# Project Report: Online Shop Management System

## Submitted to:

Md. Ashraful Haider Chowdhury  
Lecturer  
Department of Computer Science and Engineering

## Submitted by:

1. 2024-3-60-719 Samir Sarower
2. 2024-3-6-394 Md. Sahidul Islam
3. 2024-3-60-416 Rijona Rahman Orthy
4. 2024-3-60-417 Mahdi Ahmed Khan

## Section:

[20]

## Date of Submission:

[20-1-2025]

---

## Table of Contents

1. 1. Abstract
2. 2. Introduction
3. 3. Project Aim and Objectives
4. 4. Flowchart
5. 5. Program Output Screenshot
6. 6. Source Code
7. 7. Limitation of Program
8. 8. Extra Features
9. 9. Conclusion

---

## Abstract

The Online Shop Management System is a comprehensive C program designed to facilitate the core operations of an e-commerce platform. The system provides distinct functionalities for two user roles: Admin and Customer. Admins can efficiently manage inventory, while customers can seamlessly browse and purchase products. This project highlights simplicity and efficiency in managing online store transactions while offering a user-friendly experience.

---

## Introduction

In the era of digitalization, online shopping has become an integral part of daily life. This project addresses the need for a streamlined system to manage the basic functionalities of an online shop. The program provides solutions for inventory management and customer purchases, ensuring smooth operations. By incorporating features like admin authentication and product handling, this system sets the foundation for more advanced e-commerce solutions.

---

## Project Aim and Objectives

\*\*Aim:\*\*

To develop a robust online shop management system leveraging C programming, ensuring simplicity and reliability in operations.

\*\*Objectives:\*\*

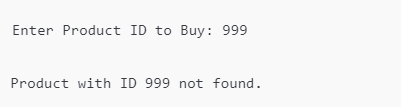
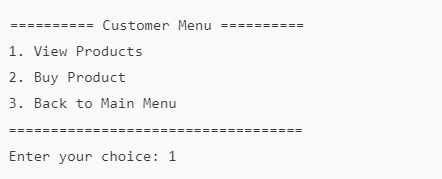
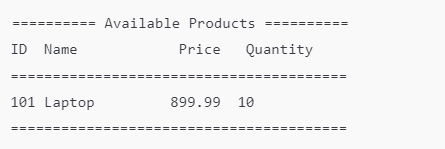
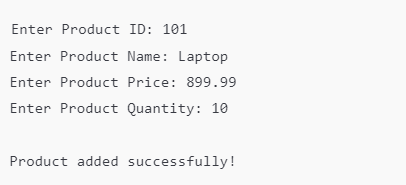
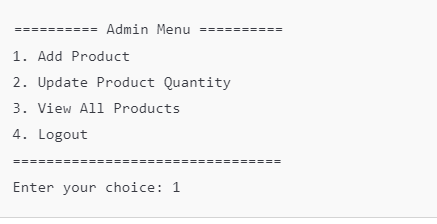
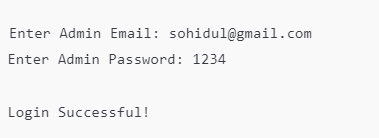
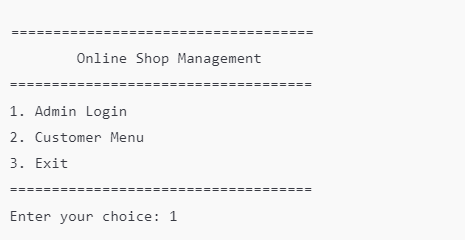
* Establish a secure and reliable login system for Admin users.
* Enable Admins to manage the product inventory through options to add, update, and view products.
* Allow customers to browse available products and make purchases.
* Create a straightforward and intuitive interface for users, minimizing errors and maximizing functionality.

---

## Flowchart

The workflow of the system is as follows:  
  
1. Main Menu: Provides options for Admin Login, Customer Menu, and Exit.  
2. Admin Login:  
 - Validates user credentials.  
 - Grants access to the Admin Menu.  
3. Admin Menu:  
 - Add Product.  
 - Update Product Quantity.  
 - View Products.  
 - Logout.  
4. Customer Menu:  
 - View Products.  
 - Buy Product.  
 - Return to Main Menu.  
5. Exit: Terminates the program.

## Program Output Screenshot



## Source Code

The program’s functionality is implemented using C. Below is an excerpt of the source code:  
  
#include <stdio.h>

#include <string.h>

#define MAX\_PRODUCTS 100

#define MAX\_NAME\_LENGTH 50

typedef struct

{

    int productID;

    char name[MAX\_NAME\_LENGTH];

    float price;

    int quantity;

} Product;

Product products[MAX\_PRODUCTS];

int productCount = 0;

void adminLogin();

void adminMenu();

void customerMenu();

void addProduct();

void updateProductQuantity();

void viewProducts();

void buyProduct();

int main()

{

    int choice;

    while (1)

    {

        printf("\n====================================\n");

        printf("        Online Shop Management       \n");

        printf("====================================\n");

        printf("1. Admin Login\n");

        printf("2. Customer Menu\n");

        printf("3. Exit\n");

        printf("====================================\n");

        printf("Enter your choice: ");

        scanf("%d", &choice);

        switch (choice)

        {

        case 1:

            adminLogin();

            break;

        case 2:

            customerMenu();

            break;

        case 3:

            printf("\nExiting the program. Goodbye!\n");

            return 0;

        default:

            printf("\nInvalid choice. Please try again.\n");

        }

    }

    return 0;

}

void adminLogin()

{

    char email[50], password[50];

    printf("\nEnter Admin Email: ");

    scanf("%s", email);

    printf("Enter Admin Password: ");

    scanf("%s", password);

    if (strcmp(email, "sohidul@gmail.com") == 0 && strcmp(password, "1234") == 0)

    {

        printf("\nLogin Successful!\n");

        adminMenu();

    }

    else

    {

        printf("\nInvalid Email or Password Try Again\n");

    }

}

void adminMenu()

{

    int choice;

    while (1)

    {

        printf("\n========== Admin Menu ==========\n");

        printf("1. Add Product\n");

        printf("2. Update Product Quantity\n");

        printf("3. View All Products\n");

        printf("4. Logout\n");

        printf("================================\n");

        printf("Enter your choice: ");

        scanf("%d", &choice);

        switch (choice)

        {

        case 1:

            addProduct();

            break;

        case 2:

            updateProductQuantity();

            break;

        case 3:

            viewProducts();

            break;

        case 4:

            printf("\nLogging out...\n");

            return;

        default:

            printf("\nInvalid choice. Please try again.\n");

        }

    }

}

void customerMenu()

{

    int choice;

    while (1)

    {

        printf("\n========== Customer Menu ==========\n");

        printf("1. View Products\n");

        printf("2. Buy Product\n");

        printf("3. Back to Main Menu\n");

        printf("===================================\n");

        printf("Enter your choice: ");

        scanf("%d", &choice);

        switch (choice)

        {

        case 1:

            viewProducts();

            break;

        case 2:

            buyProduct();

            break;

        case 3:

            printf("\nReturning to Main Menu...\n");

            return;

        default:

            printf("\nInvalid choice. Please try again.\n");

        }

    }

}

void addProduct()

{

    if (productCount >= MAX\_PRODUCTS)

    {

        printf("\nCannot add more products. Maximum limit reached.\n");

        return;

    }

    Product newProduct;

    printf("\nEnter Product ID: ");

    scanf("%d", &newProduct.productID);

    printf("Enter Product Name: ");

    getchar();

    fgets(newProduct.name, MAX\_NAME\_LENGTH, stdin);

    strtok(newProduct.name, "\n");

    printf("Enter Product Price: ");

    scanf("%f", &newProduct.price);

    printf("Enter Product Quantity: ");

    scanf("%d", &newProduct.quantity);

    products[productCount++] = newProduct;

    printf("\nProduct added successfully!\n");

}

void updateProductQuantity()

{

    int productID, quantity, found = 0;

    printf("\nEnter Product ID to Update Quantity: ");

    scanf("%d", &productID);

    for (int i = 0; i < productCount; i++)

    {

        if (products[i].productID == productID)

        {

            printf("Current Quantity: %d\n", products[i].quantity);

            printf("Enter Quantity to Add: ");

            scanf("%d", &quantity);

            products[i].quantity += quantity;

            printf("\nProduct quantity updated successfully!\n");

            found = 1;

            break;

        }

    }

    if (!found)

    {

        printf("\nProduct with ID %d not found.\n", productID);

    }

}

void viewProducts()

{

    if (productCount == 0)

    {

        printf("\nNo products available.\n");

        return;

    }

    printf("\n========== Available Products ==========\n");

    printf("ID\tName\t\tPrice\t\tQuantity\n");

    printf("========================================\n");

    for (int i = 0; i < productCount; i++)

    {

        printf("%d\t%s\t%.2f\t\t\t\t%d\n", products[i].productID, products[i].name, products[i].price, products[i].quantity);

    }

    printf("========================================\n");

}

void buyProduct()

{

    int productID, quantity, found = 0;

    printf("\nEnter Product ID to Buy: ");

    scanf("%d", &productID);

    for (int i = 0; i < productCount; i++)

    {

        if (products[i].productID == productID)

        {

            found = 1;

            printf("Product Name: %s\n", products[i].name);

            printf("Available Quantity: %d\n", products[i].quantity);

            printf("Enter Quantity to Buy: ");

            scanf("%d", &quantity);

            if (quantity > products[i].quantity)

            {

                printf("\n Out of  stock  Try again.\n");

            }

            else

            {

                products[i].quantity -= quantity;

                float totalPrice = quantity \* products[i].price;

                printf("\nPurchase successful!\n");

                printf("========== Payment Receipt ==========\n");

                printf("Product: %s\n", products[i].name);

                printf("Quantity: %d\n", quantity);

                printf("Total Price: $%.2f\n", totalPrice);

                printf("=====================================\n");

            }

            break;

        }

    }

    if (!found)

    {

        printf("\nProduct with ID %d not found.\n", productID);

    }

}

---

## Limitation of Program

While the system successfully achieves its objectives, it has the following limitations:

* No Data Persistence: Product data is not retained after the program exits.
* Customer Authentication: Customers cannot create accounts or log in.
* Error Handling: Input validations are limited, leading to potential misuse.
* Search Functionality: Users cannot search for specific products by name or category.

---

## Extra Features

* Purchase Receipt: Customers receive a detailed receipt after a successful purchase.
* Secure Admin Access: Admin credentials are verified for access to critical operations.
* Dynamic Updates: Inventory quantities are updated in real-time based on transactions.

---

## Conclusion

The Online Shop Management System demonstrates the effective use of C programming in solving real-world problems. By providing essential features for inventory management and product transactions, the system serves as a prototype for more complex e-commerce platforms. While there are areas for improvement, such as data persistence and enhanced user experience, the project successfully achieves its primary goals and offers a foundation for future advancements.