OOP LAB PROJECT

Team Members :

Muhammad Burhan Niaz : (24-cs-1)

Hassan Safdar : (24-cs-71)

Mubashir Rashid : (24-cs-42)

Submited to :

Sir Faheem

Project name :

ONLINE E-COMMERCE STORE

Preface

This comprehensive report provides an in-depth exploration into the meticulous development process and the intricate functionality imbued within a sophisticated rendition of the classic Online e-commerce store. Through a judicious application of advanced object-oriented programming (OOP) concepts within the realm of the C++ programming language, this project has transcended the boundaries of traditional development, ushering in a new era of immersive and interactive gaming experiences. By meticulously leveraging the foundational principles of OOP, including encapsulation, inheritance, and polymorphism, the project embodies the epitome of software engineering excellence, transforming the rudimentary of online store into a finely crafted masterpiece of code. This endeavor epitomizes the fusion of creativity, technical prowess, and innovation, exemplifying the profound impact that OOP methodologies can have in sculpting dynamic and engaging software applications.

Contents

Introduction

Online E-commerec is the c++ based project built in oop concepts which demonstrates the usability of object oriented programming in building stores .

There is auser console opening with free will to choose as costumer or as admin the as an admin select the options 1. View products, add products, remove products 2.login as costumer , viw products, add to cart , check cart, register as new costumer and exit.

About the Project

The project utilizes C++ programming language and OOP concepts to create a Online e-commerce store. The is implemented within a class named ` Online e-commerce store `, which encapsulates the logic and functionality. The provides a command-line interface for players to make moves and displays the current state of the board.

Working of the Code

The `displayBoard` function prints the current state of the Menu board.The user has choice to login as admin or costumer.The admin can add product , remove product and view product. The costumer can view products, add to cart , check cart.

Code Explanation

- The `displayBoard` function prints the current state of the Menu board.
- The user has choice to login as admin or costumer
- The admin can add product , remove product and view product.
- The costumer can view products, add to cart , check cart.

Scope and Limitation

- **Scope:** This project demonstrates the application of OOP concepts in creating a simple yet engaging .
- **Limitation:** The is limited to console and does not include advanced features such as GUI or customizable board sizes.

Conclusion

In conclusion, the Online e-commerce store implemented in this project showcases the practical implementation of OOP concepts in C++. It provides an interactive platform for players to enjoy the classic while demonstrating fundamental programming principles such as encapsulation, inheritance, and polymorphism. Further enhancements could be made to extend the functionality and user experience.

Source Code

//Online e-commerce store using OOP concepts…

```
#include<iostream>
#include<string>
#include<fstream>
#include<iomanip>
using namespace std;
const int MAX_PRODUCTS = 100;
const int MAX_CART_ITEMS = 100;
const int MAX_ORDERS = 100;
class Product;
class Order;
class User {
protected:
    string name;
    string email;
    string address;
public:
    void set_user_attr(string n, string e, string a) {
        name = n;
        email = e;
        address = a;
    }
    string get_user_name() {
        return name;
    }
    string get_user_email() {
        return email;
    }
    string get_user_address() {
        return address;
    }
    void display_details() {
        cout << "Name is: " << name << "\n";
        cout << "E-mail is: " << email << "\n";
        cout << "Address is: " << address << "\n";
    }
};
class Product {
private:
    int id;
    string name;
    string category;
    float price;
    int stock;
public:
    Product() {
        id = 0;
        name = "";
        category = "";
        price = 0.0;
        stock = 0;
    }
    Product(int i, string n, string c, float p, int s) {
        id = i;
        name = n;
        category = c;
        price = p;
        stock = s;
    }
    int get_id() { return id; }
    string get_name() { return name; }
    float get_price() { return price; }
    int get_stock() { return stock; }
    void set_stock(int s) { stock = s; }
    void display_product() {
        cout << "Product's id is: " << id << "\n";
        cout << "Product's name is: " << name << "\n";
        cout << "Product's category is: " << category << "\n";
        cout << "Product's price is: $" << fixed << setprecision(2) << price << "\n";
        cout << "Product left in stock: " << stock << "\n";
    }
};
Product g_products[MAX_PRODUCTS];
int g_productCount = 0;
void saveProductsToFile() {
    ofstream outFile("products.txt");
    if (outFile.is_open()) {
        for (int i = 0; i < g_productCount; i++) {
            outFile << g_products[i].get_id() << " "
                << g_products[i].get_name() << " "
                << g_products[i].get_price() << " "
                << g_products[i].get_stock() << endl;
        }
        outFile.close();
    }
    else {
        cout << "Unable to save products to file.\n";
    }
}
void loadProductsFromFile() {
    ifstream inFile("products.txt");
    g_productCount = 0;
    if (inFile.is_open()) {
        int id, stock;
        string name, category;
        float price;
        while (g_productCount < MAX_PRODUCTS &&
            inFile >> id >> name >> price >> stock) {
            g_products[g_productCount] = Product(id, name, category, price, stock);
            g_productCount++;
        }
        inFile.close();
        cout << "Products loaded from file.\n";
    }
    else {
        cout << "No product file found. Starting with empty inventory.\n";
    }
}
class Admin : public User {
private:
    string password;
public:
    Admin() {
        password = "admin123";
    }
    bool adminLogin(string pass) {
        return (pass == password);
    }
    void addproduct() {
        if (g_productCount >= MAX_PRODUCTS) {
            cout << "Product list is full. Cannot add more products.\n";
            return;
        }
        int id;
        string name, category;
        float price;
        int stock;
        cout << "Enter Product ID: ";
        cin >> id;
        for (int i = 0; i < g_productCount; i++) {
            if (g_products[i].get_id() == id) {
                cout << "Product with this ID already exists.\n";
                return;
            }
        }
        cin.ignore();
        cout << "Enter Product Name: ";
        getline(cin, name);
        cout << "Enter Product Category: ";
        getline(cin, category);
        cout << "Enter Product Price: $";
        cin >> price;
        cout << "Enter Product Stock: ";
        cin >> stock;
        g_products[g_productCount] = Product(id, name, category, price, stock);
        g_productCount++;
        cout << "Product added successfully!\n";
        saveProductsToFile();
    }
    void removeproduct() {
        int id;
        bool found = false;
        cout << "Enter Product ID to remove: ";
        cin >> id;
        for (int i = 0; i < g_productCount; i++) {
            if (g_products[i].get_id() == id) {
                found = true;
                for (int j = i; j < g_productCount - 1; j++) {
                    g_products[j] = g_products[j + 1];
                }
                g_productCount--;
```

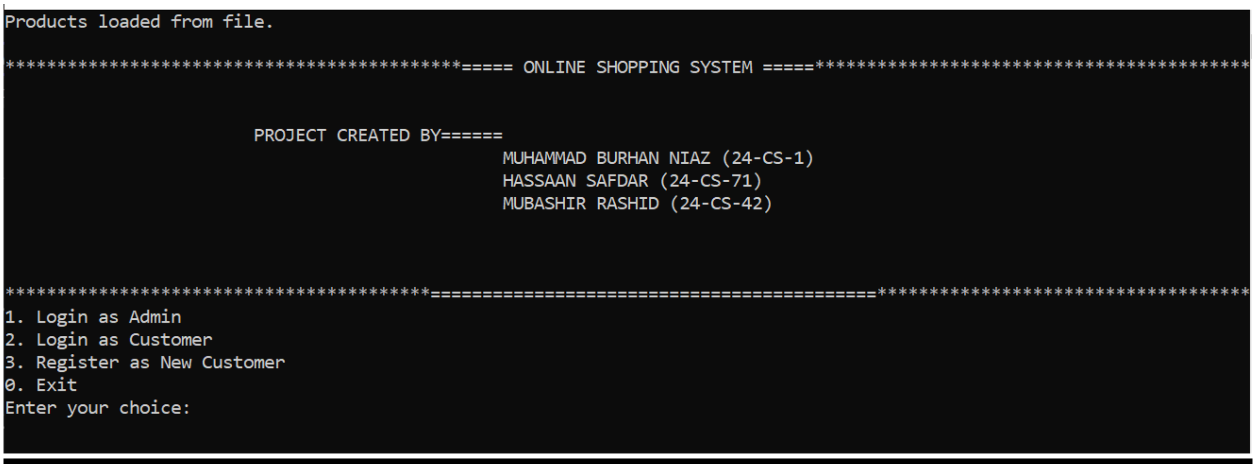
```
                cout << "Product removed successfully!\n";
                break;
            }
        }
        if (!found) {
            cout << "Product with ID " << id << " not found.\n";
        }
        saveProductsToFile();
    }
    void view_all_products() {
        if (g_productCount == 0) {
            cout << "No products available.\n";
            return;
        }
        cout << "\n===== PRODUCT INVENTORY =====\n";
        for (int i = 0; i < g_productCount; i++) {
            cout << "\nProduct #" << (i + 1) << ":\n";
            g_products[i].display_product();
            cout << "--------------------------------\n";
        }
    }
};
int g_next_order_id = 1;
void load_next_order_id() {
    ifstream inFile("orders.txt");
    string line;
    int max_id = 0;
    int id;
    if (inFile.is_open()) {
        while (inFile >> id) {
            inFile.ignore(1000, '\n');

            if (id > max_id) {
                max_id = id;
            }
        }
        inFile.close();
    }
    g_next_order_id = max_id + 1;
}
```

```
class Order {
private:
    int order_id;
    string customer_name;
    float total_price;
    int products_purchased[MAX_CART_ITEMS];
    int product_quantities[MAX_CART_ITEMS];
    int product_count;
public:
    Order() {
        order_id = g_next_order_id++;
        customer_name = "";
        total_price = 0.0;
        product_count = 0;
        for (int i = 0; i < MAX_CART_ITEMS; i++) {
            products_purchased[i] = -1;
            product_quantities[i] = 0;
        }
    }
    void set_customer_name(string name) {
        customer_name = name;
    }
    void set_total_price(float total) {
        total_price = total;
    }
    void add_product(int product_id, int quantity) {
        if (product_count < MAX_CART_ITEMS) {
            products_purchased[product_count] = product_id;
            product_quantities[product_count] = quantity;
            product_count++;
        }
    }
    void calculate_order_total() {
        total_price = 0.0;
        for (int i = 0; i < product_count; i++) {
            int prod_id = products_purchased[i];
            int qty = product_quantities[i];
            for (int j = 0; j < g_productCount; j++) {
                if (g_products[j].get_id() == prod_id) {
                    total_price += qty * g_products[j].get_price();
                    break;
                }
            }
        }
    }
    void display_order_details() {
        cout << "\n======= ORDER RECEIPT =======\n";
        cout << "Order ID: " << order_id << "\n";
        cout << "Customer: " << customer_name << "\n";
        cout << "----------------------------\n";
        for (int i = 0; i < product_count; i++) {
            int prod_id = products_purchased[i];
            int qty = product_quantities[i];
            for (int j = 0; j < g_productCount; j++) {
                if (g_products[j].get_id() == prod_id) {
                    cout << g_products[j].get_name() << " x " << qty << "\n";
                    cout << "  $" << fixed << setprecision(2) << g_products[j].get_price();
                    cout << " each = $" << fixed << setprecision(2) << (qty *
g_products[j].get_price()) << "\n";
                    break;
                }
            }
        }
        cout << "----------------------------\n";
        cout << "TOTAL: $" << fixed << setprecision(2) << total_price << "\n";
        cout << "==============================\n";
    }
    void save_to_file() {
        ofstream outFile("orders.txt", ios::app);
        if (outFile.is_open()) {
            outFile << order_id << " " << customer_name << " " << total_price;
            for (int i = 0; i < product_count; i++) {
                outFile << " " << products_purchased[i] << " " << product_quantities[i];
            }
            outFile << endl;
            outFile.close();
        }
        else {
            cout << "Unable to save order to file.\n";
        }
    }
};
class Customer : public User {
private:
    int shopping_cart[MAX_CART_ITEMS];
    int cart_quantity[MAX_CART_ITEMS];
    int cart_count;
    string password;
public:
    Customer() {
        cart_count = 0;
        password = "";
        for (int i = 0; i < MAX_CART_ITEMS; i++) {
            shopping_cart[i] = -1;
            cart_quantity[i] = 0;
        }
    }
    void set_password(string pass) {
        password = pass;
    }
    bool customerLogin(string pass) {
        return (pass == password);
    }
    void view_products() {
        if (g_productCount == 0) {
            cout << "No products available.\n";
            return;
        }
        cout << "\n===== AVAILABLE PRODUCTS =====\n";
        for (int i = 0; i < g_productCount; i++) {
            cout << "\nProduct #" << (i + 1) << ":\n";
            cout << "ID: " << g_products[i].get_id() << "\n";
            cout << "Name: " << g_products[i].get_name() << "\n";
            cout << "Price: $" << fixed << setprecision(2) << g_products[i].get_price() << "\n";
            cout << "Stock: " << g_products[i].get_stock() << "\n";
            cout << "----------------------------\n";
        }
    }
    void add_to_cart() {
        int product_id, quantity;
        bool found = false;
        view_products();
        cout << "Enter Product ID to add to cart: ";
        cin >> product_id;
        cout << "Enter quantity: ";
        cin >> quantity;
        for (int i = 0; i < g_productCount; i++) {
            if (g_products[i].get_id() == product_id) {
                found = true;
                if (g_products[i].get_stock() < quantity) {
                    cout << "Not enough stock. Available: " << g_products[i].get_stock()
<< "\n";
                    return;
                }
                for (int j = 0; j < cart_count; j++) {
                    if (shopping_cart[j] == product_id) {
                        cart_quantity[j] += quantity;
                        cout << "Updated quantity in cart.\n";
                        return;
                    }
                }
                if (cart_count < MAX_CART_ITEMS) {
                    shopping_cart[cart_count] = product_id;
                    cart_quantity[cart_count] = quantity;
                    cart_count++;
                    cout << "Product added to cart.\n";
                }
                else {
                    cout << "Cart is full.\n";
                }
                break;
            }
        }
        if (!found) {
            cout << "Product with ID " << product_id << " not found.\n";
        }
    }
    void view_cart() {
        if (cart_count == 0) {
            cout << "Your cart is empty.\n";
            return;
        }
        float total = 0.0;
        cout << "\n===== YOUR SHOPPING CART =====\n";
        for (int i = 0; i < cart_count; i++) {
            int prod_id = shopping_cart[i];
            int qty = cart_quantity[i];
            for (int j = 0; j < g_productCount; j++) {
                if (g_products[j].get_id() == prod_id) {
                    cout << "Item #" << (i + 1) << ": " << g_products[j].get_name() <<
"\n";
                    cout << "Quantity: " << qty << "\n";
                    cout << "Price per unit: $" << fixed << setprecision(2) <<
g_products[j].get_price() << "\n";
                    float item_total = qty * g_products[j].get_price();
                    cout << "Item total: $" << fixed << setprecision(2) << item_total <<
"\n";
                    cout << "---------------------------\n";
                    total += item_total;
                    break;
                }
            }
        }
        cout << "CART TOTAL: $" << fixed << setprecision(2) << total << "\n";
    }
    void check_out() {
        if (cart_count == 0) {
            cout << "Your cart is empty. Nothing to checkout.\n";
            return;
        }
        view_cart();
        char confirm;
        cout << "Confirm order (Y/N): ";
        cin >> confirm;
        if (toupper(confirm) != 'Y') {
            cout << "Checkout cancelled.\n";
            return;
        }
        float total = 0.0;
        Order new_order;
        new_order.set_customer_name(name);
        for (int i = 0; i < cart_count; i++) {
            int prod_id = shopping_cart[i];
            int qty = cart_quantity[i];
            for (int j = 0; j < g_productCount; j++) {
                if (g_products[j].get_id() == prod_id) {
                    g_products[j].set_stock(g_products[j].get_stock() - qty);
                    total += qty * g_products[j].get_price();
                    new_order.add_product(prod_id, qty);
                    break;
                }
            }
        }
        new_order.set_total_price(total);
        new_order.display_order_details();
        saveProductsToFile();
        new_order.save_to_file();
        cart_count = 0;
        for (int i = 0; i < MAX_CART_ITEMS; i++) {
            shopping_cart[i] = -1;
            cart_quantity[i] = 0;
        }
        cout << "Thank you for your purchase!\n";
    }
};
void admin_menu();
void customer_menu();
void register_customer();
Admin current_admin;
Customer current_customer;
bool is_logged_in = false;
bool is_admin = false;
int main() {
    loadProductsFromFile();
    load_next_order_id();
    int choice;
    do {
```

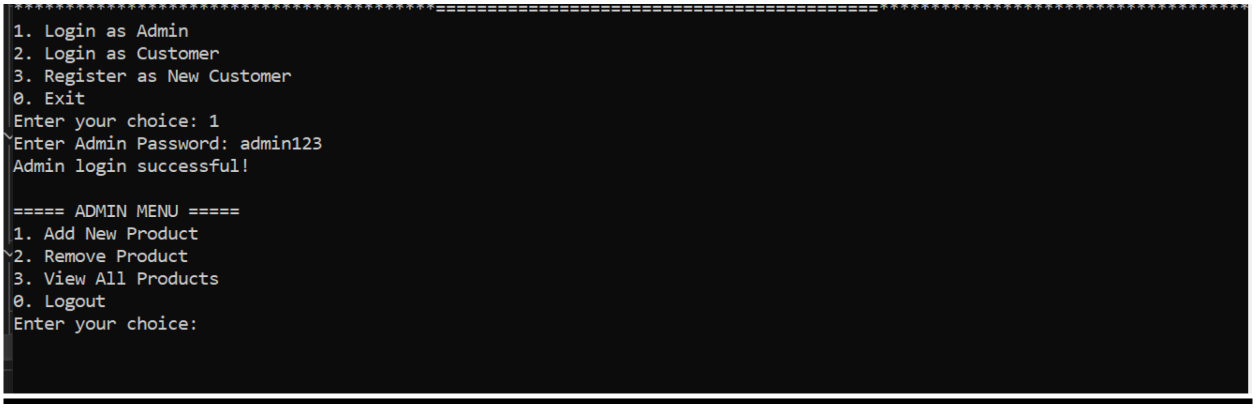
```
    cout << "\n********************************************=====
ONLINE SHOPPING SYSTEM
=====***************************\***************\n\n\n";
    cout<<"\t\t\t\PROJECT CREATED BY====== \n";
    cout << "\t\t\t\t\t\tMUHAMMAD BURHAN NIAZ (24-CS-1)\n";
    cout << "\t\t\t\t\t\tHASSAAN SAFDAR (24-CS-71)\n";
    cout << "\t\t\t\t\t\tMUBASHIR RASHID (24-CS-42)\n";
    cout <<
"\n\n\n*****************************************==================
=========================************************************\n"
;
    cout << "1. Login as Admin\n";
    cout << "2. Login as Customer\n";
    cout << "3. Register as New Customer\n";
    cout << "0. Exit\n";
    cout << "Enter your choice: ";
    cin >> choice;
    switch (choice) {
    case 1: {
      string password;
      cout << "Enter Admin Password: ";
      cin >> password;
      if (current_admin.adminLogin(password)) {
        is_logged_in = true;
        is_admin = true;
        cout << "Admin login successful!\n";
        admin_menu();
      }
      else {
        cout << "Invalid password.\n";
      }
      break;
    }
    case 2: {
      string email, password;
      cout << "Enter Email: ";
      cin >> email;
      cout << "Enter Password: ";
      cin >> password;
      ifstream inFile("customers.txt");
```

```
            string stored_email, stored_pass, name, address;
            bool found = false;
            if (inFile.is_open()) {
                while (inFile >> stored_email >> stored_pass >> name >> address) {
                    if (stored_email == email && stored_pass == password) {
                        found = true;
                        current_customer.set_user_attr(name, email, address);
                        current_customer.set_password(password);
                        is_logged_in = true;
                        is_admin = false;
                        cout << "Login successful!\n";
                        customer_menu();
                        break;
                    }
                }
                inFile.close();
            }
            if (!found) {
                cout << "Invalid email or password.\n";
            }
            break;
        }
        case 3:
            register_customer();
            break;
        case 0:
            cout << "Thank you for using our system!\n";
            break;
        default:
            cout << "Invalid choice. Please try again.\n";
        }
    } while (choice != 0);
    return 0;
}
void admin_menu() {
    int choice;
    do {
        cout << "\n===== ADMIN MENU =====\n";
        cout << "1. Add New Product\n";
        cout << "2. Remove Product\n";
        cout << "3. View All Products\n";
        cout << "0. Logout\n";
        cout << "Enter your choice: ";
        cin >> choice;
        switch (choice) {
        case 1:
            current_admin.addproduct();
            break;
        case 2:
            current_admin.removeproduct();
            break;
        case 3:
            current_admin.view_all_products();
            break;
        case 0:
            is_logged_in = false;
            is_admin = false;
            cout << "Logged out successfully.\n";
            break;
        default:
            cout << "Invalid choice. Please try again.\n";
        }
    } while (choice != 0);
}
void customer_menu() {
    int choice;
    do {
        cout << "\n===== CUSTOMER MENU =====\n";
        cout << "1. View Available Products\n";
        cout << "2. Add Product to Cart\n";
        cout << "3. View Cart\n";
        cout << "4. Checkout\n";
        cout << "0. Logout\n";
        cout << "Enter your choice: ";
        cin >> choice;
        switch (choice) {
        case 1:
            current_customer.view_products();
            break;
        case 2:
            current_customer.add_to_cart();
            break;
        case 3:
            current_customer.view_cart();
            break;
        case 4:
            current_customer.check_out();
            break;
        case 0:
            is_logged_in = false;
            is_admin = false;
            cout << "Logged out successfully.\n";
            break;
        default:
            cout << "Invalid choice. Please try again.\n";
        }

    } while (choice != 0);
}
void register_customer() {
    string name, email, address, password;
    cin.ignore();
    cout << "Enter Your Name: ";
    getline(cin, name);
    cout << "Enter Your Email: ";
    getline(cin, email);
    ifstream checkFile("customers.txt");
    string stored_email;
    bool email_exists = false;
    if (checkFile.is_open()) {
        while (checkFile >> stored_email) {
            checkFile.ignore(1000, '\n');
            if (stored_email == email) {
                email_exists = true;
                break;
            }
        }
        checkFile.close();
    }
    if (email_exists) {
        cout << "This email is already registered.\n";
        return;
    }
    cout << "Enter Your Address: ";
    getline(cin, address);
    cout << "Create Password: ";
    getline(cin, password);
    ofstream outFile("customers.txt", ios::app);
    if (outFile.is_open()) {
        outFile << email << " " << password << " " << name << " " << address <<
endl;
        outFile.close();
        cout << "Registration successful! You can now login.\n";
    }
    else {
        cout << "Unable to register. Please try again later.\n";
    }
}
```

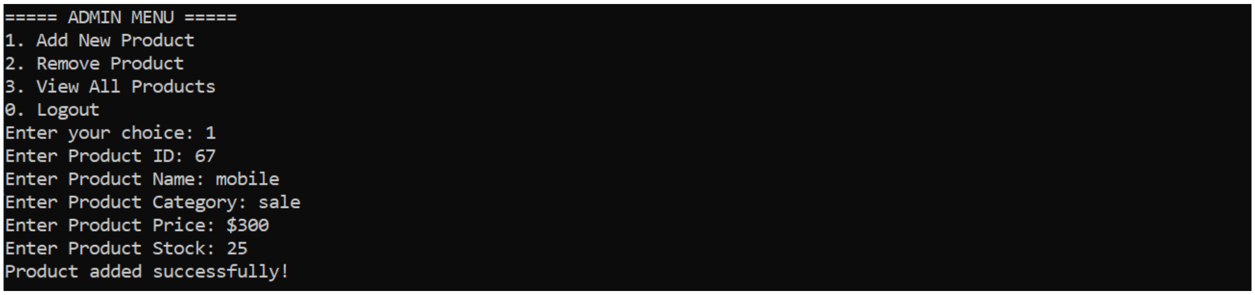
Output



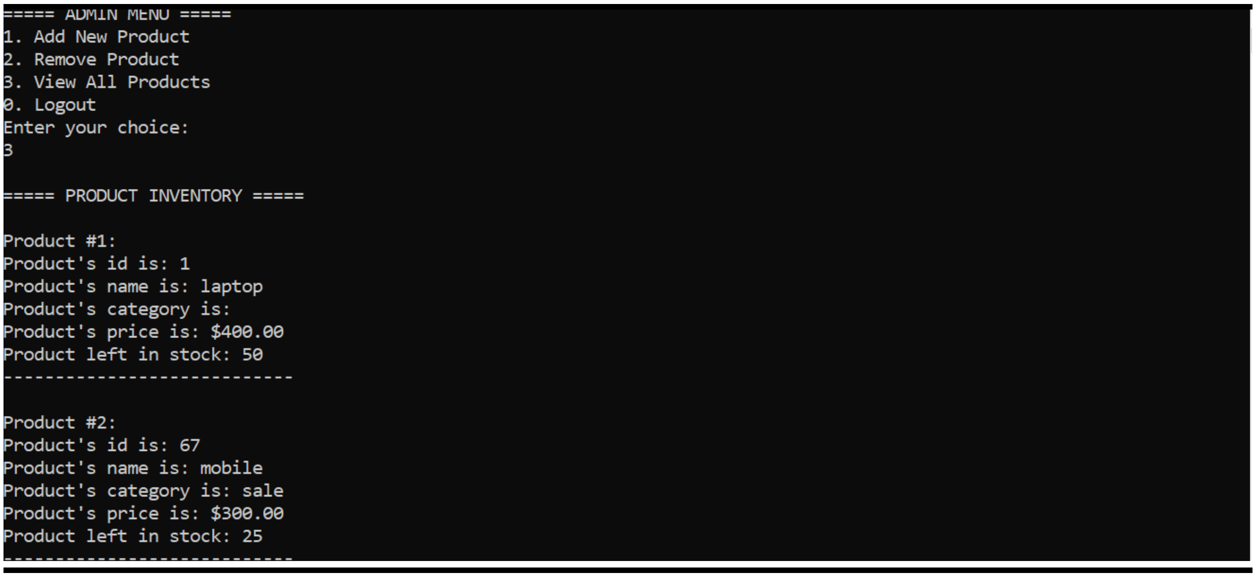
```
Products loaded from file.

********************************************===== ONLINE SHOPPING SYSTEM =====******************************************


                        PROJECT CREATED BY======
                                                MUHAMMAD BURHAN NIAZ (24-CS-1)
                                                HASSAAN SAFDAR (24-CS-71)
                                                MUBASHIR RASHID (24-CS-42)



*****************************************===========================================************************************
1. Login as Admin
2. Login as Customer
3. Register as New Customer
0. Exit
Enter your choice:
```



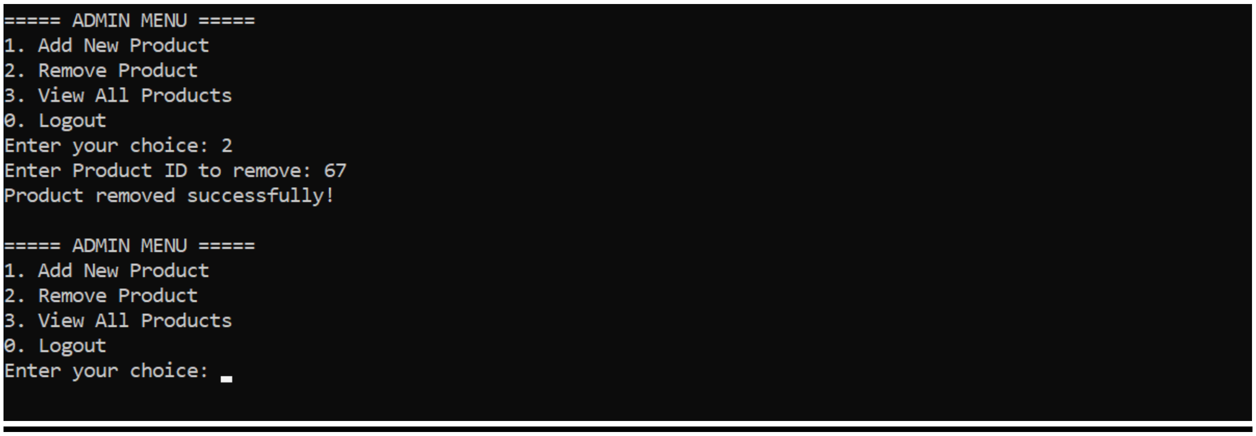
```
**************************************************======================================================**********************************************
1. Login as Admin
2. Login as Customer
3. Register as New Customer
0. Exit
Enter your choice: 1
Enter Admin Password: admin123
Admin login successful!

===== ADMIN MENU =====
1. Add New Product
2. Remove Product
3. View All Products
0. Logout
Enter your choice:
```



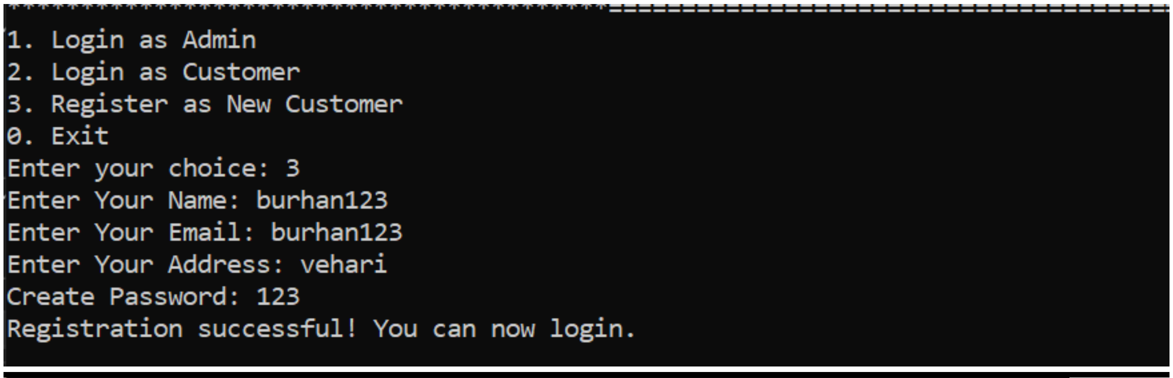
```
===== ADMIN MENU =====
1. Add New Product
2. Remove Product
3. View All Products
0. Logout
Enter your choice: 1
Enter Product ID: 67
Enter Product Name: mobile
Enter Product Category: sale
Enter Product Price: $300
Enter Product Stock: 25
Product added successfully!
```



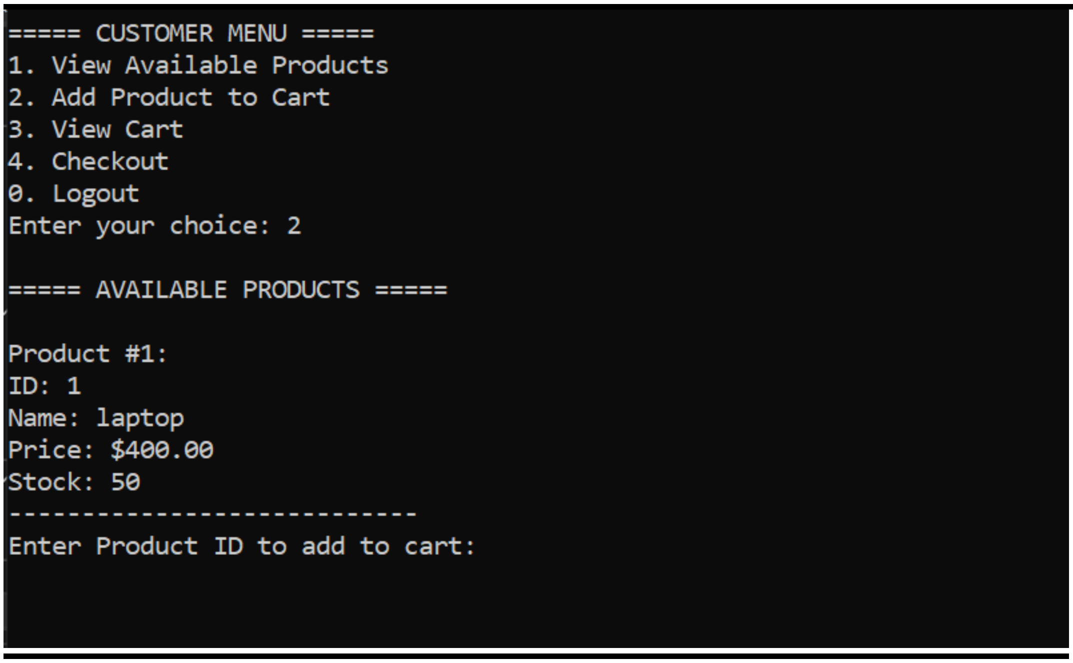
```
===== ADMIN MENU =====
1. Add New Product
2. Remove Product
3. View All Products
0. Logout
Enter your choice:
3

===== PRODUCT INVENTORY =====

Product #1:
Product's id is: 1
Product's name is: laptop
Product's category is:
Product's price is: $400.00
Product left in stock: 50
----------------------------

Product #2:
Product's id is: 67
Product's name is: mobile
Product's category is: sale
Product's price is: $300.00
Product left in stock: 25
----------------------------
```



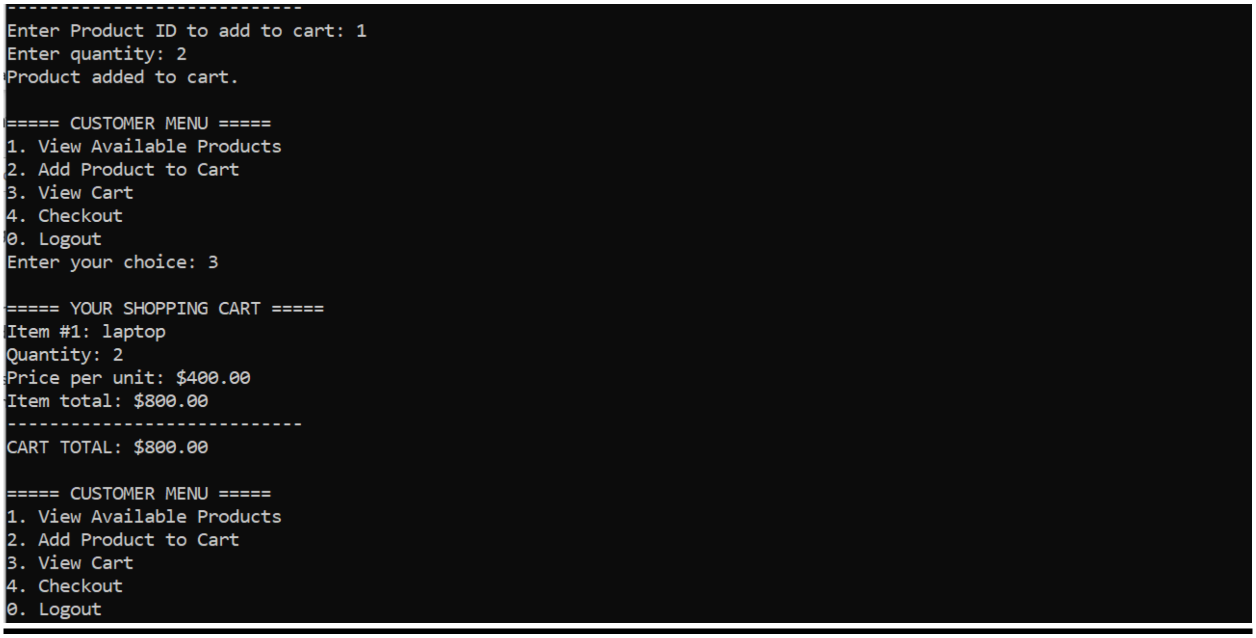
```
===== ADMIN MENU =====
1. Add New Product
2. Remove Product
3. View All Products
0. Logout
Enter your choice: 2
Enter Product ID to remove: 67
Product removed successfully!

===== ADMIN MENU =====
1. Add New Product
2. Remove Product
3. View All Products
0. Logout
Enter your choice: _
```



```
1. Login as Admin
2. Login as Customer
3. Register as New Customer
0. Exit
Enter your choice: 3
Enter Your Name: burhan123
Enter Your Email: burhan123
Enter Your Address: vehari
Create Password: 123
Registration successful! You can now login.
```



```
===== CUSTOMER MENU =====
1. View Available Products
2. Add Product to Cart
3. View Cart
4. Checkout
0. Logout
Enter your choice: 2

===== AVAILABLE PRODUCTS =====

Product #1:
ID: 1
Name: laptop
Price: $400.00
Stock: 50
----------------------------
Enter Product ID to add to cart:
```



```
----------------------------
Enter Product ID to add to cart: 1
Enter quantity: 2
Product added to cart.

===== CUSTOMER MENU =====
1. View Available Products
2. Add Product to Cart
3. View Cart
4. Checkout
0. Logout
Enter your choice: 3

===== YOUR SHOPPING CART =====
Item #1: laptop
Quantity: 2
Price per unit: $400.00
Item total: $800.00
----------------------------
CART TOTAL: $800.00

===== CUSTOMER MENU =====
1. View Available Products
2. Add Product to Cart
3. View Cart
4. Checkout
0. Logout
```