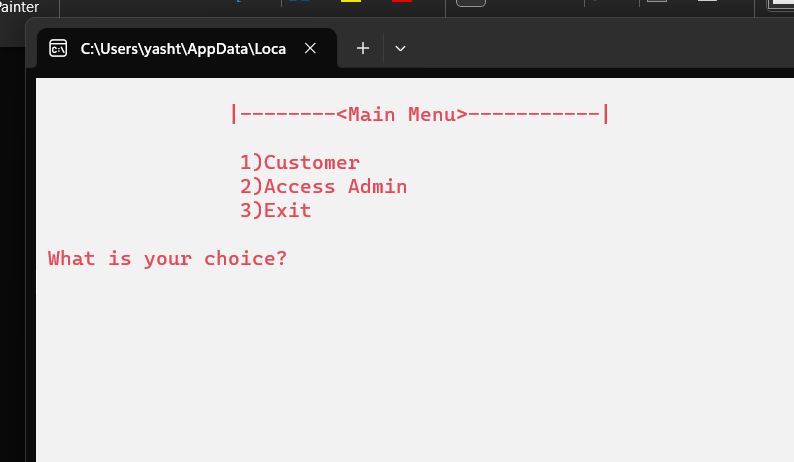
* **Administrator Interface**: Secured functionalities for product and customer list management.





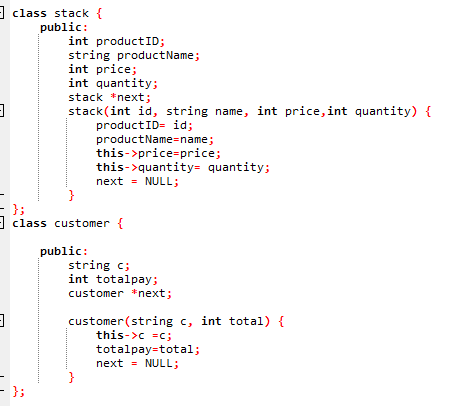
**4. User Interface**

* A console-based interface offering straightforward, menu-driven interactions for both customers and administrators.

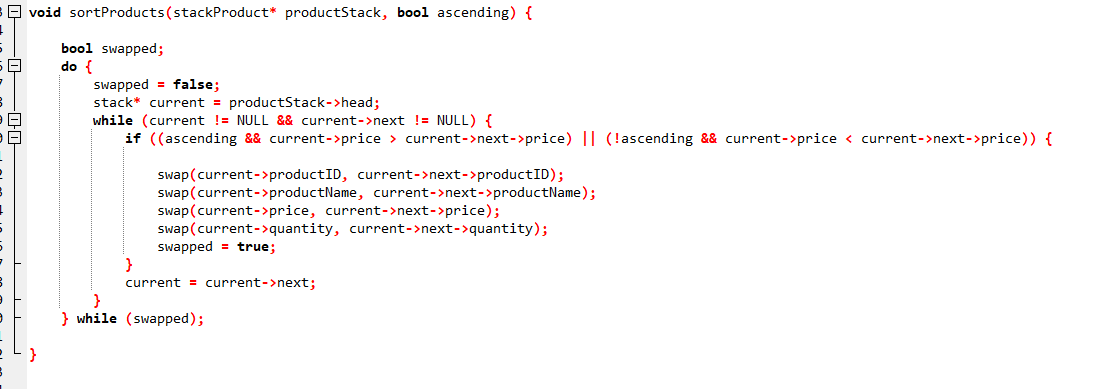


**5. Technical Implementation**

* **Efficient Data Management**: Leveraging linked lists and stack to handle dynamic data efficiently.



* **Sorting Mechanism**: Implementing sorting algorithms to arrange products by price.



**Challenges and Resolutions**

* **Dynamic Data Handling**: Addressed by the flexible nature of linked lists and stack, allowing for efficient management of an ever-changing product list and customer transactions.
* **Intuitive User Experience**: Ensuring a seamless user journey through a well-designed console interface.
* **Data Integrity and Security**: Employing input validation and secure login mechanisms for administrative tasks.

**Conclusion**

The Shopping Cart System, with its core focus on linked lists and stack, demonstrates the power and versatility of these data structures in managing dynamic data sets. This project successfully creates a realistic and efficient shopping environment, highlighting the team's proficiency in applying complex data structures in C++ to solve real-world problems.