COMSATS University Islamabad

Attock Campus

Department Of Computer Science

Course	Data Structure
Instructor	Mam Maryam Bukhari
Project	**Coca-Cola Inventory Management System**

Group Members Details

Registration No.	**Name**
FA23-BSE-059	Taimoor Shaikh

Data Structures and Algorithms

Lab Project Topic:

Coke-Cola Inventory Management System.

Introduction:

This report documents the Coca-Cola Inventory Management System, a C++ program designed to manage product inventory, warehouse operations, and delivery requests. The system utilizes multiple data structures to efficiently handle different aspects of inventory management, including:

- Linked List for product storage
- Stack for warehouse carton management (LIFO)
- Queue for delivery requests (FIFO)
- Hash Table for fast product lookup

The system provides a menu-driven interface for users to add products, manage warehouse cartons, process deliveries, and search for products.

System Overview:

Key Features

1. **Product Management**
   - Add new products with details (code, name, batch, expiry, quantity).
   - Display all products in the inventory.
   - Search for products using a unique product code.
2. **Warehouse Operations**
   - Add cartons to the warehouse (stack-based LIFO system).
   - Dispatch cartons from the warehouse (last added, first removed).
3. **Delivery Processing**
   - Add customer delivery requests (queue-based FIFO system).
   - Process delivery requests in the order they were received.
4. **Fast Product Lookup**
   - Uses a hash table for **O(1) search time** by product code.

Data Structures Used & Justification:

Data Structure	**Purpose**	**Reason for Use**
Linked List (ProductList)	Stores product details dynamically.	Flexible for frequent insertions/deletions.
Stack (warehouse)	Manages cartons in the warehouse.	Follows LIFO (Last-In-First-Out), mimicking real-world warehouse operations where the newest stock is dispatched first.
Queue (deliveryRequests)	Handles customer delivery requests.	Follows FIFO (First-In-First-Out), ensuring fair processing of orders.

Hash Table (productLookup)	Enables fast product searches by code.	Provides O(1) average-case lookup time, improving efficiency.

Code:

Code

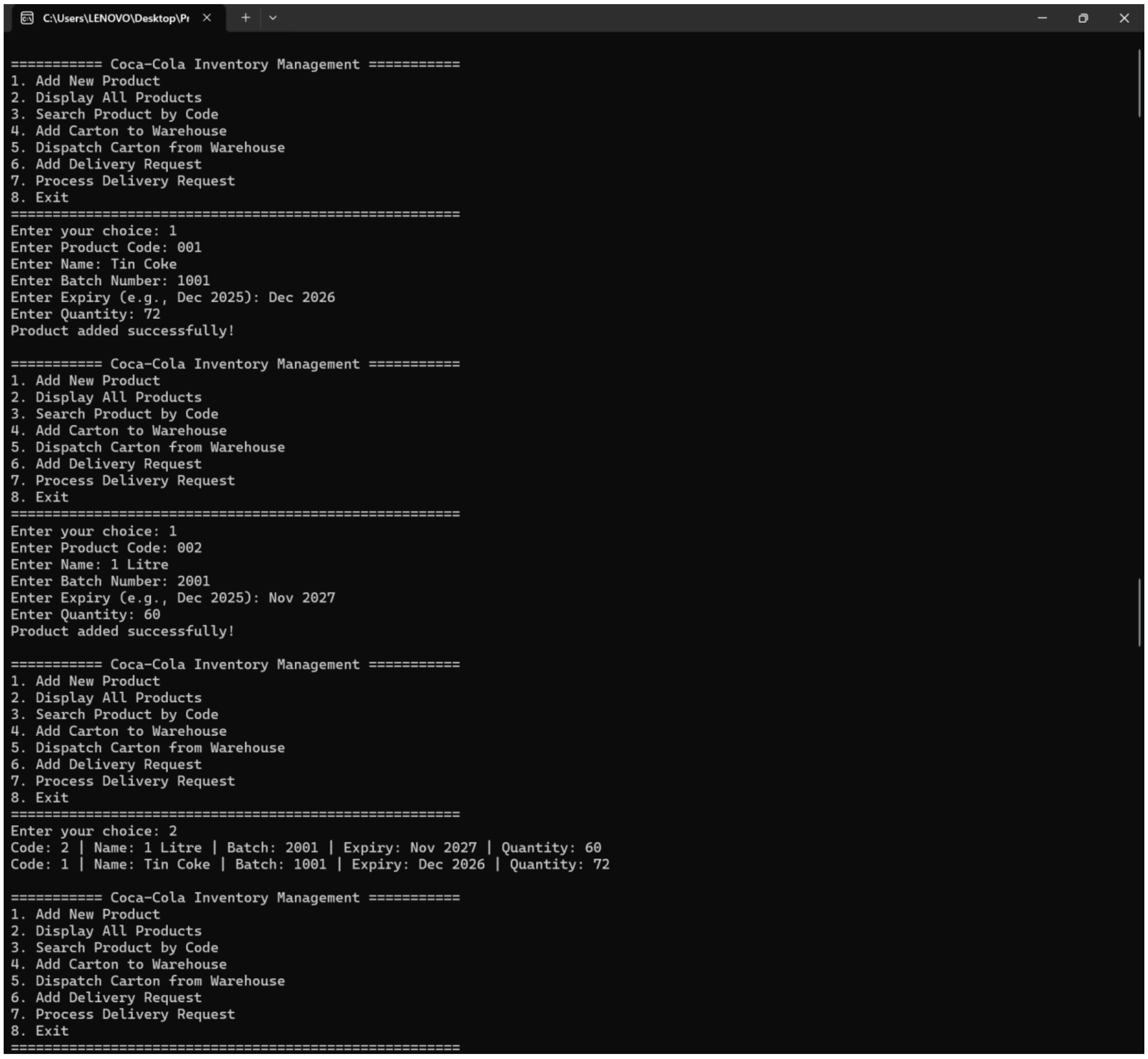
```
#include <iostream>
#include <stack>
#include <queue>
#include <unordered_map>
#include <string>
using namespace std;

// ------------------- Linked List for Product Details --------------------
struct ProductNode {
    int productCode;
    string name;
    int batch;
    string expiry;
    int quantity;
    ProductNode* next;

    ProductNode(int code, string n, int b, string e, int q) {
        productCode = code;
        name = n;
        batch = b;
        expiry = e;
        quantity = q;
        next = nullptr;
    }
};

class ProductList {
public:
    ProductNode* head;

    ProductList() {
        head = nullptr;
    }

    void addProduct(int code, string name, int batch, string expiry, int quantity) {
        ProductNode* newNode = new ProductNode(code, name, batch, expiry, quantity);
        newNode->next = head;
        head = newNode;
        cout << "Product added successfully!\n";
    }

    void displayProducts() {
        if (head == nullptr) {
            cout << "No products in the list.\n";
            return;
        }
        ProductNode* temp = head;
        while (temp != nullptr) {
            cout << "Code: " << temp->productCode
                 << " | Name: " << temp->name
                 << " | Batch: " << temp->batch
                 << " | Expiry: " << temp->expiry
                 << " | Quantity: " << temp->quantity << endl;
            temp = temp->next;
        }
    }
};

// ------------------- Stack for Warehouse (LIFO) --------------------
stack<string> warehouse;

void addCartonToWarehouse(string cartonID) {
    warehouse.push(cartonID);
    cout << "Carton " << cartonID << " added to warehouse.\n";
}

void dispatchCarton() {
    if (!warehouse.empty()) {
        cout << "Dispatched carton: " << warehouse.top() << endl;
        warehouse.pop();
    } else {
        cout << "Warehouse is empty!\n";
    }
}

// ------------------- Queue for Delivery Requests (FIFO) --------------------
queue<string> deliveryRequests;

void addDeliveryRequest(string customerName) {
    deliveryRequests.push(customerName);
    cout << "Delivery request added for: " << customerName << endl;
}
```

```
void processDeliveryRequest() {
    if (!deliveryRequests.empty()) {
        cout << "Processing delivery for: " << deliveryRequests.front() << endl;
        deliveryRequests.pop();
    } else {
        cout << "No delivery requests!\n";
    }
}

// ------------------ Hash Table for Fast Product Lookup -------------------
unordered_map<int, ProductNode*> productLookup;

void buildProductLookup(ProductList& productList) {
    productLookup.clear();
    ProductNode* temp = productList.head;
    while (temp != nullptr) {
        productLookup[temp->productCode] = temp;
        temp = temp->next;
    }
}

void searchProductByCode(int code) {
    if (productLookup.find(code) != productLookup.end()) {
        ProductNode* p = productLookup[code];
        cout << "Product Found: " << p->name
             << " | Batch: " << p->batch
             << " | Expiry: " << p->expiry
             << " | Quantity: " << p->quantity << endl;
    } else {
        cout << "Product with code " << code << " not found!\n";
    }
}

// ------------------ Main with Menu -------------------
int main() {
    ProductList productList;
    int choice;

    do {
        cout << "\n========== Coca-Cola Inventory Management ==========\n";
        cout << "1. Add New Product\n";
        cout << "2. Display All Products\n";
        cout << "3. Search Product by Code\n";
        cout << "4. Add Carton to Warehouse\n";
        cout << "5. Dispatch Carton from Warehouse\n";
        cout << "6. Add Delivery Request\n";
        cout << "7. Process Delivery Request\n";
        cout << "8. Exit\n";
        cout << "============================================================\n";
        cout << "Enter your choice: ";
        cin >> choice;

        switch (choice) {
        case 1: {
            int code, batch, qty;
            string name, expiry;
            cout << "Enter Product Code: ";
            cin >> code;
            cout << "Enter Name: ";
            cin.ignore();
            getline(cin, name);
            cout << "Enter Batch Number: ";
            cin >> batch;
            cout << "Enter Expiry (e.g., Dec 2025): ";
            cin.ignore();
            getline(cin, expiry);
            cout << "Enter Quantity: ";
            cin >> qty;
            productList.addProduct(code, name, batch, expiry, qty);
            buildProductLookup(productList); // Update hash table
            break;
        }
        case 2:
            productList.displayProducts();
            break;
        case 3: {
            int code;
            cout << "Enter Product Code to Search: ";
            cin >> code;
            searchProductByCode(code);
            break;
        }
        case 4: {
            string id;
            cout << "Enter Carton ID: ";
            cin.ignore();
            getline(cin, id);
            addCartonToWarehouse(id);
            break;
        }
        case 5:
            dispatchCarton();
```

```
            break;
        case 6: {
            string customer;
            cout << "Enter Customer Name: ";
            cin.ignore();
            getline(cin, customer);
            addDeliveryRequest(customer);
            break;
        }
        case 7:
            processDeliveryRequest();
            break;
        case 8:
            cout << "Exiting... Thank you!\n";
            break;
        default:
            cout << "Invalid choice. Try again.\n";
        }
    } while (choice != 8);

    return 0;
}
```

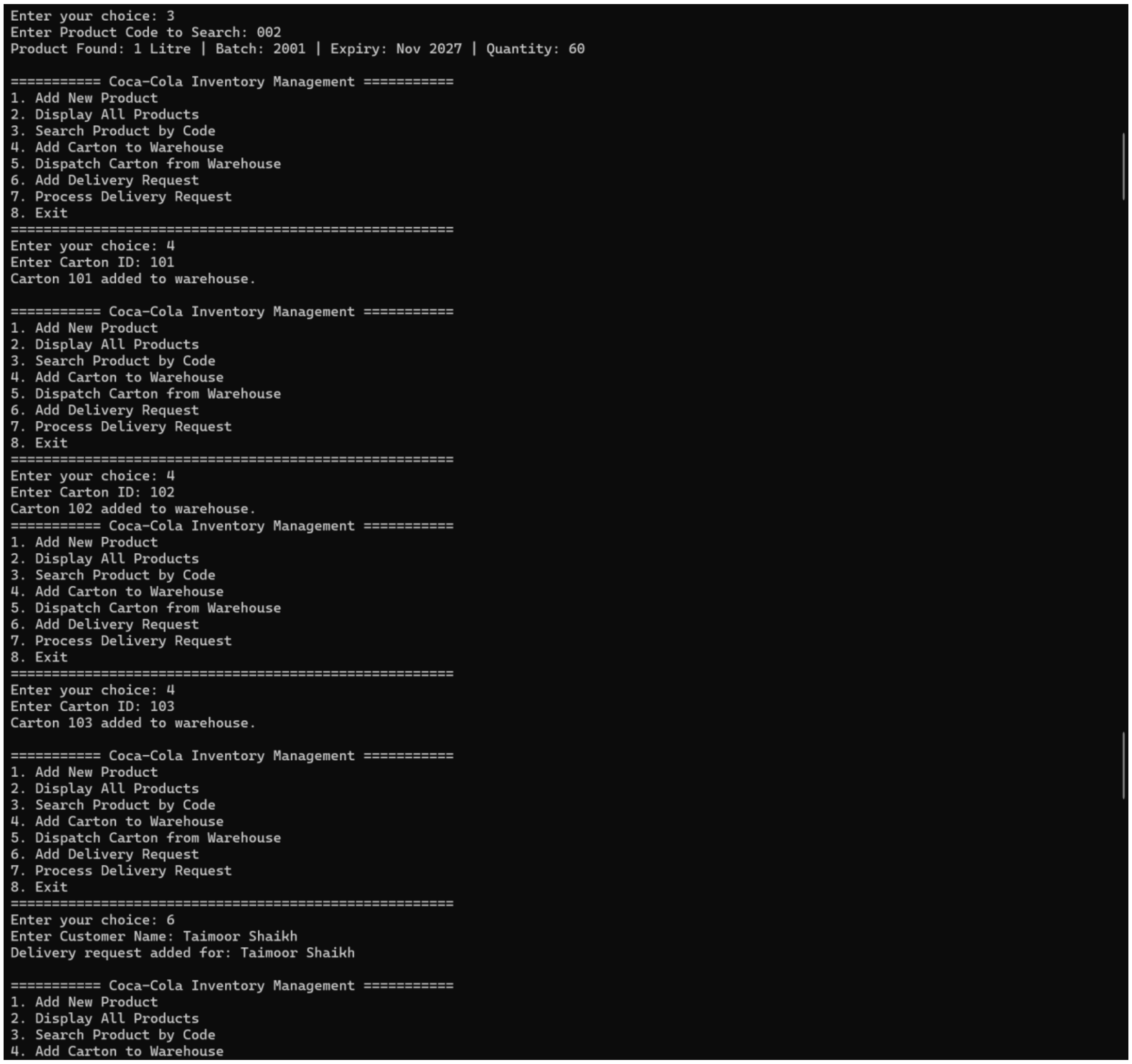
Output:

Output

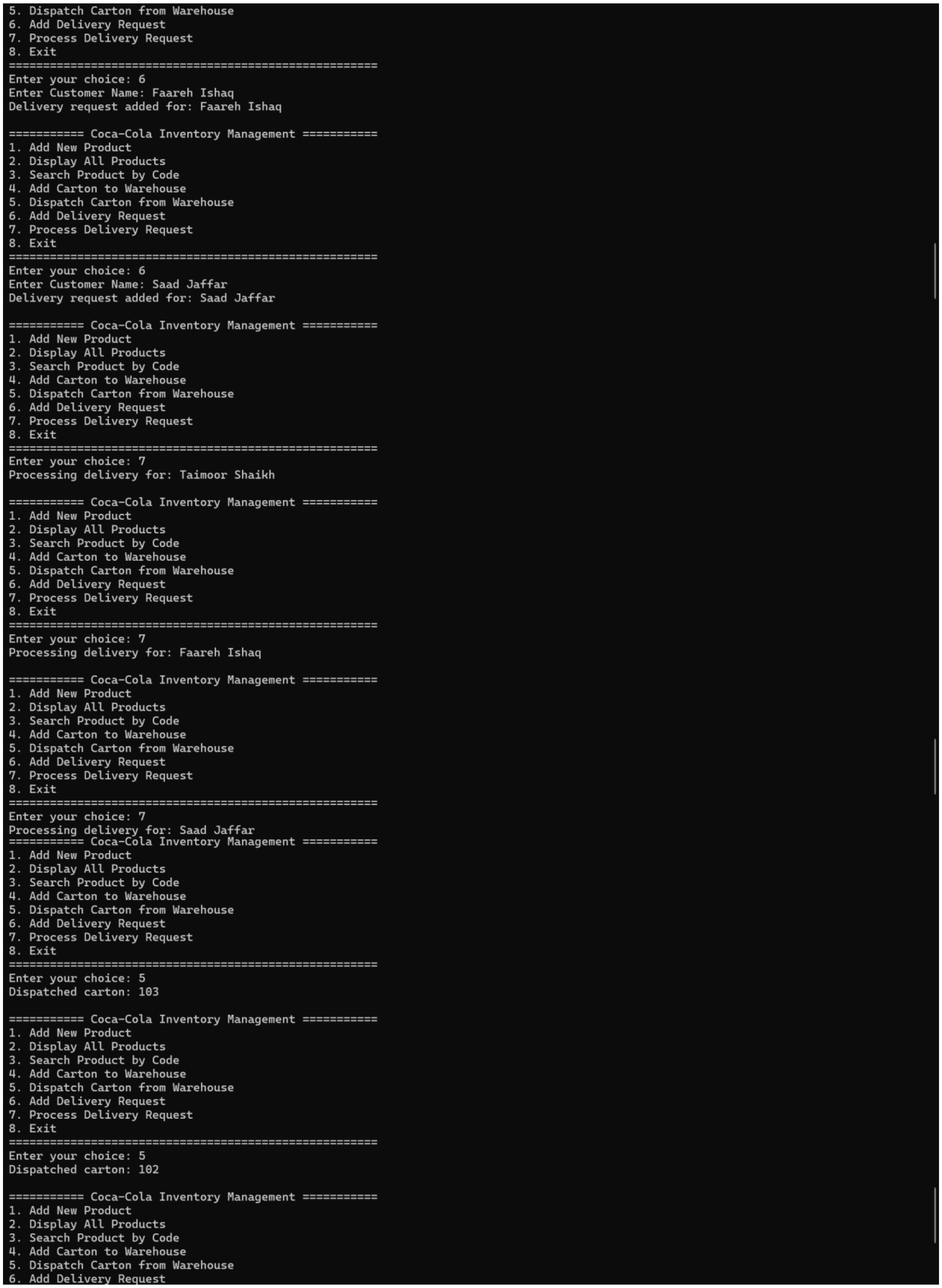


C:\Users\LENOVO\Desktop\Pr

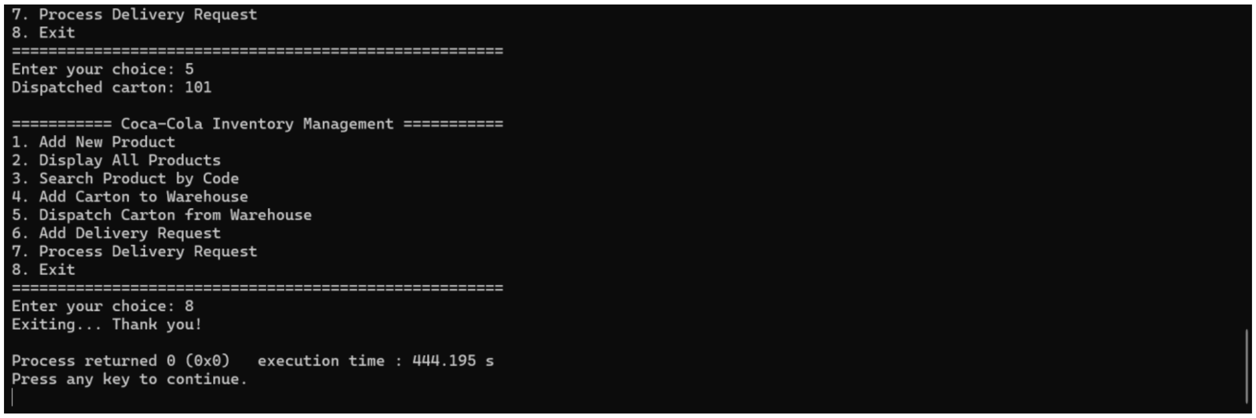
```
=========== Coca-Cola Inventory Management ===========
1. Add New Product
2. Display All Products
3. Search Product by Code
4. Add Carton to Warehouse
5. Dispatch Carton from Warehouse
6. Add Delivery Request
7. Process Delivery Request
8. Exit
======================================================
Enter your choice: 1
Enter Product Code: 001
Enter Name: Tin Coke
Enter Batch Number: 1001
Enter Expiry (e.g., Dec 2025): Dec 2026
Enter Quantity: 72
Product added successfully!

=========== Coca-Cola Inventory Management ===========
1. Add New Product
2. Display All Products
3. Search Product by Code
4. Add Carton to Warehouse
5. Dispatch Carton from Warehouse
6. Add Delivery Request
7. Process Delivery Request
8. Exit
======================================================
Enter your choice: 1
Enter Product Code: 002
Enter Name: 1 Litre
Enter Batch Number: 2001
Enter Expiry (e.g., Dec 2025): Nov 2027
Enter Quantity: 60
Product added successfully!

=========== Coca-Cola Inventory Management ===========
1. Add New Product
2. Display All Products
3. Search Product by Code
4. Add Carton to Warehouse
5. Dispatch Carton from Warehouse
6. Add Delivery Request
7. Process Delivery Request
8. Exit
======================================================
Enter your choice: 2
Code: 2 | Name: 1 Litre | Batch: 2001 | Expiry: Nov 2027 | Quantity: 60
Code: 1 | Name: Tin Coke | Batch: 1001 | Expiry: Dec 2026 | Quantity: 72

=========== Coca-Cola Inventory Management ===========
1. Add New Product
2. Display All Products
3. Search Product by Code
4. Add Carton to Warehouse
5. Dispatch Carton from Warehouse
6. Add Delivery Request
7. Process Delivery Request
8. Exit
======================================================
```



```
Enter your choice: 3
Enter Product Code to Search: 002
Product Found: 1 Litre | Batch: 2001 | Expiry: Nov 2027 | Quantity: 60

=========== Coca-Cola Inventory Management ===========
1. Add New Product
2. Display All Products
3. Search Product by Code
4. Add Carton to Warehouse
5. Dispatch Carton from Warehouse
6. Add Delivery Request
7. Process Delivery Request
8. Exit
======================================================
Enter your choice: 4
Enter Carton ID: 101
Carton 101 added to warehouse.

=========== Coca-Cola Inventory Management ===========
1. Add New Product
2. Display All Products
3. Search Product by Code
4. Add Carton to Warehouse
5. Dispatch Carton from Warehouse
6. Add Delivery Request
7. Process Delivery Request
8. Exit
======================================================
Enter your choice: 4
Enter Carton ID: 102
Carton 102 added to warehouse.
=========== Coca-Cola Inventory Management ===========
1. Add New Product
2. Display All Products
3. Search Product by Code
4. Add Carton to Warehouse
5. Dispatch Carton from Warehouse
6. Add Delivery Request
7. Process Delivery Request
8. Exit
======================================================
Enter your choice: 4
Enter Carton ID: 103
Carton 103 added to warehouse.

=========== Coca-Cola Inventory Management ===========
1. Add New Product
2. Display All Products
3. Search Product by Code
4. Add Carton to Warehouse
5. Dispatch Carton from Warehouse
6. Add Delivery Request
7. Process Delivery Request
8. Exit
======================================================
Enter your choice: 6
Enter Customer Name: Taimoor Shaikh
Delivery request added for: Taimoor Shaikh

=========== Coca-Cola Inventory Management ===========
1. Add New Product
2. Display All Products
3. Search Product by Code
4. Add Carton to Warehouse
```



```
5. Dispatch Carton from Warehouse
6. Add Delivery Request
7. Process Delivery Request
8. Exit
=====================================================
Enter your choice: 6
Enter Customer Name: Faareh Ishaq
Delivery request added for: Faareh Ishaq

=========== Coca-Cola Inventory Management ===========
1. Add New Product
2. Display All Products
3. Search Product by Code
4. Add Carton to Warehouse
5. Dispatch Carton from Warehouse
6. Add Delivery Request
7. Process Delivery Request
8. Exit
=====================================================
Enter your choice: 6
Enter Customer Name: Saad Jaffar
Delivery request added for: Saad Jaffar

=========== Coca-Cola Inventory Management ===========
1. Add New Product
2. Display All Products
3. Search Product by Code
4. Add Carton to Warehouse
5. Dispatch Carton from Warehouse
6. Add Delivery Request
7. Process Delivery Request
8. Exit
=====================================================
Enter your choice: 7
Processing delivery for: Taimoor Shaikh

=========== Coca-Cola Inventory Management ===========
1. Add New Product
2. Display All Products
3. Search Product by Code
4. Add Carton to Warehouse
5. Dispatch Carton from Warehouse
6. Add Delivery Request
7. Process Delivery Request
8. Exit
=====================================================
Enter your choice: 7
Processing delivery for: Faareh Ishaq

=========== Coca-Cola Inventory Management ===========
1. Add New Product
2. Display All Products
3. Search Product by Code
4. Add Carton to Warehouse
5. Dispatch Carton from Warehouse
6. Add Delivery Request
7. Process Delivery Request
8. Exit
=====================================================
Enter your choice: 7
Processing delivery for: Saad Jaffar
=========== Coca-Cola Inventory Management ===========
1. Add New Product
2. Display All Products
3. Search Product by Code
4. Add Carton to Warehouse
5. Dispatch Carton from Warehouse
6. Add Delivery Request
7. Process Delivery Request
8. Exit
=====================================================
Enter your choice: 5
Dispatched carton: 103

=========== Coca-Cola Inventory Management ===========
1. Add New Product
2. Display All Products
3. Search Product by Code
4. Add Carton to Warehouse
5. Dispatch Carton from Warehouse
6. Add Delivery Request
7. Process Delivery Request
8. Exit
=====================================================
Enter your choice: 5
Dispatched carton: 102

=========== Coca-Cola Inventory Management ===========
1. Add New Product
2. Display All Products
3. Search Product by Code
4. Add Carton to Warehouse
5. Dispatch Carton from Warehouse
6. Add Delivery Request
```



```
7. Process Delivery Request
8. Exit
==================================================
Enter your choice: 5
Dispatched carton: 101

=========== Coca-Cola Inventory Management ===========
1. Add New Product
2. Display All Products
3. Search Product by Code
4. Add Carton to Warehouse
5. Dispatch Carton from Warehouse
6. Add Delivery Request
7. Process Delivery Request
8. Exit
==================================================
Enter your choice: 8
Exiting... Thank you!

Process returned 0 (0x0)   execution time : 444.195 s
Press any key to continue.
```

Limitations:

1. **No Data Persistence**
   - All data is lost when the program exits (no file/database storage).
2. **No Input Validation**
   - The program does not check for duplicate product codes or invalid inputs.
3. **Basic Error Handling**
   - Limited error recovery (e.g., no retry mechanism for invalid inputs).
4. **No Multithreading**
   - Not suitable for concurrent operations in a real-world warehouse.

Future Enhancements:

1. **Database Integration**
   - Store product and delivery data in **SQLite/MySQL** for persistence.
2. **Input Validation**
   - Prevent duplicate product codes and invalid entries.
3. **Advanced Search Features**
   - Search by name, batch, or expiry date (currently only by code).
4. **User Authentication**
   - Add login roles (Admin, Warehouse Staff, Delivery Personnel).
5. **GUI Implementation**
   - Develop a **Qt-based** or **web-based** frontend for better usability.
6. **Automated Expiry Alerts**
   - Notify users when products are near expiry.

Conclusion

This **Coca-Cola Inventory Management System** demonstrates the effective use of **linked lists, stacks, queues, and hash tables** to manage inventory, warehouse operations, and deliveries efficiently. While functional, it can be expanded with **persistent storage, input validation, and a GUI** for real-world deployment.