



Ahsanullah University of Science and Technology

Open ended lab report Project

Title: Online Shopping System

Submitted by,

Name : Promit Ghosh

ID : 00724205131168

Course Title : EEE-1110

Course name : Programming Language Laboratory

Department : Electrical and Electronic Engineering

Year : 1st

Semester : 1st

Section : C (C2)

Date of Submission : 20th September, 2025

Submitted To : Mr. Mohammad Minhaz Akram

This is a console-based online shopping system which is developed in C++ and allows its users to register, login and manage an online based shopping cart.

This program does the simulation of a basic shopping environment where users can view products, add products or even remove products.

This program includes two types of users:

1. Customer: can browse products, manage cart and checkout.

2. Admin: can manage products like adding, editing or removing and view the inventory.

By this particular system one can understand the use of object-oriented programming concepts such as classes, structures, objects and encapsulation. It also demonstrates file handling in C++. It also demonstrates dynamic data handling with the help of vectors and mani-

pulating text files for persistent storage.

Objectives:

The objective of this program is to,

1. Implement a basic online shopping system using C++.
2. Provide different role-based access for both the admins and customers.
3. Demonstrate cart management, billing and product inventory.
4. Practice file handling to preserve customer's and admin's user name and password information, cart and product information.

Features of the program:

• Admin features:

1. Add new products with name, price and stocks available.
2. View list of all available products.

3. Edit any product details.
4. Remove product from the list.
5. These data automatically updates in the text file products.txt.

- Customer features:

1. Registers or log in with role as customer.
2. View available products.
3. Add selected products into shopping cart.
4. View items in cart before checkout.
5. Checkout and generate bill file (bill-username.txt)

This ensures that the admins and customers are completely separate.

6. Stock updates automatically after every purchase.
7. Cart data is saved per user in different text files (cart-username.txt)

Libraries used in the program:

1. <iostream> → Displaying menus and messages and taking input from the user.
2. <fstream> → For reading from files (admin.txt, products.txt, customer.txt) and writing to files (bill_username.txt, cart_username.txt)
3. <string> → Used for storing usernames, passwords, product names and roles
4. <vector> → Used for implementing a dynamic storage (dynamic array like), used to store:
 1. vector<Product> → all products
 2. vector<Cart-Item> → items in the customer's cart
 3. vector<UserRecord> → all registered users.

File handling in the program

This program uses text files to store different data permanently.

- admins.txt → This text file stores admin account informations.
- customers.txt → This text file stores customer account informations.
- products.txt → This text file stores all the product information.
- cart_username.txt → This text file stores individual customer cart.
- bill_username.txt → This text file stores individual customer bills.

This ensures admins and customers are completely separate.

Data structures used:

This program makes use of 'structs' and 'vectors'.

- struct Product → This contains product name, price and stock.
- struct Cart_Item → This stores items selected by customer.
- struct UserRecord → This stores login details and role.
- vector< Product > → This stores product list dynamically.
- vector< Cart_Item > → This stores shopping cart for customers.
- vector< UserRecord > → This stores registered users (admins + customers)

Class implementation:

Two main classes are implemented in the program:

1. Customer Class:

• Functions used:

* Private functions:

- load_cart() → Loads cart data from file.
- save_cart() → Saves cart data to file.
- view_cart() → Displays items in the cart.

* Public functions:

- Customer() → It's a constructor, initializes customer and loads cart.
- show_customer_info() → Displays 'username'.

- `checkout()` → It handles billing and stock updates.
- `Customer-menu()` → It provides customer menu (view, add, cart, checkout, logout)

2. Admin Class:

• Functions used:

* Private functions:

- `save-products()` → It saves product list to a file.

* Public functions:

- `Admin()` → It is a constructor which initializes admin.

- `Admin-menu()` → It provides admin menu (add, view, edit, remove, logout)

Global functions:

1. `load_users()` → Loads all registered admins and customers from separate files.
2. `load_products()` → Loads all products from `products.txt`
3. `register_user()` → Registers a new admin or customer.
4. `login()` → Authenticates username and password and assigns role.

Main function:

`main()` → Works as a control center.
Loads data, shows main menu,
calls registration, login and
directs users as per role
in different menus.

when the program runs it first shows the Main Menu:

1. Register
2. Login
3. Exit

- If user chooses 'Register' then a new account is created and a specific role is given (admin or customer).
- If login credentials are checked.
 - if role = admin, Admin-menu() is called.
 - if role = customer, Customer-menu() is called.
- Exit ends the program.

Program Flow:

1. Display Main Menu: Register, Login, Exit
2. On Registration: Validation of username, password and role.

if role = admin, informations will be saved in admin.txt file

if role = customer, informations will be saved in customers.txt file.

3. On Login: Authenticate user and assign the role

4. If role = admin, the user can have access to the Admin Menu - add, view, edit, remove products.

5. If role = customer,

the customer menu /

Customer Menu - View products, Add to cart, View cart, checkout becomes accessible for the user.

6. Checkout updates stock in products.txt
and clears customer's cart.

Sample Program Output:

Main Menu:

===== Main Menu =====

1. Register
2. Login
3. Exit

Enter your choice :

Customer Menu:

===== Customer Menu =====

1. View Products
2. Add to Cart
3. View Cart
4. Checkout
5. Logout

Enter your choice : 1

1. Laptop | Price: 50000 BDT | Stock: 5
2. Monitor | Price: 3000 BDT | Stock: 20

Admin Menu

===== Admin Menu =====

1. Add Product
2. View Products
3. Edit Product
4. Remove Product
5. Logout

Algorithm:

1. Load all users from admin.txt and customer.txt
2. Load all the products from products.txt
3. Display "Main Menu".
4. Registration → Save new user.
5. Login → Validation of credentials
→ Determine role.
6. If Admin → "Admin Menu" opens → Perform actions.
7. If Customer → "Customer Menu" opens
→ Perform actions
8. Customer checkout → Generate bill
→ Update stock

9. Repeat until user chooses Exit.

Conclusion:

The Online Shopping System is a practical application of C++ programming which demonstrates:

1. OOP (Object-Oriented Programming) with classes and objects.
2. Role based access control.
3. Preserving file handling for admin, customers, products and cart data.
4. Menu-driven interface for user friendly navigation.
5. Cart and billing management for customers.

This program teaches real-world programming concepts, improves understanding of OOP, vectors, file handling, and simulates a simple online shopping platform. This program can be used as a foundation for building more advanced online shopping system

applications.