National Textile University,

Faisalabad

A logo with a shield and text

Description automatically generated

Department of Computer Science

|  |  |
| --- | --- |
| Class: | BS AI 4th Semester |
| Project Proposal | Grocery Management System |
| Course Code: | COC-2073 |
| Course Name: | Data Structures and Algorithms |
| Submitted To: | *Dr. Asif and Mr.Basit* |
| Submission Date: | June 09,2024 |

# Group Members:

Muhammad Hassaan Raza 22-NTU-CS-1362

Kanza Kashaf 22-NTU-CS-1350

Grocery Management System

Introduction:

The Grocery Management System is designed for efficient management of a grocery mart. The system will provide functionality for cashier login, mart manager login, product management, receipt generation, customer management, stock management, account management, and receipt retrieval. This proposal outlines the software's appearance, flow, and features to effectively manage grocery store operations.

# Functional Requirements:

## Manager Portal

### 1.Manage Stock:

* **Add Product**
* **Remove Product:** Enable the removal of obsolete or expired products.
* **Update Product:** Modify product details such as price, quantity, and other relevant information.
* **View Products:** Display all products in the inventory.

### 2.View Receipts:

* **All Receipts:** List all the receipts generated by the store.
* **Receipt by ID:** Search and view specific receipts using receipt ID.
* **Receipt by Customer Name:** Search and view receipts by customer name.
* **Customer Purchase History:** Display the purchase history of a specific customer.

### 3.Manage Accounts:

* **Sales:** Display date wise sales.

## Cashier Portal

### 1.Take Customer Orders:

* **Generate Receipt:** Create receipts for customer purchases, including calculating totals and applying discounts.
* **Search Receipts:** Look up receipts by ID for customer queries.

## View Receipts:

**All Receipts:** Display all receipts generated.

**Receipt by ID:** Search and view specific receipts using receipt ID

# Data Structures and Algorithms:

Doubly Linked Lists:

For managing dynamic data where Insertions and deletions required such as Product/Receipt insertions and deletions.

# Strategy for Storing Data

For data storage, the system will utilize file-based storage using file handling concepts in C++. Relevant data such as product details, customer information, and receipts will be stored in separate files. File I/O operations will ensure data consistency and integrity.

## Loading Data & Writing Data:

Loading Product,Reciept and customer data from files to LIST for dynamic CRUD operations.Similarly Writing data from LISTs datastructure

#include <iostream>

#include <cstring>

#include <conio.h>

#include <iomanip>

#include <windows.h>

#include "ProductLinkedList.h"

#include "ReceiptLinkedList.h"

#include "BUYINGPRODUCTLINKEDLIST.h"

using namespace std;

## These are the standard and custom header files included in the program.

# Class Declaration: Management

### Functionality Description

* addProduct(): Allows adding a new product with details like ID, name, price, buy cost, and quantity.
* removeProduct(): Facilitates removing a product based on its ID.
* updateProduct(): Enables updating details of an existing product, like name, price, buy cost, and quantity.
* printAllProducts(): Prints details of all products available in the system.
* printProductById(): Prints details of a specific product identified by its ID.
* printProductByName(): Prints details of a specific product identified by its name.
* isQuantityAvailable(): Checks if a specified quantity of a product is available in stock.
* addReceipt():This function is responsible for adding a new receipt to the system. It prompts the user to enter customer details and select products to include in the receipt. It calculates the total bill amount, applies discounts based on loyalty program eligibility, generates a unique ID for the receipt, and adds it to the system.

# Usage of Linked Lists

This class utilizes linked lists (ProductLinkedList and ReceiptLinkedList) for managing products and receipts, respectively.

class Management

{

private:

    ProductLinkedList products;

    ReceiptLinkedList receipts;

public:

    Management()

    {

        products.loadProduct();

        receipts.loadReceipt();

    }

    void addProduct()

    {

        cls();

        Product temp;

        cout << "\t\t==================================================\n";

        cout << "\t\t\t\t Adding Product \n";

        cout << "\t\t==================================================\n\n";

        cout << "\t\t---------------------------------------------------\n";

        while (1)

        {

            cout << "\t\t\tEnter product Id: ";

            cin >> temp.id;

            if (products.isIdAvailable(temp.id))

            {

                break;

            }

            system("Color 7C");

            cout << "Id is already available!" << endl;

        }

        cls();

        system("Color 71");

        cout << "\t\t==================================================\n";

        cout << "\t\t\t\t Adding Product \n";

        cout << "\t\t==================================================\n\n";

        cout << "\t\t---------------------------------------------------\n";

        cout << "\t\t\tEnter product Id: " << temp.id << endl;

        cout << "\t\t\tEnter Product Name: ";

        cin >> temp.name;

        cout << "\t\t\tEnter Product Price: ";

        cin >> temp.price;

        cout << "\t\t\tEnter Product Buy Cost: ";

        cin >> temp.buyCost;

        cout << "\t\t\tEnter Product Quantity: ";

        cin >> temp.quantity;

        cout << "\t\t---------------------------------------------------\n";

        products.insertAtEnd(temp);

        products.writeProducts();

    }

    void removeProduct()

    {

        string id;

        cls();

        cout << "\n\n\t\tEnter Id of Product to Remove: ";

        cin >> id;

        ProductNode \*delNode = products.FindNodeById(id);

        if (delNode == NULL)

        {

            system("Color 7C");

            cout << "Product not found!" << endl;

            return;

        }

        products.removeNode(delNode);

        products.writeProducts();

        cout << "\n\n\t\t  Successfully Removed!\n"

             << endl;

    }

    void updateProduct()

    {

        cls();

        string id;

        cout << "\n\nEnter Id of Product to Update: ";

        cin >> id;

        ProductNode \*updateNode = products.FindNodeById(id);

        if (updateNode == NULL)

        {

            system("Color 7C");

            cout << "Product not found!" << endl;

            return;

        }

        Product temp = updateNode->GetData();

        int selectedChoice;

        while (1)

        {

            system("Color 71");

            cout << "\n\t\t==================================================\n";

            cout << "\t\t\t\tProduct Details                " << endl;

            cout << "\t\t==================================================\n\n";

            cout << "\t\t--------------------------------------------------\n";

            cout << "\t\t\tProduct ID: " << temp.id << endl;

            cout << "\t\t\tProduct Name: " << temp.name << endl;

            cout << "\t\t\tProduct Price: " << temp.price << endl;

            cout << "\t\t\tProduct Buy Cost: " << temp.buyCost << endl;

            cout << "\t\t\tProduct Quantity: " << temp.quantity << endl;

            cout << "\t\t--------------------------------------------------\n"

                 << endl;

            cout << "\n\t\t==================================================\n";

            cout << "\t\t\t\tUpdating Choices                " << endl;

            cout << "\t\t==================================================\n\n";

            cout << "\t\t---------------------------------------------------\n";

            cout << "\t\t||\t1. Change Name\t\t\t\t ||" << endl;

            cout << "\t\t||\t2. Change Price\t\t\t\t ||" << endl;

            cout << "\t\t||\t3. Change Buy Cost\t\t\t ||" << endl;

            cout << "\t\t||\t4. Change Quantity\t\t\t ||" << endl;

            cout << "\t\t||\t5. Save & Exit\t\t\t\t ||" << endl;

            cout << "\t\t||\t6. Exit without save\t\t\t ||" << endl;

            cout << "\t\t---------------------------------------------------\n\n";

            cout << "Enter your Choice: ";

            cin >> selectedChoice;

            cls();

            switch (selectedChoice)

            {

            case 1:

                cout << "\t\t==================================================\n";

                cout << "\t\t\t\t Changing Name \n";

                cout << "\t\t==================================================\n\n";

                cout << "\t\t---------------------------------------------------\n";

                cout << "\n\t\t\tEnter Name: ";

                cin >> temp.name;

                break;

            case 2:

                cout << "\t\t==================================================\n";

                cout << "\t\t\t\t Changing Price \n";

                cout << "\t\t==================================================\n\n";

                cout << "\t\t---------------------------------------------------\n";

                cout << "\n\t\t\tEnter Price: ";

                cin >> temp.price;

                break;

            case 3:

                cout << "\t\t==================================================\n";

                cout << "\t\t\t\t Changing Buy Cost \n";

                cout << "\t\t==================================================\n\n";

                cout << "\t\t---------------------------------------------------\n";

                cout << "\n\t\t\tEnter Buy Cost: ";

                cin >> temp.buyCost;

                break;

            case 4:

                cout << "\t\t==================================================\n";

                cout << "\t\t\t\t Changing Quantity \n";

                cout << "\t\t==================================================\n\n";

                cout << "\t\t---------------------------------------------------\n";

                cout << "\n\t\t\tEnter Quantity: ";

                cin >> temp.quantity;

                break;

            case 5:

                updateNode->SetData(temp);

                cout << "\n\n\n\t\tSuccessfully updated!\n"

                     << endl;

                return;

            case 6:

                return;

            default:

                system("Color 7C");

                cout << "\n\n\t\tPlease Enter a valid option" << endl;

                system("pause");

            }

            cls();

        }

        products.writeProducts();

    }

    void printAllProducts()

    {

        cout << "\t\t==================================================\n";

        cout << "\t\t\t\tViewing All Products \n";

        cout << "\t\t==================================================\n\n";

        cout << "\t\t--------------------------------------------------\n\n";

        cout << "--------------------------------------------------------------------------------\n";

        cout << "Sr.#\tId\tProduct Name\t\tPrice\t\tBuy Cost\tQuantity\n\n";

        cout << "--------------------------------------------------------------------------------\n";

        products.PrintAllProducts();

        cout << endl

             << endl;

        pause();

        // print all products!

    }

    void printProductById()

    {

        string id;

        cout << "\t\t==================================================\n";

        cout << "\t\t\t\tViewing Product By ID \n";

        cout << "\t\t==================================================\n\n";

        cout << "\n\nEnter Id of product : ";

        cin >> id;

        ProductNode \*productNode = products.FindNodeById(id);

        if (productNode == NULL)

        {

            system("Color 7C");

            cout << "Product not found!" << endl;

            return;

        }

        Product product = productNode->GetData();

        cout << "\n\t\t--------------------------------------------------\n";

        cout << "\t\t     Product ID: " << product.id << endl;

        cout << "\t\t     Product Name: " << product.name << endl;

        cout << "\t\t     Product Price: " << product.price << endl;

        cout << "\t\t     Product Buy Cost: " << product.buyCost << endl;

        cout << "\t\t     Product Quantity: " << product.quantity << endl;

        cout << "\t\t--------------------------------------------------\n\n\n";

        // print single product by id

    }

    void printProductByName()

    {

        string name;

        cout << "\t\t==================================================\n";

        cout << "\t\t\t\tViewing Product By Name \n";

        cout << "\t\t==================================================\n\n";

        cout << "\n\nEnter Name of product: ";

        cin >> name;

        ProductNode \*productNode = products.FindByName(name);

        if (productNode == NULL)

        {

            system("Color 7C");

            cout << "Product not found!" << endl;

            return;

        }

        Product product = productNode->GetData();

        cout << "\n\t\t--------------------------------------------------\n";

        cout << "\t\t     Product ID: " << product.id << endl;

        cout << "\t\t     Product Name: " << product.name << endl;

        cout << "\t\t     Product Price: " << product.price << endl;

        cout << "\t\t     Product Buy Cost: " << product.buyCost << endl;

        cout << "\t\t     Product Quantity: " << product.quantity << endl;

        cout << "\t\t--------------------------------------------------\n\n\n";

        // print single product by name

    }

    bool isQuantityAvailable(ProductNode \*productNode, int quantity)

    {

        if (productNode->GetData().quantity >= quantity)

            return true;

        return false;

    }

    void addReceipt()

    {

        Receipt temp;

        int customerCounter = 0;

        string date;

        cout << "\t\t==================================================\n";

        cout << "\t\t\t\t Making Receipt \n";

        cout << "\t\t==================================================\n\n";

        cout << "\t\t---------------------------------------------------\n";

        cout << "\t\t\tEnter Customer Name: ";

        cin >> temp.customerName;

        cout << "\t\t\tEnter Customer Phone Number: ";

        cin >> temp.customerPhoneNumber;

        // adding products to bill/recceipts

        temp.buyingProducts.SetHead(addProductToReceipt());

        // Loyalty program check >=3

        ReceiptNode \*tempNode = receipts.GetHead();

        while (tempNode != NULL)

        {

            if (tempNode->GetData().customerPhoneNumber == temp.customerPhoneNumber)

                customerCounter++;

            tempNode = tempNode->GetNext();

        }

        if (customerCounter >= 3)

            temp.loyaltyStatus = 1;

        else

            temp.loyaltyStatus = 0;

        //////////////////////////////////////////////

        // billAmountWithoutDis

        temp.billAmountWithoutDis = 0;

        BuyingProductNode \*tempBuyingProduct = temp.buyingProducts.GetHead();

        while (tempBuyingProduct != NULL)

        {

            temp.billAmountWithoutDis += tempBuyingProduct->GetData().price \* tempBuyingProduct->GetData().quantity;

            tempBuyingProduct = tempBuyingProduct->GetNext();

        }

        if (temp.loyaltyStatus)

            temp.totalBillAmount = temp.billAmountWithoutDis - temp.billAmountWithoutDis \* 0.1;

        // 0.1=>10% discount if Customer is repeaatdedly coming

        else

            temp.totalBillAmount = temp.billAmountWithoutDis;

        // Date Time return

        dateTimeReceipt(temp);

        sprintf(temp.id, "kk-%03d", receipts.getSize() + 1);

        // pushing into linklist

        receipts.insertAtEnd(temp);

        receipts.writeReceipts();

        cls();

        char choice;

        cout << "\n\n\n\t\t\t\t Receipt Added Successfully!" << endl;

        cout << "\n\n\t\tPress 1 to Generate Receipt or any other key to continue..." << endl;

        cin >> choice;

        if (choice == '1')

            printReceipt(temp);

    }

    void printReceipt(Receipt temp)

    {

        cout << "\n\n\t\t\t\t  K.K Mart" << endl;

        cout << "\t\t==================================================\n\n";

        cout << "\t\t\t\t" << temp.id << endl;

        cout << "\t\tName: " << temp.customerName << "    \t\tPhone: " << temp.customerPhoneNumber << endl;

        cout << "\t\t==================================================\n\n";

        cout << "Sr.#    |    Product Name    |    Price    |    Quantity    |    Total Price\n\n";

        cout << "--------------------------------------------------------------------------------\n";

        BuyingProductNode \*tempNode = temp.buyingProducts.GetHead();

        int i = 0;

        while (tempNode != NULL)

        {

            cout << " " << ++i << "\t\t" << tempNode->GetData().name << "\t\t   " << tempNode->GetData().price

                 << "\t\t   " << tempNode->GetData().quantity << "\t\t    " << tempNode->GetData().quantity \* tempNode->GetData().price << endl;

            tempNode = tempNode->GetNext();

        }

        cout << "--------------------------------------------------------------------------------\n\n";

        cout << "Date: " << temp.date << endl;

        cout << "\t\t\t Amount:\tRs. " << temp.billAmountWithoutDis << endl;

        cout << "\t\t\t Discount:\tRs. ";

        if (temp.loyaltyStatus == 1)

            cout << temp.billAmountWithoutDis \* 0.1 << endl;

        else

            cout << "0" << endl;

        cout << "\t\t\t Total Amount:\tRs. " << temp.totalBillAmount << endl;

        cout << "\n\t\t    Thanks For Shopping Here...!!" << endl

             << endl;

    }

    void dateTimeReceipt(Receipt &r)

    {

        // Get the current time

        time\_t now = time(0);

        // Extract the day, month, and year from the string

        tm \*currentTime = localtime(&now);

        int day = currentTime->tm\_mday;

        int month = currentTime->tm\_mon + 1;

        int year = currentTime->tm\_year + 1900;

        sprintf(r.date, "%02d-%02d-%04d", day, month, year);

    }

    BuyingProductNode \*addProductToReceipt()

    {

        BuyingProduct temp;

        BuyingProductLinkedList tempBuyingProductList;

        int choiceAddProduct;

        bool checkBuy = 0;

        while (1)

        {

            system("Color 71");

            cout << "\t\t\tEnter Product Id: ";

            cin >> temp.id;

            ProductNode \*buyingProduct = products.FindNodeById(temp.id);

            if (buyingProduct != NULL)

            {

                while (true)

                {

                    Product tempProduct = buyingProduct->GetData();

                    system("Color 71");

                    cout << "\t\t\tEnter Quantity: ";

                    cin >> temp.quantity;

                    if (isQuantityAvailable(buyingProduct, temp.quantity))

                    {

                        tempProduct.quantity -= temp.quantity;

                        buyingProduct->SetData(tempProduct);

                        temp.name = tempProduct.name;

                        temp.price = tempProduct.price;

                        tempBuyingProductList.insertAtEnd(temp);

                        checkBuy = 1;

                        break; // Exit the inner loop if the quantity is available

                    }

                    else

                    {

                        char choice;

                        system("Color 7C");

                        cout << "Insufficient Quantity!!! Only " << tempProduct.quantity << " are available." << endl;

                        cout << "\n\tPress 1 to Re-enter quantity or any other key to go back: ";

                        cin >> choice;

                        if (choice != '1')

                            break; // Exit the inner loop if the user chooses not to re-enter the quantity

                    }

                }

                // If the quantity was successfully modified, exit the outer loop

                cout << "\n\tPress 1 to Add More Product! or any other key to Finish ";

                cin >> choiceAddProduct;

                if (choiceAddProduct != 1)

                {

                    return tempBuyingProductList.GetHead();

                    break;

                }

            }

            else

            {

                system("Color 7C");

                cout << "\n\tInvalid product ID. " << endl;

                pause();

            }

        }

    }

    void printAllReceipts()

    {

        cout << "\t\t==================================================\n";

        cout << "\t\t\t K.K Mart All Receipts Records \n";

        cout << "\t\t==================================================\n\n";

        cout << "\t\t---------------------------------------------------\n";

        cout << "--------------------------------------------------------------------------------\n";

        cout << "Sr.#\tId\tCustomer Name\tPhone No.\tDate\tLoyalty Status\tTotal Bill\\n\n";

        cout << "--------------------------------------------------------------------------------\n";

        ReceiptNode \*tempNode = receipts.GetHead();

        int i = 0;

        while (tempNode != NULL)

        {

            cout << " " << ++i << "\t" << tempNode->GetData().id << "\t" << tempNode->GetData().customerName << "\t\t"

                 << tempNode->GetData().customerPhoneNumber << "\t" << tempNode->GetData().date << "\t"

                 << tempNode->GetData().loyaltyStatus << "\t" << tempNode->GetData().totalBillAmount << endl;

            tempNode = tempNode->GetNext();

        }

    }

    void printReceiptById()

    {

        string id;

        cout << "\t\t==================================================\n";

        cout << "\t\t\t\tViewing Receipt By ID \n";

        cout << "\t\t==================================================\n\n";

        cout << "\n\nEnter Id of Receipt : ";

        cin >> id;

        ReceiptNode \*recieptNode = receipts.FindNodeById(id);

        if (recieptNode == NULL)

        {

            system("Color 7C");

            cout << "Receipt not found!" << endl;

            return;

        }

        Receipt temp = recieptNode->GetData();

        cout << "\n\n\t\t\t\t  K.K Mart" << endl;

        cout << "\t\t==================================================\n\n";

        cout << "\t\t\t\t" << temp.id << endl;

        cout << "\t\tName: " << temp.customerName << "    \t\tPhone: " << temp.customerPhoneNumber << endl;

        cout << "\t\t==================================================\n\n";

        cout << "Sr.#    |    Product Name    |    Price    |    Quantity    |    Total Price\n\n";

        cout << "--------------------------------------------------------------------------------\n";

        BuyingProductNode \*tempNode = temp.buyingProducts.GetHead();

        int i = 0;

        while (tempNode != NULL)

        {

            cout << " " << ++i << "\t\t" << tempNode->GetData().name << "\t\t   " << tempNode->GetData().price << "\t\t   "

                 << tempNode->GetData().quantity << "\t\t    " << tempNode->GetData().quantity \* tempNode->GetData().price << endl;

            tempNode = tempNode->GetNext();

        }

        cout << "--------------------------------------------------------------------------------\n\n";

        cout << "Date: " << temp.date << endl;

        cout << "\t\t\t Amount:\tRs. " << temp.billAmountWithoutDis << endl;

        cout << "\t\t\t Discount:\tRs. ";

        if (temp.loyaltyStatus == 1)

            cout << temp.billAmountWithoutDis \* 0.1 << endl;

        else

            cout << "0" << endl;

        cout << "\t\t\t Total Amount:\tRs. " << temp.totalBillAmount << endl;

        cout << "\n\t\t    Thanks For Shopping Here...!!" << endl

             << endl;

    }

    void printReceiptByName()

    {

        string name;

        cout << "\t\t==================================================\n";

        cout << "\t\t\tViewing Customers Receipts by Name \n";

        cout << "\t\t==================================================\n\n";

        cout << "\n\nEnter Name of Customer : ";

        cin >> name;

        bool j = 0;

        ReceiptNode \*receiptNode = receipts.GetHead();

        while (receiptNode != NULL)

        {

            Receipt temp = receiptNode->GetData();

            if (temp.customerName == name)

            {

                j = 1;

                cout << "\n\n\t\t\t\t  K.K Mart" << endl;

                cout << "\t\t==================================================\n\n";

                cout << "\t\t\t\t" << temp.id << endl;

                cout << "\t\tName: " << temp.customerName << "    \t\tPhone: " << temp.customerPhoneNumber << endl;

                cout << "\t\t==================================================\n\n";

                cout << "Sr.#    |    Product Name    |    Price    |    Quantity    |    Total Price\n\n";

                cout << "--------------------------------------------------------------------------------\n";

                BuyingProductNode \*tempNode = temp.buyingProducts.GetHead();

                int i = 0;

                while (tempNode != NULL)

                {

                    cout << " " << ++i << "\t\t" << tempNode->GetData().name << "\t\t   " << tempNode->GetData().price << "\t\t   "

                         << tempNode->GetData().quantity << "\t\t    " << tempNode->GetData().quantity \* tempNode->GetData().price << endl;

                    tempNode = tempNode->GetNext();

                }

                cout << "Date: " << temp.date << endl;

                cout << "\t\t\t Amount:\tRs. " << temp.billAmountWithoutDis << endl;

                cout << "\t\t\t Discount:\tRs. ";

                if (temp.loyaltyStatus == 1)

                    cout << temp.billAmountWithoutDis \* 0.1 << endl;

                else

                    cout << "0" << endl;

                cout << "\t\t\t Total Amount:\tRs. " << temp.totalBillAmount << endl;

                cout << "\n\t\t    Thanks For Shopping Here...!!" << endl

                     << endl;

            }

            receiptNode = receiptNode->GetNext();

        }

        cout << "-------------------------------------------------------------------\n";

        if (j == 0)

        {

            system("Color 7C");

            cout << "Receipt not found!" << endl;

            return;

        }

    }

    ///////////////////// Manager /////////////////////////////

    void managerAuth()

    {

        system("Color 71");

        char name[30];

        char pass[8];

        char ch;

        system("cls");

        cout << "\t\t==================================================\n";

        cout << "\t\t\t\tManager Log In \n";

        cout << "\t\t==================================================\n\n";

        cout << "\t\t---------------------------------------------------\n";

        cout << "\t\t\tEnter Your Name: ";

        cin >> name;

        cout << "\t\t\tEnter the password: ";

        for (int z = 0; z < 7; z++)

        {

            pass[z] = getch();

            system("cls");

            cout << "\t\t==================================================\n";

            cout << "\t\t\t\tManager Log In \n";

            cout << "\t\t==================================================\n\n";

            cout << "\t\t---------------------------------------------------\n";

            cout << "\t\t\tEnter User Name: " << name << endl;

            cout << "\t\t\tEnter the password: ";

            for (int i = 1; i <= (z + 1); i++)

            {

                cout << "\*";

            }

        }

        if (strcmp(pass, "manager") == 0 && strcmp(name, "manager@kkmart.com") == 0)

        {

            cout << "\n\n\nCongrats!!Access Granted!!\n\n";

            getch();

            displayManagerPortal();

        }

        else

        {

            system("Color 7C");

            cout << "\n\n\nYou Entered the Wrong Password!!!\n\n";

            cout << "Plz Enter y/Y to renter the Password or Press any key to exit" << endl;

            cin >> ch;

            if (ch == 'y' || ch == 'Y')

                managerAuth();

        }

    }

    void displayManagerPortal()

    {

        int choice;

        while (1)

        {

            system("Color 71");

            cls();

            cout << "\t\t==================================================\n";

            cout << "\t\t\t\tManager Portal\n";

            cout << "\t\t==================================================\n\n";

            cout << "\t\t---------------------------------------------------\n";

            cout << "\t\t||   1. Manage Stock  \t\t\t\t ||\n";

            cout << "\t\t||   2. View customer details and purchase\t ||\n\t\t||      history\t\t\t\t\t ||\n";

            cout << "\t\t||   0. Back\t\t\t\t\t ||\n";

            cout << "\t\t---------------------------------------------------\n\n";

            cout << "Enter Your Choice: ";

            cin >> choice;

            switch (choice)

            {

            case 1:

                displayManageStock();

                break;

            case 2:

                displayCustomer();

                system("pause");

                break;

            case 0:

                return;

            default:

                system("Color 7C");

                cout << "Invalid Choice!!!\nRetry...\n";

                system("pause");

            }

        }

    }

    void displayManageStock()

    {

        int choice;

        while (1)

        {

            cls();

            system("Color 71");

            cout << "\t\t==================================================\n";

            cout << "\t\t\t\tManage Stock\n";

            cout << "\t\t==================================================\n\n";

            cout << "\t\t---------------------------------------------------\n";

            cout << "\t\t||   1. Add new product to the inventory\t ||\n";

            cout << "\t\t||   2. Remove a product from the inventory\t ||\n";

            cout << "\t\t||   3. Update product quantity or price\t ||\n";

            cout << "\t\t||   4. View product details and inventory\t ||\n";

            cout << "\t\t||   0. Back\t\t\t\t\t ||\n";

            cout << "\t\t---------------------------------------------------\n\n";

            cout << "Enter Your Choice: ";

            cin >> choice;

            switch (choice)

            {

            case 1:

                addProduct();

                break;

            case 2:

                removeProduct();

                system("pause");

                break;

            case 3:

                updateProduct();

                system("pause");

                break;

            case 4:

                displayProducts();

                system("pause");

                break;

            case 0:

                return;

            default:

                system("Color 7C");

                cout << "Invalid Choice!!!\nRetry...\n";

                system("pause");

            }

        }

    }

    void displayProducts()

    {

        int choice;

        while (1)

        {

            cls();

            system("Color 71");

            cout << "\t\t==================================================\n";

            cout << "\t\t\t\tDisplay Products\n";

            cout << "\t\t==================================================\n\n";

            cout << "\t\t---------------------------------------------------\n";

            cout << "\t\t||   1. View All Products\t\t\t ||\n";

            cout << "\t\t||   2. View Product By Id\t\t\t ||\n";

            cout << "\t\t||   3. View Product By Name\t\t\t ||\n";

            cout << "\t\t||   0. Back\t\t\t\t\t ||\n";

            cout << "\t\t---------------------------------------------------\n\n";

            cout << "Enter Your Choice: ";

            cin >> choice;

            switch (choice)

            {

            case 1:

                cls();

                printAllProducts();

                break;

            case 2:

                cls();

                printProductById();

                system("pause");

                break;

            case 3:

                cls();

                printProductByName();

                system("pause");

                break;

            case 0:

                return;

            default:

                system("Color 7C");

                cout << "Invalid Choice!!!\nRetry...\n";

                system("pause");

            }

        }

    }

    void displayCustomer()

    {

        int choice;

        while (1)

        {

            cls();

            system("Color 71");

            cout << "\t\t==================================================\n";

            cout << "\t\tDisplaying Customer detail and Receipts Detail\n";

            cout << "\t\t==================================================\n\n";

            cout << "\t\t---------------------------------------------------\n";

            cout << "\t\t||   1. View All Receipts of Customers\t\t ||\n";

            cout << "\t\t||   2. View Customer Record By name\t\t ||\n";

            cout << "\t\t||   3. View Receipt By Id\t\t\t ||\n";

            cout << "\t\t||   0. Back\t\t\t\t\t ||\n";

            cout << "\t\t---------------------------------------------------\n\n";

            cout << "Enter Your Choice: ";

            cin >> choice;

            switch (choice)

            {

            case 1:

                cls();

                printAllReceipts();

                system("pause");

                break;

            case 2:

                cls();

                printReceiptByName();

                system("pause");

                break;

            case 3:

                cls();

                printReceiptById();

                system("pause");

                break;

            case 0:

                return;

            default:

                system("Color 7C");

                cout << "Invalid Choice!!!\nRetry...\n";

                system("pause");

            }

        }

    }

    ///////////////////// Cashier /////////////////////////////

    void cashierAuth()

    {

        system("Color 71");

        char name[30];

        char pass[8];

        char ch;

        system("cls");

        cout << "\t\t==================================================\n";

        cout << "\t\t\t\tCashier Info \n";

        cout << "\t\t==================================================\n\n";

        cout << "\t\t--------------------------------------------------\n";

        cout << "\t\t\tEnter User Name: ";

        cin >> name;

        cout << "\t\t\tEnter the Password: ";

        for (int z = 0; z < 7; z++)

        {

            pass[z] = getch();

            system("cls");

            cout << "\t\t==================================================\n";

            cout << "\t\t\t\tCashier Info \n";

            cout << "\t\t==================================================\n\n";

            cout << "\t\t--------------------------------------------------\n";

            cout << "\t\t\tEnter User Name: " << name << endl;

            cout << "\t\t\tEnter the Password: ";

            for (int i = 1; i <= (z + 1); i++)

            {

                cout << "\*";

            }

        }

        if (strcmp(pass, "cashier") == 0 && strcmp(name, "cashier@kkmart.com") == 0)

        {

            cout << "\n\n\nCongrats!!Access Granted!!\n\n";

            getch();

            displayCashierPortal();

        }

        else

        {

            system("Color 7C");

            cout << "\n\n\nYou Entered the Wrong Password!!!\n\nPlz Enter y/Y to renter the Password or Press any key to exit" << endl;

            cin >> ch;

            if (ch == 'y' || ch == 'Y')

                cashierAuth();

        }

    }

    void displayCashierPortal()

    {

        int choice;

        while (1)

        {

            system("Color 71");

            cls();

            cout << "\t\t==================================================\n";

            cout << "\t\t\t\tCashier Portal\n";

            cout << "\t\t==================================================\n\n";

            cout << "\t\t---------------------------------------------------\n";

            cout << "\t\t||   1. Take Customer\'s Order\t\t\t ||\n";

            cout << "\t\t||   2. Print all receipts    \t\t\t ||\n";

            cout << "\t\t||   3. Print a receipt by Id \t\t\t ||\n";

            cout << "\t\t||   0. Back\t\t\t\t\t ||\n";

            cout << "\t\t---------------------------------------------------\n\n";

            cout << "Enter Your Choice: ";

            cin >> choice;

            switch (choice)

            {

            case 1:

                cls();

                addReceipt();

                pause();

                break;

            case 2:

                cls();

                printAllReceipts();

                system("pause");

                break;

            case 3:

                cls();

                printReceiptById();

                system("pause");

                break;

            case 0:

                return;

                break;

            default:

                system("Color 7C");

                cout << "Invalid Choice!!!\nRetry...\n";

                system("pause");

            }

        }

    }

};

int main()

{

    int choice;

    Management manage;

    cls();

    system("Color 71");

    cout << "================================================================================\n";

    cout << "                      \*                                        \*\n";

    cout << "            \*        \* \*                                      \* \*        \*\n";

    cout << "           \* \*        \*             WELCOME   To               \*        \* \*\n";

    cout << "          \*   \*            \*         K.K   Mart          \*             \*   \*\n";

    cout << "           \* \*            \* \*                           \* \*             \* \*\n";

    cout << "            \*              \*                             \*               \*\n\n";

    cout << "================================================================================\n";

    cout << "\t\t\t\t\t\tRepresented By:\n\t\t\t\t\t\tKanza Kashaf 22-NTU-CS-1350\n\t\t\t\t\t\tHassaan Raza 22-NTU-CS-1362";

    getch();

    while (1)

    {

        cls();

        system("Color 71");

        cout << "\t\t=============================================\n";

        cout << "\t\t\t\tMain Menu\n";

        cout << "\t\t=============================================\n\n";

        cout << "\t\t----------------------------------------------\n";

        cout << "\t\t||   1. Manager Login\t\t\t    ||\n";

        cout << "\t\t||   2. Cashier Login\t\t\t    ||\n";

        cout << "\t\t||   3. Exit\t\t\t\t    ||\n";

        cout << "\t\t----------------------------------------------\n\n";

        cout << "Enter Your Choice: ";

        cin >> choice;

        switch (choice)

        {

        case 1:

            manage.managerAuth();

            break;

        case 2:

            manage.cashierAuth();

            system("pause");

            break;

        case 3:

            return 0;

        default:

            system("Color 7C");

            cout << "Invalid Choice!!!\nRetry...\n";

            system("pause");

        }

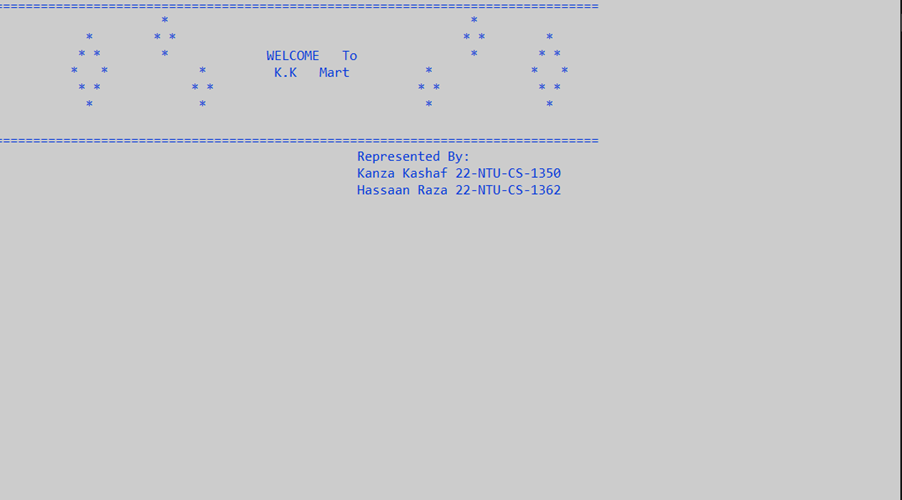
    }

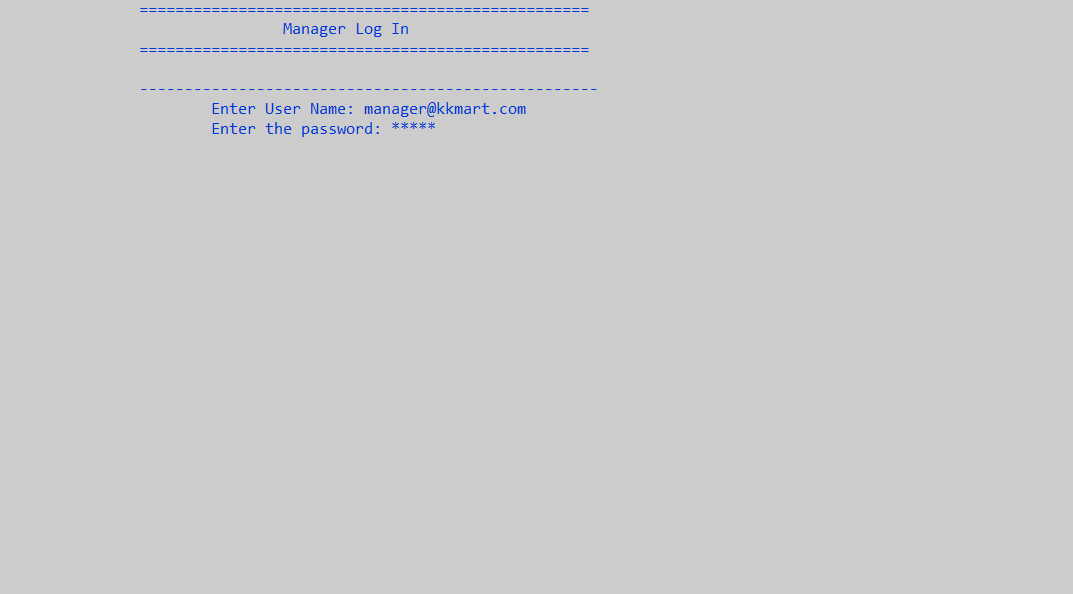
    return 0;

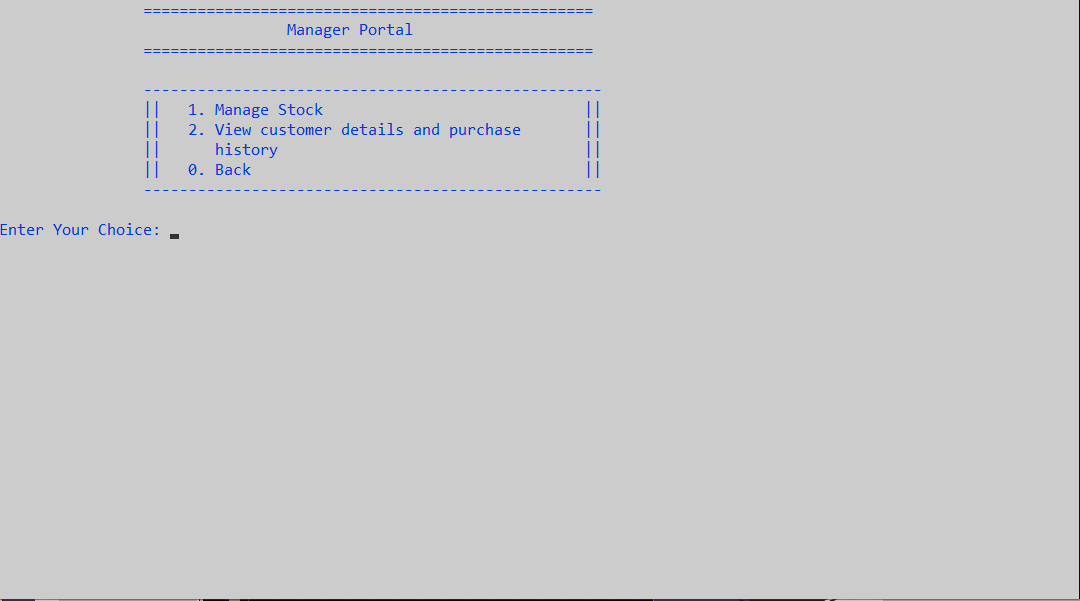
}

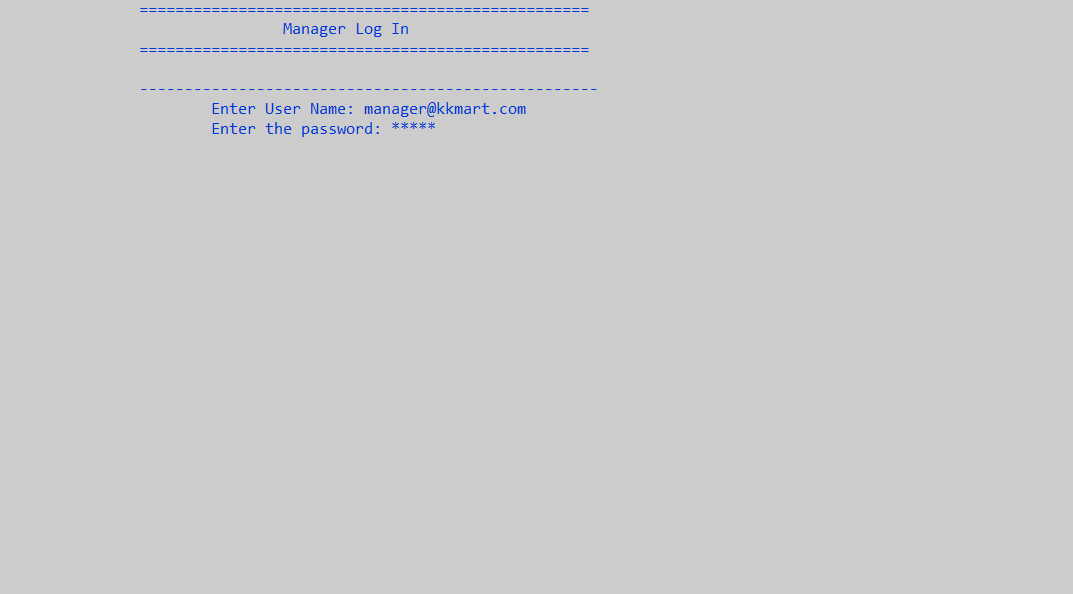
# Output:

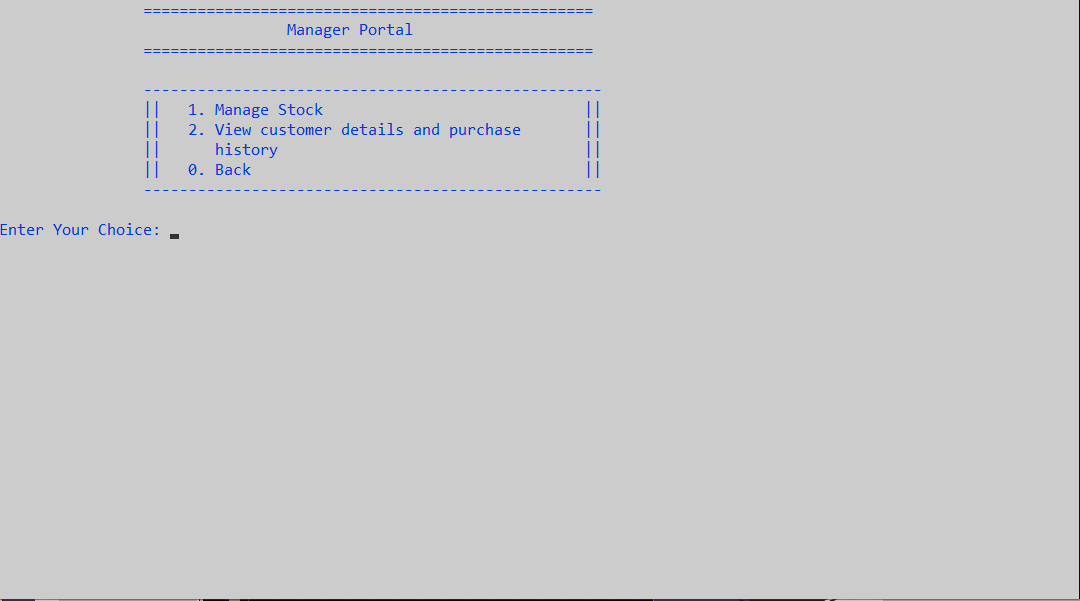
Some Main output

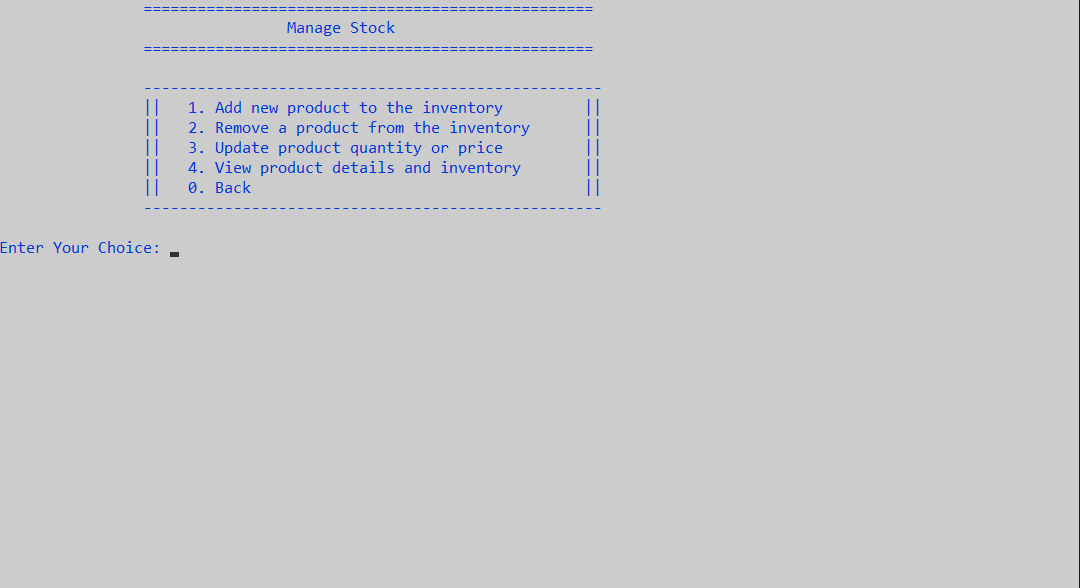


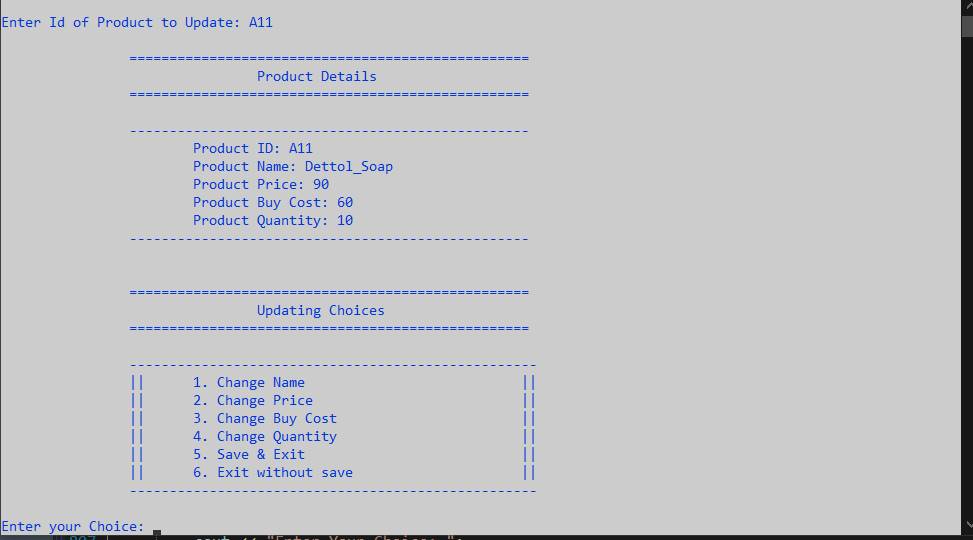


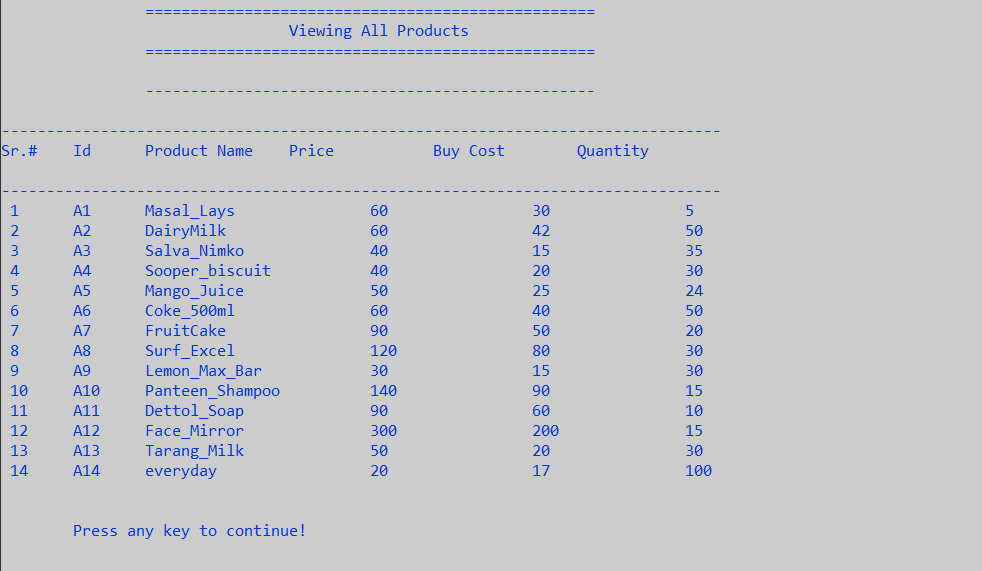


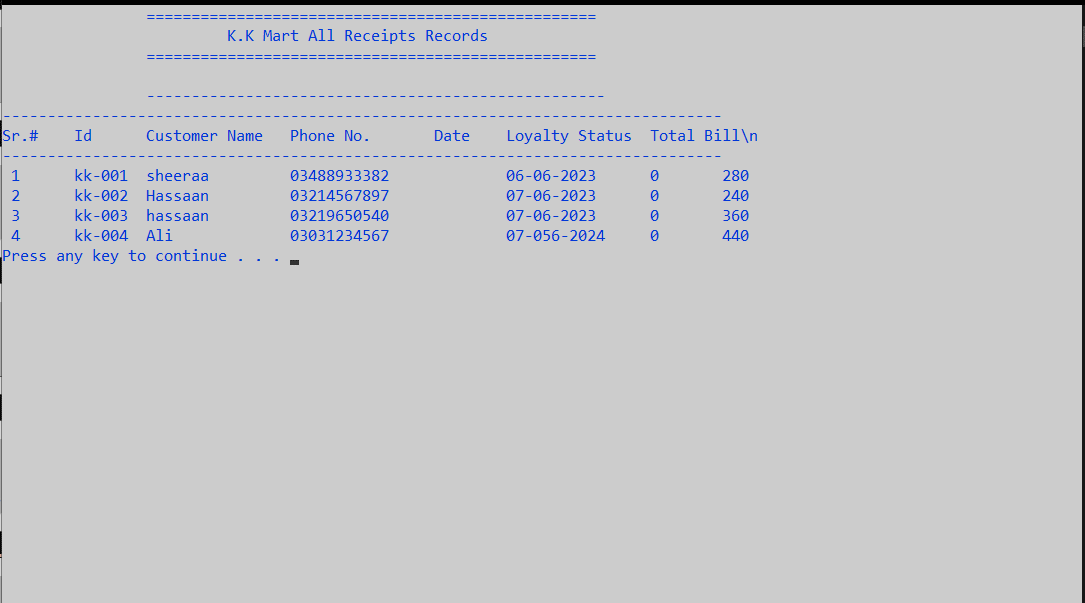












A screenshot of a receipt

Description automatically generated