# ­­BOOK STORE MANAGEMENT SYSTEM



Session: 2021 – 2024

**Submitted by:**

Muhammad Ibrahim       2021-CS-169

**Supervised by:**

Mis Maida Shahid

Department of Computer Science

**University of Engineering and Technology**

**Lahore Pakistan**

# CONTENTS

[Project Description: 3](#_gjdgxs)

[Users of Application:](#_30j0zll) 3

[Functional Requirements:](#_1fob9te) 4

Wireframes of Application5

Data Structures: 11

Functions Prototypes: 11

Functions Working Flow: 13

Complete Code: 14

Test Cases for Project: 37

# PROJECT DESCRIPTION

My project is Book Store Management System. It is a business application which is console based. My basic purpose to make this application is to digitalize Pakistan. It helps book shop owner in multiple ways like

* creating customers’ accounts
* managing the expenditures of shop
* view customers (keeps the track of multiple customers)
* view shopping history
* keep the track of stock present in shop

On other side it also helps the customer in many ways like a customer can

* view all items present in shop,
* a customer can place his order if the required item is not present in shop
* he can also buy items using this application
* can change his pin

This application is very user friendly as it provides separate menu for admin as well as customers.

From security po of view a customer can only access the buying section if his account is created by the admin. In case of wrong user name or pin a customer cannot access to many useful features of customers’ menu.

# USERS OF APPLICATION

There are two types of users of this management system application which are as follows:

* Admin (owner of shop)
* Customer (total customers: 5)

# FUNCTIONAL REQUIREMENTS

**Admin:**

* I can add customers:
  + By name.
  + Give them their pin.
* View customers.
* Add stock to the exiting stock.
* View Bills of the customers.
* View the orders placed by the customers.
* Manage the expenditures of the shop.

**Customers:**

* I can view the items available.
* I can change my pin.
* Place the order If item is not available.
* Buy the required items:
* View the bill of shopping.

# WIRE FRAMES OF APPLICATION

MAIN MENU

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*Book Store Management System\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Welcome To Book Store

You want to login as an ADMIN OR CUSTOMER.

For ADMIN press 1

For CUSTOMER press 2

To EXIT program, press 3

Your option---

Press any key to continue-----

\_

ADMIN MENU

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*Book Store Management System\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Welcome To Book Store

Main Menu-----------------------------------------------

1-Add Customers

2-View Customers

3-View Bills (history)

4-View orders placed by the customers.

5-Add / Update stock of the shop.

6-Manage the expenditures of the shop.

7-Back to main menu.

Your option---

\_

­­­­­­­­­­­­­­­­­­­

ADMIN OPTION-1

Enter name of customer:

\_

Enter pin of customer:

\_

ADMIN OPTION-2

Following(s) are customers and their pins:

Ali 1234

Ahmad 3231

Hassan 3433

Ibrahim 3434

Saad 2233

ADMIN OPTION-3

Followings are the bills of customers:

1-Ali Ahmed 2300

2-Asad 3400

3-Hassan 3500

4-Ibrahim 5000

ADMIN OPTION-4

Followings are the orders placed by the customers:

Ahmad Bahadur Pen, Eraser

Hassan English Book

3-Ibrahim Scale

ADMIN OPTION-5

Stock Management Menu--------------------------------------

1-Add Stock of shop

2-View added stock

3-Update added stock

4-Back to admin menu

Your option---

Following’s things have been added to the shop:

Pen 90

English Book 100

Scale 30

Books 200

Eraser 20

ADMIN OPTION-5(a)

Following’s things have been added to the shop:

Pen 90

English Book 100

Scale 30

Books 200

Eraser 20

Maximum number of stocks you can enter: 10

How many stocks you want to add?

\_

Enter stock name:

\_

Enter stock price:

\_

Your stocks are being added.

ADMIN OPTION-5(b)

Following’s things have been added to the shop:

Pen 90

English Book 100

Scale 30

Books 200

Eraser 20

Following’s things have been added to the shop:

Pen 90

English Book 100

Scale 30

Books 200

Eraser 20

ADMIN OPTION-5(c)

Enter name of stock which you want to update:

\_

Enter name of new stock:

\_

Your stock is updated---

Following’s things have been added to the shop:

Pen 90

English Book 100

Scale 30

Books 200

Eraser 20

ADMIN OPTION-6

Shop Management Menu----------------------------------------

1-Add Bills of shop

2-View added bills

3-Update added bills

Your option---

ADMIN OPTION-6(a)

Enter bill name:

\_

Enter bill amount:

\_

Your bills added

ADMIN OPTION-6(b)

Your bills in ascending order are as show below:

Electricity Bill 12000

Food Bill 30000

Workers Salary 43000

Total expenditure 85000

ADMIN OPTION-6(c)

Enter name of bill which you want to update:

\_

Enter amount of bill:

\_

Your amount is updated---

CUSTOMER MENU

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*Book Store Management System\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Welcome To Book Store

Main Menu--------------------------------------------------

1-View Items

2-Buy the required items

3-Change pin

4-Place orders

5-Back to main menu.

Your option---

CUSTOMER OPTION-1

Following(s) items are available in shop:

Pen 90

English Book 100

Scale 30

Books 200

Eraser 20

CUSTOMER OPTION-2

Enter name

\_

Enter pin:

\_

Enter number of items you want to buy:

\_

Enter item name:

­­­­­­­\_

Total Bill:

CUSTOMER OPTION-3

Enter name:

\_

Enter pin:

\_

Enter your new pin:

\_

Your pin is being updated---

Your pin is updated successfully----

CUSTOMER OPTION-4

Enter name:

\_

Enter pin:

\_

Enter number of orders you want to place (max = 5):

\_

Enter order:

\_

# DATA STRUCTURES

**ARRAYS:**

In this application following arrays are used:

customer\_n\_a[N\_CUSTOMER] = {" "} (names given by admin)

customer\_p\_a[N\_CUSTOMER] = {" "} (pins given by admin)

customers\_bills[N\_CUSTOMER] = {0} (used to store the bills of customers)

stock\_adding[N\_STOCKS] = {" "} (used for adding stock of shop)

bills\_shop[N\_BILLS] = {0}     (used for adding bills of shop)

bills\_shop\_names[N\_BILLS] = {" "} (used to store names of bills)

stock\_prices[N\_STOCKS] = {0} (used for adding prices of stock)

orders[N\_ORDERS] = {" "}           (used for taking orders from customers)

customer\_1[N\_ORDERS] = {" "}       (store the orders placed by customer 1)

customer\_2[N\_ORDERS] = {" "}       (store the orders placed by customer 2)

customer\_3[N\_ORDERS] = {" "}       (store the orders placed by customer 3)

customer\_4[N\_ORDERS] = {" "}       (store the orders placed by customer 4)

customer\_5[N\_ORDERS] = {" "}       (store the orders placed by customer 5)

# FUNCTIONS PROTOTYPES

void header ()

int main\_menu()

**FOR ADMIN SECTION:**

int admin\_menu()

void add\_customer()

void add\_customer\_array(string name, string pin)

void display\_customer()

void display\_customers\_bills()

void display\_order(string array[])

void view\_orders()

int stock\_management\_menu()

void adding\_stock\_with\_price()

void view\_stock()

void update\_stock()

int shop\_management\_menu()

void add\_bills\_shop()

void view\_bills\_shop()

void update\_bill()

void sorting(int list[], string names[], int array\_size)

**FOR CUSTOMER SECTION**

int customer\_menu()

void view\_items()

void clear\_screen()

int calculateBill(string item)

void bill\_generator()

void pin\_change()

void order\_placing()

**FOR FILE HANDLING**

loadcustomer()

savecustomerInFile(string name, string pin)

loadcustomer\_bills()

save\_customer\_n\_bill\_infile(string name, int bill)

loadstockandprice()

savestockInFile()

# FUNCTIONS WORKING FLOW

# COMPLETE CODE

#include <iostream>

#include <stdlib.h>

#include <conio.h>

#include <fstream>

//----------------------PROBLEM IS IN BREAKOUTS RED DOTS CHECK IT--------------

using namespace std;

// CONST VARIABLES

int const N\_CUSTOMER = 5;

int const N\_STOCKS = 10;

int const N\_BILLS = 4;

int const N\_ORDERS = 10;

// GLOBAL VARIABLES

int count = 0;

int user\_option;

int entry\_count = 0;

int stock\_count = 0;

int shop\_bill\_count = 0;

int customer\_display = 0; // used for counting customer while displaying

int bill\_count = 0;

int order\_place\_count = 0;

int pin\_change\_count = 0;

// GLOBAL ARRAYS

string customer\_n\_a[N\_CUSTOMER] = {" "}; // names given by admin

string customer\_p\_a[N\_CUSTOMER] = {" "}; // pins given by admin

int customers\_bills[N\_CUSTOMER] = {0}; // used to store the bills of customers.

string stock\_adding[N\_STOCKS] = {" "}; // used for adding stock of shop.

int bills\_shop[N\_BILLS] = {0}; // used for adding bills of shop.

string bills\_shop\_names[N\_BILLS] = {" "}; // used to store names of bills.

int stock\_prices[N\_STOCKS] = {0}; // used for adding prices of stock.

string orders[N\_ORDERS] = {" "}; // used for taking orders from customers

string customer\_1[N\_ORDERS] = {" "}; // used to store the orders placed by customer 1.

string customer\_2[N\_ORDERS] = {" "}; // used to store the orders placed by customer 2.

string customer\_3[N\_ORDERS] = {" "}; // used to store the orders placed by customer 3.

string customer\_4[N\_ORDERS] = {" "}; // used to store the orders placed by customer 4.

string customer\_5[N\_ORDERS] = {" "}; // used to store the orders placed by customer 5.

// PROTOTYPES 0F FUNCTIONS

void header();

int main\_menu();

//------------------------FOR FILE HANDLING---------------------------

void loadcustomer(); // usedto load customers with pins from file

void savecustomerInFile(string name, string pin); // used to store data back to file

// used for customer billing section

void loadcustomer\_bills();

void save\_customer\_n\_bill\_infile(string name, int bill);

// used for adding stock in file

void loadstockandprice();

void savestockInFile();

// used for adding shop bills in file

// void loadbills\_shop();

// void saveshopbillsInFile(string name, int prices);

//------------------------FOR ADMIN SECTION---------------------------

int admin\_menu();

void add\_customer();

void add\_customer\_array(string name, string pin);

void display\_customer();

void display\_customers\_bills();

void display\_order(string array[]);

void view\_orders();

int stock\_management\_menu();

void adding\_stock\_with\_price();

void add\_stock\_in\_array(string name, int price);

void view\_stock();

void update\_stock();

int shop\_management\_menu();

void add\_bills\_shop();

void add\_shopbills\_array(string name, int amount);

void view\_bills\_shop();

void update\_bill();

void sorting(int list[], string names[], int array\_size);

//------------------------FOR CUSTOMER SECTION---------------------------

int customer\_menu();

void view\_items();

void clear\_screen();

int calculateBill(string item);

void bill\_generator();

void store\_bill(int bill, int count);

void pin\_change();

void order\_placing();

//------START OF MAIN-----------------

main()

{

// local VARIABLES

int main\_option;

int admin\_option;

int customer\_option;

//---------load files-----------

loadcustomer();

loadcustomer\_bills();

loadstockandprice();

// loadbills\_shop();

//\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_END OF LOCAL VARIABLES\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

while (true)

{

main\_option = main\_menu();

clear\_screen();

if (main\_option == 1) //------------start of ADMIN SECTION--------

{

while (true)

{

admin\_option = admin\_menu();

clear\_screen();

//------------start of ADMIN OPTIONS--------

if (admin\_option == 1) // for adding customers

{

add\_customer();

clear\_screen();

}

else if (admin\_option == 2) // View Customers

{

if (customer\_n\_a[0] != " " && customer\_p\_a[0] != " ")

{

display\_customer();

clear\_screen();

}

else

{

cout << "No customer is being added yet" << endl;

cout << endl;

clear\_screen();

}

}

else if (admin\_option == 3) // View Bills (history)

{

if (customer\_n\_a[0] != " " && customers\_bills[0] != 0)

{

display\_customers\_bills();

clear\_screen();

}

else

{

cout << "No customer has done shoping yet" << endl;

cout << endl;

clear\_screen();

}

}

else if (admin\_option == 4) // View odders placed by the customers

{

if (customer\_1[1] != " ")

{

view\_orders();

clear\_screen();

}

else

{

cout << "No order has been placed." << endl;

cout << endl;

clear\_screen();

}

}

else if (admin\_option == 5) // Add / Update stock of the shop

{

int Stock\_menu\_option;

while (true)

{

Stock\_menu\_option = stock\_management\_menu();

clear\_screen();

if (Stock\_menu\_option == 1)

{

adding\_stock\_with\_price();

clear\_screen();

}

else if (Stock\_menu\_option == 2)

{

if (stock\_prices[0] != 0 && stock\_adding[0] != " ")

{

view\_stock();

clear\_screen();

}

else

{

cout << "No stock has been added yet." << endl;

clear\_screen();

}

}

else if (Stock\_menu\_option == 3)

{

update\_stock();

clear\_screen();

}

else if (Stock\_menu\_option == 4)

{

savestockInFile();

break;

}

else

{

cout << "Enter a valid input" << endl;

clear\_screen();

break;

}

}

}

else if (admin\_option == 6) // Manage the expenditures of the shop

{

int Shop\_menu\_option;

while (true)

{

Shop\_menu\_option = shop\_management\_menu();

clear\_screen();

if (Shop\_menu\_option == 1)

{

add\_bills\_shop();

clear\_screen();

}

else if (Shop\_menu\_option == 2)

{

if (bills\_shop\_names[0] != " ")

{

view\_bills\_shop();

clear\_screen();

}

else

{

cout << "No bills are added by the admin." << endl;

cout << endl;

clear\_screen();

}

}

else if (Shop\_menu\_option == 3)

{

update\_bill();

clear\_screen();

}

else if (Shop\_menu\_option == 4)

{

break;

}

else

{

cout << "Enter a valid input" << endl;

cout << endl;

clear\_screen();

break;

}

}

}

else if (admin\_option == 7) // exit from admin section

{

break;

}

else

{

cout << "Invalid input" << endl;

cout << endl;

clear\_screen();

break;

}

} // END OF WHILE LOOP

} // END OF ADMIN SECTION

else if (main\_option == 2) //------------start of CUSTOMER SECTION--------

{

while (true)

{

customer\_option = customer\_menu();

clear\_screen();

//------------start of CUSTOMER OPTIONS--------

if (customer\_option == 1) // View Items

{

if (stock\_adding[0] != " ")

{

view\_items();

clear\_screen();

}

else

{

cout << "There is no stock to display" << endl;

cout << endl;

clear\_screen();

}

}

else if (customer\_option == 2) // Buy the required items

{

bill\_generator();

clear\_screen();

}

else if (customer\_option == 3) // Change pin

{

pin\_change();

clear\_screen();

}

else if (customer\_option == 4) // Place odders

{

order\_placing();

clear\_screen();

}

else if (customer\_option == 5) // exit from customer section

{

break;

}

else

{

cout << "Invalid input" << endl;

cout << endl;

clear\_screen();

break;

}

} // END OF WHILE LOOP

} // END OF CUSTOMER SECTION

else if (main\_option == 3) //-------- EXIT FROM THE PROGRAM------

{

cout << "Thanks for using BOOK STORE MANAGEMENT SYSTEM." << endl;

cout << endl;

break;

}

else //--------------IN CASE OF WRONG INPUT OF MAIN MENU--------

{

cout << "Enter a valid input" << endl;

clear\_screen();

continue;

}

} //----------END OF WHILE LOOP------------------

} // END OF MAIN

//--------------FUNCTIONS-------------------

void header()

{

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* " << endl;

cout << " \*Book Store Management System\* " << endl;

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* " << endl;

}

int main\_menu()

{

header();

cout << " " << endl;

cout << " Welcome To Book Store " << endl;

cout << " " << endl;

cout << "You want to login as an ADMIN OR CUSTOMER. " << endl;

cout << " " << endl;

cout << "For ADMIN press 1. " << endl;

cout << "For CUSTOMER press 2. " << endl;

cout << "To EXIT program, press 3. " << endl;

cout << " " << endl;

// OPTION FROM USER.

cout << "Your option--- " << endl;

cin >> user\_option;

return user\_option;

}

//------------------------FOR ADMIN SECTION---------------------------

int admin\_menu()

{

header();

cout << " " << endl;

cout << " Welcome To Book Store " << endl;

cout << " " << endl;

cout << "Main Menu--------------------------------------------------------" << endl;

cout << " " << endl;

cout << "1-Add Customers" << endl;

cout << "2-View Customers" << endl;

cout << "3-View Bills (history)" << endl;

cout << "4-View odders placed by the customers." << endl;

cout << "5-Add / Update stock of the shop." << endl;

cout << "6-Manage the expenditures of the shop." << endl;

cout << "7-Back to main menu." << endl;

cout << "Your option--- " << endl;

cin >> user\_option;

return user\_option;

}

void loadcustomer()

{

fstream file;

file.open("names\_pins.txt", ios::in);

string name, pin;

entry\_count = 0;

while (!file.eof())

{

file >> name;

file >> pin;

add\_customer\_array(name, pin);

}

file.close();

}

void savecustomerInFile(string name, string pin)

{

fstream file;

file.open("names\_pins.txt", ios::out);

file << "\n"

<< name << " " << pin;

// cout << "Record saved into file " << endl;

file.close();

}

void add\_customer()

{

if (entry\_count < 5)

{

string name, pin;

cout << "Enter customer name" << endl;

cin >> name;

cout << "Enter customer pin" << endl;

cin >> pin;

add\_customer\_array(name, pin);

savecustomerInFile(name, pin);

}

else

{

cout << "Only " << N\_CUSTOMER << " can be added " << endl;

}

}

void add\_customer\_array(string name, string pin)

{

customer\_n\_a[entry\_count] = name;

customer\_p\_a[entry\_count] = pin;

entry\_count++;

}

void display\_customer()

{

if (entry\_count < N\_CUSTOMER)

{

cout << "Name\tPin" << endl;

for (int i = 0; i < entry\_count; i++)

{

cout << customer\_n\_a[i] << "\t" << customer\_p\_a[i] << endl;

}

}

else

{

cout << "Name\tPin" << endl;

for (int i = 0; i < N\_CUSTOMER; i++)

{

cout << customer\_n\_a[i] << "\t" << customer\_p\_a[i] << endl;

}

}

}

void display\_customers\_bills()

{

cout << "Following(s) are the customers and their bills : " << endl;

cout << "Name\t\t\tBill" << endl;

if (bill\_count < N\_CUSTOMER)

{

for (int i = 0; i <= bill\_count; i++)

{

if (customer\_n\_a[i] != " " && customers\_bills[i] != 0)

cout << customer\_n\_a[i] << "\t\t\t" << customers\_bills[i] << endl;

}

}

else

{

for (int i = 0; i <= N\_CUSTOMER - 1; i++)

{

cout << customer\_n\_a[i] << "\t\t\t" << customers\_bills[i] << endl;

}

}

}

void display\_order(string array[]) // used in another function for displaying array.

{

int index = 0;

while (array[index] != "\0")

{

cout << array[index] << " ";

index++;

}

}

void view\_orders()

{

if (orders[0] != " ")

{

cout << "Following are the orders." << endl;

cout << endl;

if (customer\_1[0] != " " && customer\_1[0] != " ")

{

cout << customer\_n\_a[0] << "\t";

display\_order(customer\_1);

cout << endl;

}

if (customer\_2[0] != " ")

{

cout << customer\_n\_a[1] << "\t";

display\_order(customer\_2);

cout << endl;

}

if (customer\_3[0] != " ")

{

cout << customer\_n\_a[2] << "\t";

display\_order(customer\_3);

cout << endl;

}

if (customer\_4[0] != " ")

{

cout << customer\_n\_a[3] << "\t";

display\_order(customer\_4);

cout << endl;

}

if (customer\_5[0] != " ")

{

cout << customer\_n\_a[4] << "\t";

display\_order(customer\_5);

cout << endl;

}

}

else

{

cout << "No order have been placed yet." << endl;

}

}

int stock\_management\_menu()

{

header();

cout << " " << endl;

cout << "Stock Management Menu--------------------------------------------------------" << endl;

cout << " " << endl;

cout << "1-Add Stock of shop" << endl;

cout << "2-View added stock " << endl;

cout << "3-Update added stock" << endl;

cout << "4-Back to admin menu" << endl;

cout << "Your option--- " << endl;

cin >> user\_option;

return user\_option;

}

void loadstockandprice()

{

fstream file;

file.open("stocks\_of\_shop.txt", ios::in);

string name;

int price;

// stock\_count = 0;

while (!file.eof())

{

file >> name;

file >> price;

add\_stock\_in\_array(name, price);

}

file.close();

}

void add\_stock\_in\_array(string name, int price)

{

stock\_adding[stock\_count] = name;

stock\_prices[stock\_count] = price;

stock\_count++;

}

void savestockInFile()

{

fstream file;

int index = 0;

file.open("names\_pins.txt", ios::out);

while (index < stock\_count)

{

file << "\n"

<< stock\_adding[index] << " " << stock\_prices[index];

index++;

}

// cout << "Record saved into file " << endl;

file.close();

}

void adding\_stock\_with\_price()

{

if (stock\_count < 10)

{

int n\_stock;

cout << "Maximum number of stocks you can enter : " << N\_STOCKS << endl;

cout << "How many stocks you want to add : ";

cin >> n\_stock;

string name;

int prices;

for (int i = 0; i < n\_stock; i++)

{

cout << "Enter stock " << i + 1 << " : ";

cin >> name;

cout << "Enter stock " << i + 1 << " price : ";

cin >> prices;

add\_stock\_in\_array(name, prices);

}

cout << "You stocks are being added." << endl;

}

else

{

cout << "You can only add " << N\_STOCKS << " stocks" << endl;

}

}

void view\_stock()

{

cout << "Following are the stocks added by ADMIN." << endl;

cout << "Item\t\tPrice(Rs)" << endl;

for (int i = 0; i < stock\_count; i++)

{

cout << stock\_adding[i] << "\t\t" << stock\_prices[i] << endl;

}

}

void update\_stock()

{

string stock\_name, stock\_name\_final;

cout << "Enter name of stock which you want to update : ";

cin >> stock\_name;

cout << "Enter name of new stock : ";

cin >> stock\_name\_final;

int index = 0;

while (stock\_adding[index] != "\0")

{

if (stock\_adding[index] == stock\_name)

{

stock\_adding[index] = stock\_name\_final;

}

index++;

}

cout << "Your stock is updated---" << endl;

}

int shop\_management\_menu()

{

header();

cout << " " << endl;

cout << "Shop Management Menu--------------------------------------------" << endl;

cout << " " << endl;

cout << "1-Add Bills of shop" << endl;

cout << "2-View added bills " << endl;

cout << "3-Update added bills" << endl;

cout << "4-Back to admin menu" << endl;

cout << "Your option--- " << endl;

cin >> user\_option;

return user\_option;

}

void loadbills\_shop()

{

fstream file;

file.open("bills\_of\_shop.txt", ios::in);

string name;

int amount;

shop\_bill\_count = 0;

while (!file.eof())

{

file >> name;

file >> amount;

add\_stock\_in\_array(name, amount);

}

file.close();

}

void add\_shopbills\_array(string name, int amount)

{

bills\_shop\_names[shop\_bill\_count] = name;

bills\_shop[shop\_bill\_count] = amount;

shop\_bill\_count++;

}

void saveshopbillsInFile(string name, int prices)

{

fstream file;

file.open("bills\_of\_shop.txt", ios::app);

file << "\n"

<< name << " " << prices << endl;

// cout << "Record saved into file " << endl;

file.close();

}

void add\_bills\_shop()

{

if (shop\_bill\_count < N\_BILLS)

{

string bill\_name;

int bill\_amount;

for (int i = 0; i < N\_BILLS; i++)

{

cout << "Enter bill " << i + 1 << " name : ";

cin >> bill\_name;

cout << "Enter " << bill\_name << " bill amount : ";

cin >> bill\_amount;

add\_shopbills\_array(bill\_name, bill\_amount);

saveshopbillsInFile(bill\_name, bill\_amount);

}

cout << "Your bills added" << endl;

}

else

{

cout << "You can only add " << N\_BILLS << " types of bills" << endl;

}

}

void sorting(int list[], string names[], int array\_size)

{

int min;

int i;

int j;

int loc;

int temp;

string temp\_name;

for (i = 0; i < array\_size - 1; i++)

{

min = list[i];

loc = i; // INDEX OF LIST[i].

for (j = i + 1; j < array\_size; j++)

{

if (list[j] < min)

{

min = list[j];

loc = j;

}

temp = list[i]; // IT STORES THE VALUE OF FIRST ELEMENT OF ARRAY IN SEPERATE VARIABLE.

temp\_name = names[i]; // FOR NAME SWAPING

list[i] = list[loc]; // IT PLACE THE SMALLEST VALUE TO FIRST PLACE.

names[i] = names[loc]; // FOR NAME SWAPING

list[loc] = temp; // IT PLACE THE ELEMENT PLACED IN SEPERATE VARIABLE TO THE LOCATION WHERE SMALLEST VALUE WAS FOUND.

names[loc] = temp\_name; // FOR NAME SWAPING

}

}

}

void view\_bills\_shop()

{

cout << "Your bills in asseding order are as show below : " << endl;

cout << "Name\t\tAmount(Rs)" << endl;

sorting(bills\_shop, bills\_shop\_names, N\_BILLS);

int total;

for (int i = 0; i < N\_BILLS; i++)

{

cout << bills\_shop\_names[i] << ":\t\t" << bills\_shop[i] << endl;

total = total + bills\_shop[i];

}

cout << "Total expenditure"

<< ":\t\t" << total - 4 << endl;

}

void update\_bill()

{

string bill\_name;

int bill\_amount\_final;

cout << "Enter name of bill which you want to update : ";

cin >> bill\_name;

cout << "Enter amount of bill : ";

cin >> bill\_amount\_final;

for (int index = 0; index < N\_BILLS; index++)

{

if (bills\_shop\_names[index] == bill\_name)

{

bills\_shop[index] = bill\_amount\_final;

}

}

cout << "Your amount is updated---" << endl;

}

//------------------------FOR CUSTOMER SECTION---------------------------

int customer\_menu()

{

header();

cout << " " << endl;

cout << " Welcome To Book Store " << endl;

cout << " " << endl;

cout << "Main Menu--------------------------------------------------------" << endl;

cout << " " << endl;

cout << "1-View Items" << endl;

cout << "2-Buy the required items" << endl;

cout << "3-Change pin" << endl;

cout << "4-Place odders." << endl;

cout << "5-Back to main menu." << endl;

cout << "Your option--- " << endl;

cin >> user\_option;

return user\_option;

}

void view\_items()

{

cout << "Following(s) items are available in shop:" << endl;

cout << endl;

int index = 0;

cout << "Item\t\tPrice(Rs)" << endl;

while (stock\_adding[index] != "\0")

{

cout << stock\_adding[index] << "\t\t" << stock\_prices[index] << endl;

index++;

}

}

int calculateBill(string item)

{

int price;

int index = 0;

while (stock\_adding[index] != "\0")

{

if (item == stock\_adding[index])

{

price = stock\_prices[index];

}

index++;

}

return price;

}

void bill\_generator()

{

string customer\_n, customer\_p;

cout << "Enter name : ";

cin >> customer\_n;

cout << "Enter pin : ";

cin >> customer\_p;

if (customer\_n\_a[bill\_count] == customer\_n && customer\_p\_a[bill\_count] == customer\_p)

{

int n;

int bill;

int t\_bill = 0;

string item;

cout << "Enter number of items you want to buy : " << endl;

cin >> n;

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

{

cout << "Enter item name:";

cin >> item;

bill = calculateBill(item);

t\_bill = t\_bill + bill;

}

cout << "Total Bill :\t" << t\_bill << endl;

store\_bill(t\_bill, bill\_count);

save\_customer\_n\_bill\_infile(customer\_n, t\_bill);

bill\_count++;

}

else

{

cout << "User\_name or pin is wrong." << endl;

}

}

void store\_bill(int bill, int count)

{

customers\_bills[count] = bill;

}

void loadcustomer\_bills()

{

fstream file;

file.open("names\_bills.txt", ios::in);

string name;

int bill;

bill\_count = 0;

while (!file.eof())

{

file >> name;

file >> bill;

store\_bill(bill, bill\_count);

}

file.close();

}

void save\_customer\_n\_bill\_infile(string name, int bill)

{

fstream file;

file.open("names\_bills.txt", ios::app);

file << "\n"

<< name << " " << bill << endl;

cout << "Record saved into file " << endl;

file.close();

}

void pin\_change()

{

string customer\_n, customer\_p;

cout << "Enter name : ";

cin >> customer\_n;

cout << "Enter pin : ";

cin >> customer\_p;

if (customer\_n\_a[pin\_change\_count] == customer\_n && customer\_p\_a[pin\_change\_count] == customer\_p)

{

string customer\_p\_changed;

cout << "Enter your new pin : ";

cin >> customer\_p\_changed;

customer\_p\_a[pin\_change\_count] = customer\_p\_changed;

cout << "Your pin is being updated---" << endl;

cout << "Your pin is updated successfully----" << endl;

pin\_change\_count++;

}

else

{

cout << "User\_name or pin is wrong " << endl;

}

}

void order\_placing()

{

string customer\_n, customer\_p;

cout << "Enter name : ";

cin >> customer\_n;

cout << "Enter pin : ";

cin >> customer\_p;

if (customer\_n\_a[order\_place\_count] == customer\_n && customer\_p\_a[order\_place\_count] == customer\_p)

{

int n;

cout << "Enter number of orders you want to place (max = 5) : ";

cin >> n;

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

{

cout << "Enter order " << i + 1 << " : ";

cin >> orders[i];

}

// cout << "Nmae " << customer\_n << endl;

// cout << "Name in array " << customer\_n\_a[0] << endl;

// cout << "Checking array:" << endl;

// for (int i = 0; i < n; i++)

// {

// cout << orders[i] << " ";

// }

// cout << endl;

if (customer\_n == customer\_n\_a[0])

{

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

{

customer\_1[i] = orders[i];

}

}

else if (customer\_n == customer\_n\_a[1])

{

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

{

customer\_2[i] = orders[i];

}

}

else if (customer\_n == customer\_3[0])

{

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

{

customer\_3[i] = orders[i];

}

}

else if (customer\_n == customer\_4[0])

{

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

{

customer\_4[i] = orders[i];

}

}

else if (customer\_n == customer\_5[0])

{

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

{

customer\_5[i] = orders[i];

}

}

order\_place\_count++;

}

else

{

cout << "User\_name or pin is wrong " << endl;

}

}

//------------------------FOR CLEARING SCREEN---------------------------

void clear\_screen()

{

cout << "Press any key to continue.";

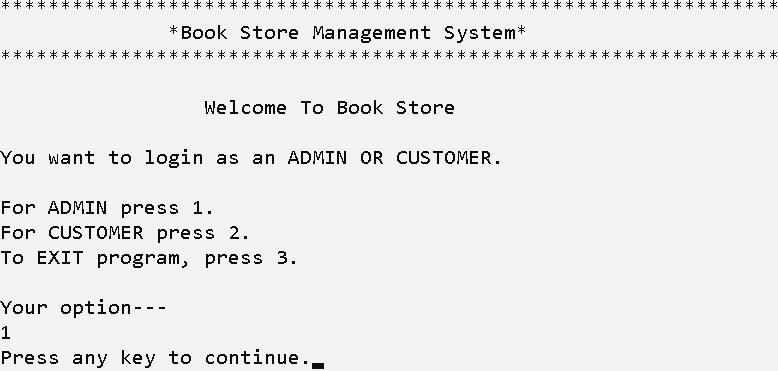
getch();

system("cls");

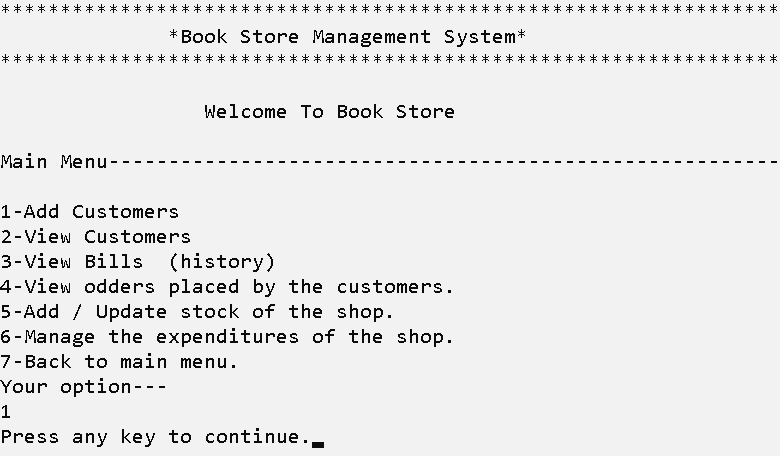
}

# TEST CASES FOR PROJECT

**MAIN MENU**

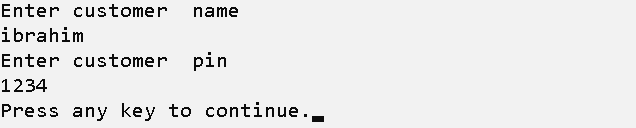


**ADMIN MENU**

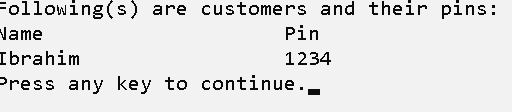


**ADMIN SECTION**

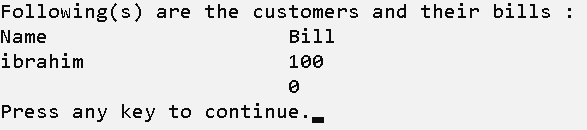
**OPTION 1**



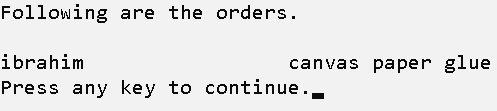
**OPTION 2**



**OPTION 3**



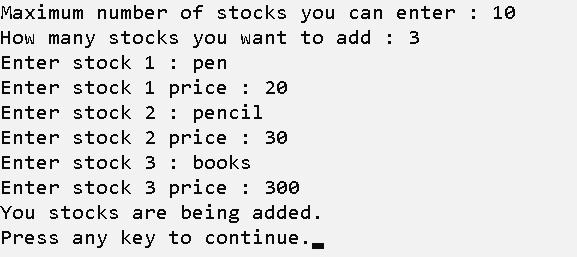
**OPTION 4**



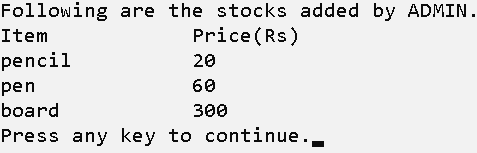
**OPTION 5**(**STOCK MANAEMENT MENU)**



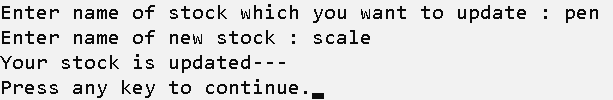
**Option 1**



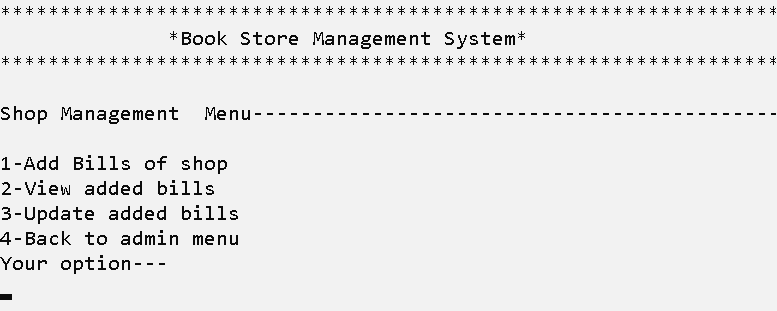
**Option 2**



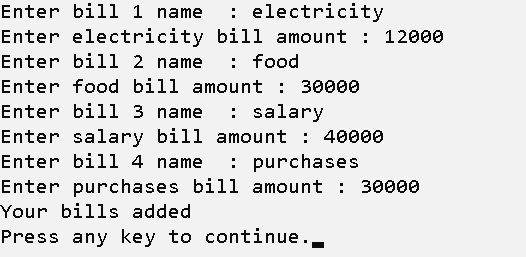
**Option 3**



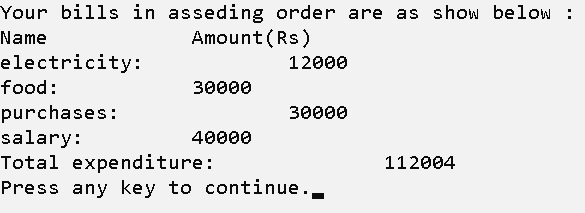
**OPTION 6**(**SHOP MANAGEMENT MENU**)



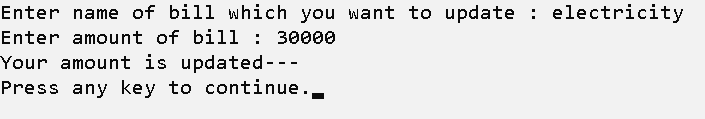
**Option 1**



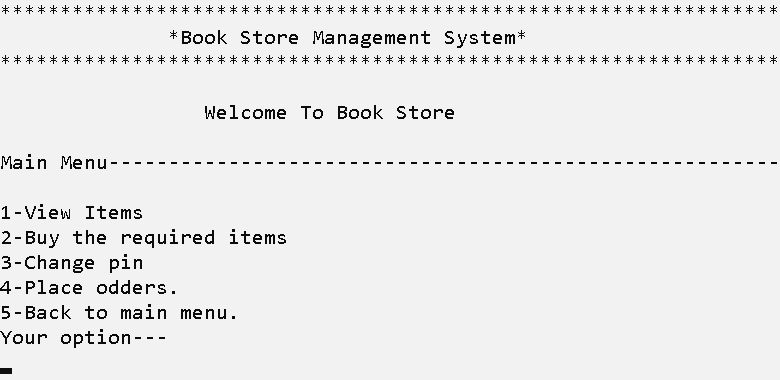
**Option 2**



**Option 3**

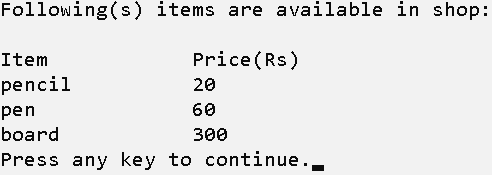


**CUSTOMER MENU**

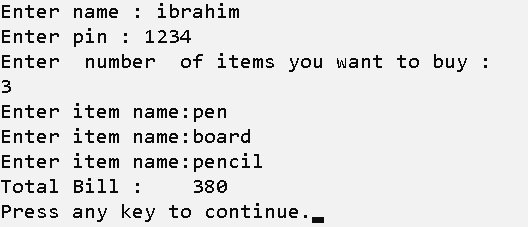


**CUSTOMER SECTION**

**OPTION 1**



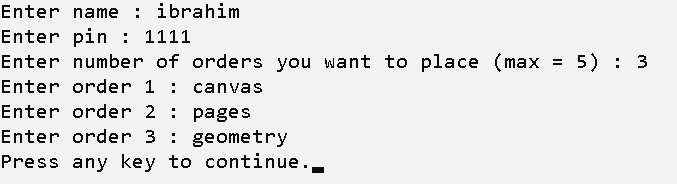
**OPTION 2**



**OPTION 3**

****

**OPTION 4**



**END SECTION**

