**DATA STRUCTURE AND ALGORITHM**

**PROJECT REPORT**

**BOOK SHOP MANAGEMENT SYSTEM**



**GROUP MEMBERS:**

|  |  |
| --- | --- |
| **ALI FAWAD** | **02-134191-094** |
| **BASIL ASLAM** | **02-134191-002** |

Contents

[**ACKNOWLEDGEMENT: 4**](#_Toc28239411)

[**ABSTRACT: 4**](#_Toc28239412)

[**PROJECT IDEA: 5**](#_Toc28239413)

[**PROJECT OVERVIEW: 5**](#_Toc28239414)

* [Purpose: **5**](#_Toc28239415)
* [Scope: **5**](#_Toc28239416)

[**INTRODUCTION 5**](#_Toc28239418)

[**BACKGROUND** 6](#_Toc28239419)

[**PROBLEMS FACED BY EXISTING MANUAL SYSTEM** 7](#_Toc28239420)

* [Difficulty in Maintenance of Records 7](#_Toc28239421)
* [Time Consuming 8](#_Toc28239422)
* [Editing of Data 8](#_Toc28239423)
* [Data Insecurity 8](#_Toc28239424)
* [Report Generation 8](#_Toc28239425)
* [High Data Redundancy 8](#_Toc28239426)
* [Data Inconsistency 9](#_Toc28239427)

[**COMPUTERIZED SYSTEM 9**](#_Toc28239428)

[**FEATURES OF PROPOSED SYSTEM 10**](#_Toc28239429)

[**CONCLUSION: 11**](#_Toc28239430)

[**Functionalities Used: 12**](#_Toc28239431)

* [int main ():](#_Toc28239433) 12
* [login\_option: 13](#_Toc28239434)
* [admin\_staff\_login:](#_Toc28239435) 16
* [Admin ():](#_Toc28239436) 19
* [Staff ():](#_Toc28239437) 23
* [Customer ():](#_Toc28239438) 27
* [add\_book ():](#_Toc28239439) 29
* [edit\_book ():](#_Toc28239440) 33
* [delete\_book ():](#_Toc28239441) 41
* [view\_stock ():](#_Toc28239442) 38
* [search\_book ()](#_Toc28239443) 44
* [Add\_Staff\_Details ():](#_Toc28239444) 51
* [view\_Staff \_Details ():](#_Toc28239445) 52
* [generate\_bill ():](#_Toc28239441) 55
* [All\_bill:](#_Toc28239442) 61

[**Additional Functionalities:**](#_Toc28239432) **63**

* [print\_select\_one ():](#_Toc28239433) 63
* [print\_select\_valid ():](#_Toc28239434) 64
* [time ():](#_Toc28239435) 64
* [header ():](#_Toc28239436) 65
* [print\_searched ():](#_Toc28239438) 65
* [edited\_successfully ():](#_Toc28239439) 66
* [fontsize ():](#_Toc28239439) 67

# **ACKNOWLEDGEMENT:**

The efforts which we made to complete project were done by our team but it could not have been possible to complete this project without getting help from individuals.

This project consumed huge amount of work, research and dedication. Still, implementation would not have been possible if we did not have a support of some individuals. Therefore we would like to extend our sincere gratitude to all of them.

First of all we are thankful to Lab Engineer Sir Marouf for their support and for providing necessary guidance concerning projects implementation.

We are also grateful to Bahria University for provision of expertise, and technical support in the implementation. Without their superior knowledge and experience, the Project would like in quality of outcomes, and thus their support has been essential.

We would like to express our sincere thanks towards our Data Structure and Algorithm Professor Miss Lubna Siddique who devoted their time and knowledge in the implementation of this project.

Nevertheless, we express our gratitude toward our families and colleagues for their kind co-operation and encouragement which help us in completion of this project.

**Abstract**

The goal of this project is to design a bookshop that mainly sells historical, books related toeach subject and technical books. The book inventories are stored in database. Customers will be able to search the books they want, check the availability, and place the order to buy the book by approaching the salesman in the shop.

Manual system of purchasing of books is so difficult to keep record of available books for selling books to a customer. So the purpose of this purchasing of books is to make the work of buyer, seller and company books easier by keeping record of available books of different subjects & authors and provide details of the books to customers in a quick manner.

## 

## **PROJECT IDEA**

Having an eye on the current processing of most of the bookshops in our surroundings, having very poor system of keeping the record of each and every book and the sales record as well. The idea was to build an application based solution to get rid of the manual systems by letting computers work instead of humans, according to the idea the results will can be highly attractive as it may save the time as well as money which is wasted in huge amount each your by following the old assessment systems.

### **PROJECT OVERVIEW:**

### **Purpose:**

* Maintaining the Database for books, amount collect on daily basis, etc.
* Removal of Data Redundancy.
* Data Consistency.
* Ensure data security.

### **Scope:**

### The project would be very useful to manage book shop data, that it will manage various management activities in the book shop. Book Shop Management system is the system that manages the book inventory, add a book in the inventory, searching the book, update the book details as well quantity, remove any book from the inventory and manages the track of each book on daily basis with a perfect record.

**Introduction**

The Book Shop management system is to provide a relief to the staff, management and as well as customers also by using computer application and database. Computer application and utilization can be seen in almost all working areas. Computer has easier for men as all those hectic tasks, once done by the man, have been taken over by computers. This has reduced time consumption by manual procedure to a very high level. Besides this, there is always a very low risk of error; inconsistency and loss of information in computer operate calculations and computer driven procedures. Although there are certain computer disadvantages of computer usage as well, such as inefficiency due to loss of data accidently, need for having skilled labor that is computer literate people and expenditure upon computer and its accessories , which make it expensive. But these problems can be rectified by using various techniques or strategies and taking different measures and, thus computer can never be compromised for its services to the mankind.

Computer’s database system has become more popular in this regard. Due to more functionality and support offered by computers, almost all organizations and institution prefer to use computer in order to maintain their databases.

According to the above facts, managing and maintaining a book shop could also be controlled by efficient software. This project focuses attention on designing efficient and reliable software which controls the transactions and stock of a bookshop.

When we are concerning the manual process of a bookshop, the major problem is the waste of time. A customer has to waste his/her valuable time when he needs to buy a book as all the events such as searching, purchasing are done by members of the staff .In briefly, the manual process is very slow. But automation will reduce the time taken in the whole process.

In this project there will be some restrictions on the information for customer and staff as well. The staff will not have the access to update the stock as updating the stock can performed by the admin and the customer can only check the stock. When customer enters the shop he/she will first check the stock, that the book which he requires is available or not. If the book is available then the customer will go and take that book with the help of ISBN number of the book. The customer will then carry all the books to our staff which he/she wish to buy. The staff will issue all that books to the customer and at last he/she will pay the bill and get the slip.

The main purpose of the project is to have an automated system for customer dealings as well as for the stock. This automated system will result as an accurate system without any errors. The customer will also be provided the best services as regarding to stock and customer representative.

When the bookshop issues an item to a customer, all the stages of the transaction procedure will be facilitated by the system & it will be more accurate.

**Background**

The Bookshop Management System is to automate all operations in a bookshop. Generally it includes the Order Processing, Stock Management and Accounts Management.

Here we are try to develop such type system which is provides a system to manage all the operations of bookshop. That means a shop which has the type system which provides the facility to the customers of the shop to purchase the books from the shop without any complexity.

For example any customer wants to purchase any book from the shop than first of all customer just choose the stream of the book than he/she can see the more than one type of books there and then he/she can choose the specific book from there. And then purchase it by paying price on bookshop cash counter and receives its invoice.

**PROBLEMS FACED BY EXISTING MANUAL SYSTEM**

The phase of system analysis process deals with problems that are affecting in the current manual system. The problems are those, which are affecting the book shop in daily routine work.

As the growing trend in InfoTech World of computers need of accuracy, perfectness, speed and high memory data storage is a must. Each and every problem must be solved with a least amount of time and energy.

The problems faced by existing system are described as below:

* Difficulty in Maintenance of Records.
* Time Consuming.
* Editing of data becomes a tedious job.
* No Security of Data.
* Mistakes Occurring in long Calculations [Annual selling calculation, daily basis calculations, etc.]
* Proper Generation of Report
* Lack of Efficiency and Man Power.
* High Data Redundancy.
* Data Inconsistency.

##### **Difficulty in Maintenance of Records:**

It is very difficult to maintain data record in the system as all the records are entered in the register or the respective record books. There are chances of the record books or files in which all the data is kept may be torn or wearied out or some other damages which results in the destroyed data. Also a problem occurs if the data file or register is misplaced somewhere else and is not getting at the time of actual requirement of the data stored. It is also difficult to maintain old files and registers which have data of past years.

##### **Time Consuming:**

It is very time consuming process to write each and every entry in the books database register. Also it takes a lot of time if all the entries are repeated. In the system processes such as making different type of reports, tedious calculations are examples of time consuming process in the system.

##### **Editing of Data:**

Manual written data cannot be changed or edited once written. If there is a mistake and if we try to rub and write it again it makes the register very dirty and untidy, which creates a bad impression of the business. If data is entered incorrect whole document gets incorrect while editing wrongly entered data cannot easily solve errors.

If one had done some wrong entry then to edit the data one has to go through lots of records and, again and again editing the record makes it difficult to read.

##### **Data Insecurity:**

As the data is stored in files or registers, it is not in a secure place. As the storage media here are files and books or registers there are chances of getting these storage media lost, torn, or it may go in the hand of wrong person, which can destroy the database, or it can also be destroyed accidentally. Also in the system data should be shown to the person according to his position in the Book shop, everybody should not be allowed to use all the data.

If the data goes in the hand of wrong person then he/she may take the advantage of the data and the Book shop may lose its market value. So security is the major aspects of the Book Shop System.

##### 

##### **Report Generation:**

After selling all the books in the stock, the payments collection report is prepared and on the basis of which the new stock and the decision to increase the quantity and category of books is carried out.

##### **High Data Redundancy:**

As mentioned in the current system, due to maintenance of so many registers there is a high redundancy of data i.e. same data is recorded repeatedly.

##### **Data Inconsistency:**

Here as mentioned in the above step the same information is written in more than one place that creates the problem, when there is a change or deletion in the recorded data.

For example if we would like to change the name of the Book due to some mistake earlier or due to some other circumstances, first we will have to identify the places where the name are recorded and then are updated this leads to a very time consuming process and if this is not done properly causes high data inconsistency in the later stages.

###### **COMPUTERIZED SYSTEM**

###### 

The manual system of Hostel is to be computerized in order to overcome the problems, which affects the existing manual system. Computerizing the existing system with the help of some programming language, database package ease the work of the system up to a great extent.

Generally, there has been a measure to work on any job or task for a specific purpose. Nobody works without specific detailed information about the particular task he is performing. Thus, any transaction can be performed either Books Stock payments or payments collection after selling books or check in and check out. In the computerized system, the first screen of the system would be a welcome message and a menu to login as admin or customer or book shop staff. These menus contain the options of either collecting the information of any book or the number of books available in Book shop, etc.

Any transaction can be performed in Book Shop Management System i.e. routine processes. After the transaction is completed the staff can log off from the system by simply quitting from the system. The main objective of the proposed system is to help the staff and customer. The system can be handy to the customer in the following reasons:

* To provide quick and efficient means for gathering the book information along with their ISBN number, cost of books and availability of books etc.
* To atomize the work such as gathering book information, payment information etc.

**FEATURES OF PROPOSED SYSTEM**

The “Book Shop Management System” is developed to overcome the most of the problems occurring in the manual system by computerizing the existing system. The features of the newly proposed computerized system are described in brief as below:

After computerizing the system, the workers in the shop or the customer can finish their work in least amount of time and efforts. The computerized system has many gains and efforts, which the manual system can’t give, in any type of situations.

In any manual system if we take, the main problem arising is to maintain the number of records and finding a particular record.

Computerized systems are most helpful in dealing with areas where database comes into the existence. A computer can hold large amount of data in its storage devices and it can operate at very high speed. The customer can search the entire information in the computer and can be able to perform any type of task which when done manually is tedious and time consuming. The new system will have the facility to sort data according to any specific type on the basis of what the staff wants in any order. Also with the help of computerized system if the customer wants to access the stock of the available books and its prices and go through the offers available at that moment, the customer can automatically get the desired data of the desired book or price or details of books, etc. in a fraction of second which is again time saving and very quick.

Some of the features of the proposed system are given below:

* Maintaining the Database for the stock of books, annual sell record and daily basis record, etc.
* Removal of Data Redundancy.
* Data Consistency.
* Ensure data security.

The main features of Book Shop Management System are given below:

1. As computer can store large amount of data it is very useful to store information of such a large database. Any information regarding total number of books in the inventory of book shop. Also information regarding payment of each and every book and the data of book bought by any of the customer.
2. Inconsistency caused due to changes in the database is removed as a separate database can be maintained for total number of books in the stock. So the database is directly updated in response to the any change taking place and we don’t have to go through all the forms once again.

**Conclusion**

Our project Book Shop Management is the computerize application to automate all kinds of activity in the book shop. The main aim of this is to manage the books in the book store. It can also manage the order processing, stock management and accounts management. This project is very helpful for maintain the records of sales, purchase and staff records. This project Book Shop Management System which has the type system which provides the facility to the customers of the shop to purchase the books from the shop without any problem. This system keeps all the record of books, receipts, sales and stocks.

In this project there will be some restrictions on the information for customer and staff as well. The staff will not have the access to update the stock as updating the stock can performed by the admin and the customer can only check the stock. When customer enters the shop he/she will first check the stock, that the book which he requires is available or not. If the book is available then the customer will go and take that book with the help of ISBN number of the book. The customer will then carry all the books to our staff which he/she wish to buy. The staff will issue all that books to the customer and at last he/she will pay the bill and get the slip.

The main purpose of the project is to have an automated system for customer dealings as well as for the stock. This automated system will result as an accurate system without any errors. The customer will also be provided the best services as regarding to stock and customer representative.

When the bookshop issues an item to a customer, all the stages of the transaction procedure will be facilitated by the system & it will be more accurate.

**Functionalities Used:**

**int main():**

The functionality of int main function is to execute the whole code without it any of the function cannot execute. In this function the front of the bookshop management system is printed and a function is used to set the font size of overall program. The code for int main function is given below:

int main()

{

system("color 0b"); //text color f //Front Method Print Details of Developers

fontsize(25,25);

header();

cout << "\n\t\n\t\t BOOK SHOP MANAGEMENT SYSTEM ";

cout << "\n\n\t\t GROUP MEMBERS ";

cout << "\n\n\t\t Ali Fawad 02-134191-094";

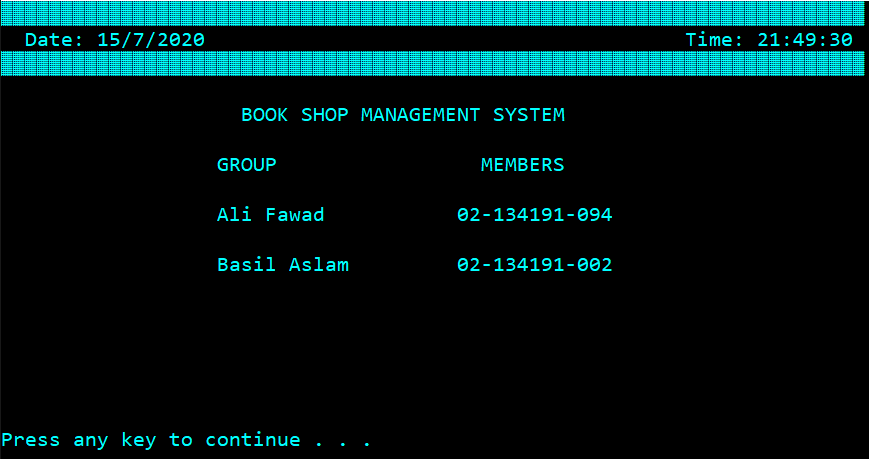
cout << "\n\n\t\t Basil Aslam 02-134191-002\n\n\n\n\n\n\n";

system ("pause");

login\_option();

}

The front end of the given code:



**login\_option:**

The void login\_option function is used for the login purpose of the book shop portal. This function shows three options, out of which one is to be selected. The options are administrator, staff and customer. Each option owns a username and password except the customer option, only that username and password will enable the operations according to the designation of the user. Each option doesn’t carries the same functionality. Administrator option owns the functionality to control each and every process, while the staff option owns some limited functionality as per their work. The code for void login\_option function is:

void login\_option()

{

system("cls");

print\_select\_one();

cout<<"\n\n\t1: Administrator\n";

cout<<"\n\n\t2: Staff \n\n";

cout<<"\n\n\t3: Customer \n\n";

int choice;

cout<<"\n\tEnter Your Choice : ";

cin>>choice;

switch(choice)

{

case 1:

{

admin\_staff\_login();

break;

}

case 2:

{

admin\_staff\_login();

break;

}

case 3:

{

costumer();

break;

}

default:

{

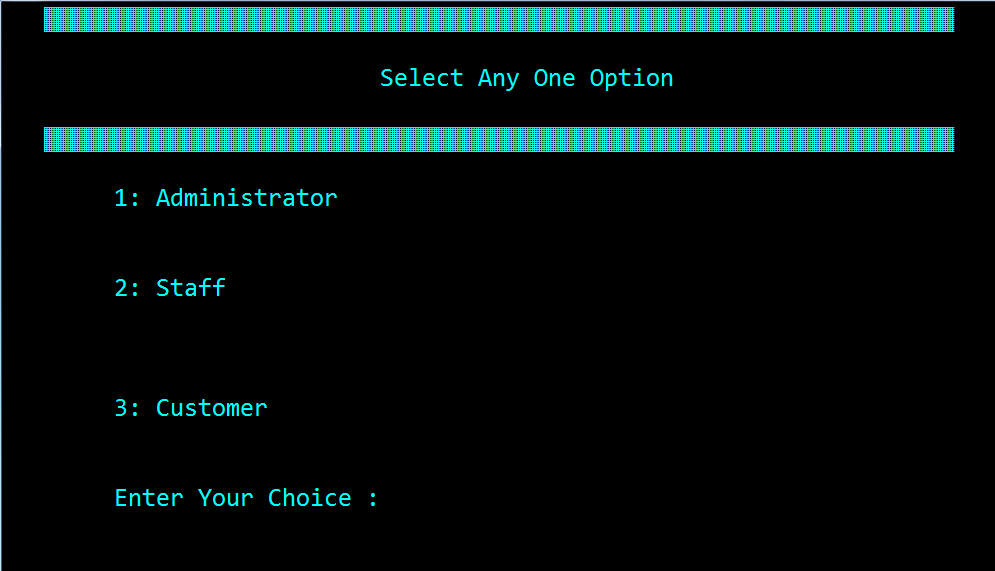
print\_select\_valid();

login\_option();

}}

}

The front end of the void login\_option function:



**admin\_staff\_login ():**

The functionality of admin\_staff\_login () function is to read and compare the admin and staff passwords store for the login of staff or admin portal. The code of admin\_staff\_login () function is:

void admin\_staff\_login()

{

system("cls");

header();

cout<<"\n\n\t\t\t\*\*\*\* LOGIN SCREEN \*\*\*\*\n\n";

string password="";

string username="";

char un,ps;

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

un=getch();

while(un!=13) // ASCII OF 'ENTER' KEY

{

cout<<un;

username+=un;

un=getch();

}

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

ps=getch();

while(ps!=13) // ASCII OF 'ENTER' KEY

{

cout<<'\*';

password+=ps;

ps=getch();

} if(username=="admin"&&password=="admin"||username=="staff"&&password=="staff")

{

cout<<"\n\n\t\t\t \xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\n";

cout<<"\t\t\t Access Approved \n";

cout<<"\t\t\t \xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\n";

Sleep(1000);

getch();

system("cls");

if(username=="admin")

{

admin();

}

else if(username=="staff")

{

staff();

}

}

else

{

gotoxy(10,10);

cout<<"\n\n\t\t\t \xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\n";

cout<<"\t\t\t Access Denied \n";

cout<<"\t\t\t \xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\n";

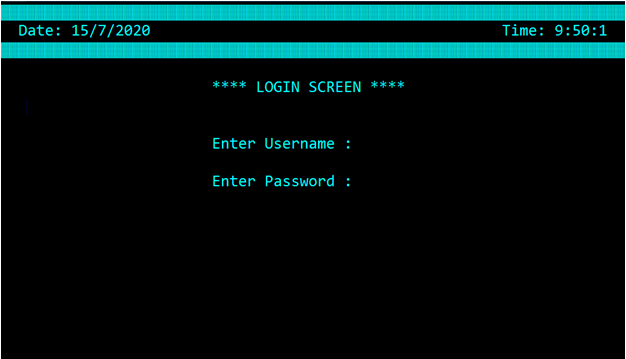
Sleep(1000);

getch();

login\_option();

}}

The front end of admin\_staff\_login function is:



**Admin ():**

The functionality of admin () function is to provide the operations which administrator can perform in the shop, after he logins in the administrator portal. The code for admin () function is:

void admin()

{

system("cls");

print\_select\_one();

cout<<"\n\n\t1 : ADD NEW BOOK \n";

cout<<"\t2 : EDIT BOOK \n";

cout<<"\t3 : DELETE BOOK \n";

cout<<"\t4 : VIEW STOCK \n";

cout<<"\t5 : SEARCH BOOK \n";

cout<<"\t6 : ADD STAFF DETAILS \n";

cout<<"\t7 : VIEW STAFF DETAILS \n";

cout<<"\t8 : GENERATE BILL \n";

cout<<"\t9 : VIEW ALL BILLS \n";

cout<<"\t10 : LOG OUT \n";

cout<<"\t11 : SHUT DOWN \n\n";

int choice;

cout<<"\tEnter your choice : ";

cin>>choice;

switch(choice)

{

case 1:

{

add\_book();

admin();

}

case 2:

{

edit\_book();

admin();

}

case 3:

{

delete\_book();

admin();

}

case 4:

{

view\_stock();

admin();

}

case 5:

{

search\_book();

admin();

}

case 6:

{

Add\_Staff\_Details();

admin();

}

case 7:

{

view\_Staff\_Details();

admin();

}

case 8:

{

generate\_bill();

admin();

}

case 9:

{

All\_bill() ;

admin();

}

case 10:

{

login\_option();

break;

}

case 11:

{

system("cls");

break;

}

default:

{

print\_select\_valid();

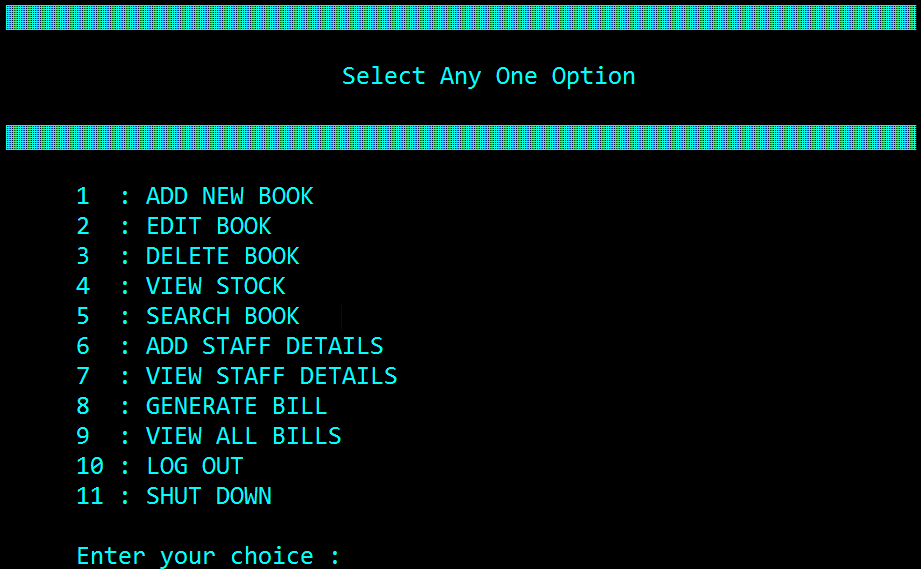
admin();

}

}

}

The front end of the admin () function is:



**Staff ():**

The functionality of the staff () function is to provide the operations that a staff can perform within the prescribed limits, after logging in its portal with the username and password. The code of the function staff () is:

void staff ()

{

system("cls");

print\_select\_one();

cout<<"\n\n\t1 : ADD NEW BOOK \n";

cout<<"\t2 : EDIT BOOK \n";

cout<<"\t3 : DELETE BOOK \n";

cout<<"\t4 : VIEW STOCK \n";

cout<<"\t5 : SEARCH BOOK \n";

cout<<"\t6 : GENERATE BILL \n";

cout<<"\t7 : LOG OUT \n";

cout<<"\t8 : SHUT DOWN \n\n";

int choice;

cout<<"\tEnter your choice : ";

cin>>choice;

switch(choice)

{

case 1:

{

add\_book();

staff();

}

case 2:

{

edit\_book();

staff();

}

case 3:

{

delete\_book();

staff();

}

case 4:

{

view\_stock();

staff();

}

case 5:

{

search\_book();

staff();

}

case 6:

{

generate\_bill();

staff();

}

case 7:

{

login\_option();

}

case 8:

{

system("cls");

break;

}

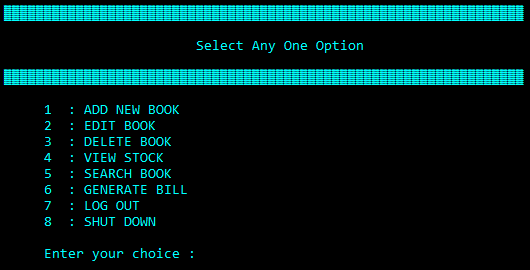
default:

{

print\_select\_valid();

staff(); }}}

The front end of staff () function is:



**Costumer ():**

The functionality of costumer () function is to provide the functions that a customer can perform after selecting the option from login menu as the customer enters in the shop. After selecting the customer option from the login menu the customer can easily access the customer option with any username or password. The code of costumer () function is:

void costumer()

{

system("cls");

print\_select\_one();

cout<<"\n\n\t1 : SEARCH BOOK "<<endl;

cout<<"\t2 : VIEW ALL BOOK "<<endl;

cout<<"\t3 : MAIN MENU "<<endl;

int choice;

cout<<"\tEnter your choice : ";

cin>>choice;

switch(choice)

{

case 1:

{

search\_book();

costumer();

}

case 2:

{

view\_stock();

costumer();

}

case 3:

{

login\_option();

}

default:

{

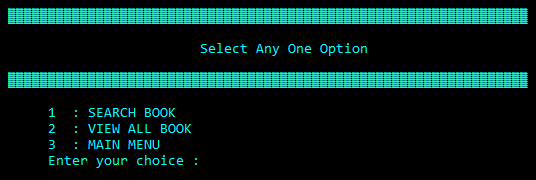
print\_select\_valid();

costumer();

} }

}

The front end of customer () function is:



**add\_book:**

The functionality of add\_book function is the addition of the books in the stock. This function carries the ISBN number, quantity, title, author and price of the book. The code of add\_book is given below:

void add\_book()

{

system("cls");

header();

ofstream fout;

ifstream fin;

node\* item;

item = new node;

string temp\_isbn;

bool x=true;

cout<<"\n\n\t\t\t\*\*\*\* ADD NEW BOOK IN STOCK \*\*\*\*\n\n";

cout << "\n\n\t\t\tEnter ISBN of Book : ";

cin >> item->isbn;

temp\_isbn=item->isbn;

fin.open("Book.txt",ios::in);

if(!fin)

{

fin.open("Book.txt",ios::in);

}

else

{

int quantity,price;

string title, author,isbn;

while(!fin.eof())

{

fin>>isbn >> quantity >> title >> author >> price;

if (temp\_isbn==isbn)

{

cout<<"\n\n\t\t\t \xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2";

cout<<"\n \t\t\t BOOK ALREADY EXIST \n";

cout<<"\t\t\t \xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2";

x=false;

getch();

break;

}

}

}

if(x==true)

{

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

cin >> item->quantity;

cout << "\t\t\tEnter Title of Book : ";

cin.ignore();

getline(cin, item->title);

cout << "\t\t\tEnter Book Author's Name: ";

getline(cin, item->author);

cout << "\t\t\tEnter Price of Book : ";

cin >> item->price;

item->next = NULL;

fout.open("Book.txt",ios::app);

fout<<item->isbn<<" "<<item->quantity<<" "<<item->title<<" "<<item->author<<" "<<item->price<<"\n";

fout.close();

if (head == NULL)

{

head = item;

}

else

{

node \*temp=head;

while (temp->next != NULL)

{

temp=temp->next;

}

temp->next = item;

}

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

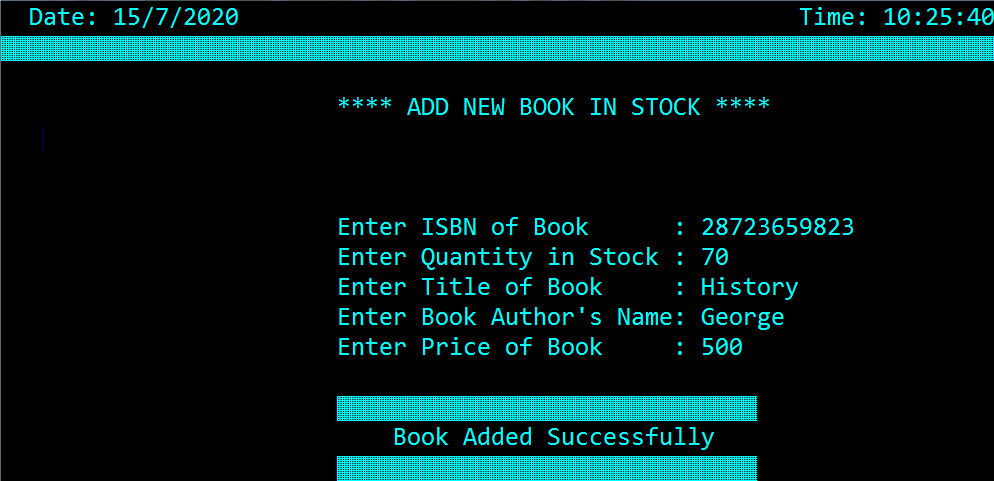
for(int i=0;i<30;i++){ cout<<"\xB2";}

cout<<"\n \t\t\t Book Added Successfully \n\t\t\t";

for(int i=0;i<30;i++){ cout<<"\xB2";}

getch();}}

Front end of the add\_book function:



**edit\_book:**

The functionality of void edit\_book () function is to modify any of the detail of the book present in the stock. We can modify the ISBN number, quantity, title, author and even price also. The code of the void edit\_book () function is:

void edit\_book()

{

system("cls");

header();

cout<<"\n\n\t\t\t\*\*\*\* EDIT BOOK DETAILS \*\*\*\*\n\n";

int isbn,quantity,price;

string title, author;

ofstream fout,fout\_up;

ifstream fin;

fin.open("Book.txt",ios::in);

node\* data=new node;

bool x=0,y=true;

if(!fin)

{

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

for(int i=0;i<30;i++){ cout<<"\xB2";}

cout<<"\n \t\t\t Error With Opening File \n\t\t\t";

for(int i=0;i<30;i++){ cout<<"\xB2";}

getch();

}

else

{

x=search\_book() ;

if(x==1)

{

int user\_isbn;

cout<<"\n\n\t\t\tPlease Again Enter ISBN : ";

cin>>user\_isbn;

fout.open("NewBook.txt",ios::out);

fin>>isbn >> quantity >> title >> author >> price;

while(!fin.eof())

{

if(user\_isbn!=isbn)

{

fout<<isbn<<" "<<quantity<<" "<<title<<" "<<author<<" "<<price<<"\n";

fin>>isbn >> quantity >> title >> author >> price;

}

else

{

cout<<"\n\t\t\tEnter ISBN To Update : ";

cin>>data->isbn;

int quantity,price;

string title, author,isbn;

while(!fin.eof())

{

fin>>isbn >> quantity >> title >> author >> price;

if (data->isbn==isbn)

{

cout<<"\n\n\t\t\t \xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2";

cout<<"\n \t\t\t BOOK ALREADY EXIST \n";

cout<<"\t\t\t \xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2";

x=false;

getch();

break;

}

}

if (y==true)

{

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

cin>>data->quantity;

cout<<"\t\t\tEnter Title To Update : ";

cin>>data->title;

cout<<"\t\t\tEnter Author To Update : ";

cin>>data->author;

cout<<"\t\t\tEnter Price To Update : ";

cin>>data->price;

fout<<data->isbn<<" "<<data->quantity<<" "<<data->title<<" "<<data->author<<" "<<data->price<<"\n";

fin>>isbn>>quantity>>title>>author>>price;

edited\_successfully();

}

}

fin.close();

fout.close();

remove("Book.txt");

rename("NewBook.txt","Book.txt");

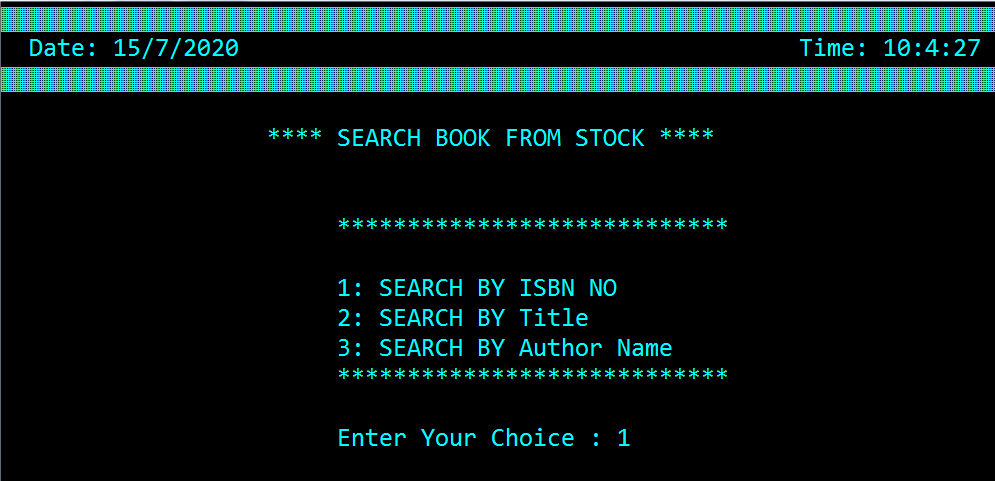
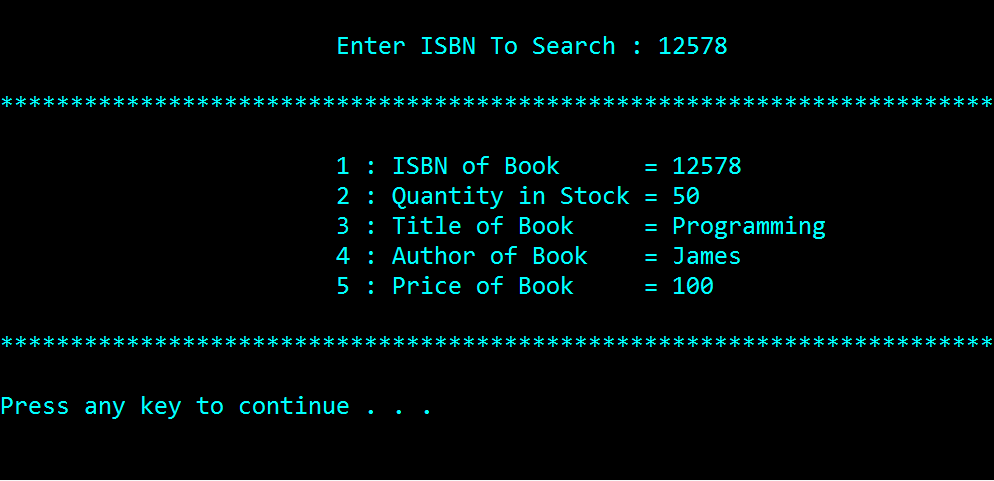
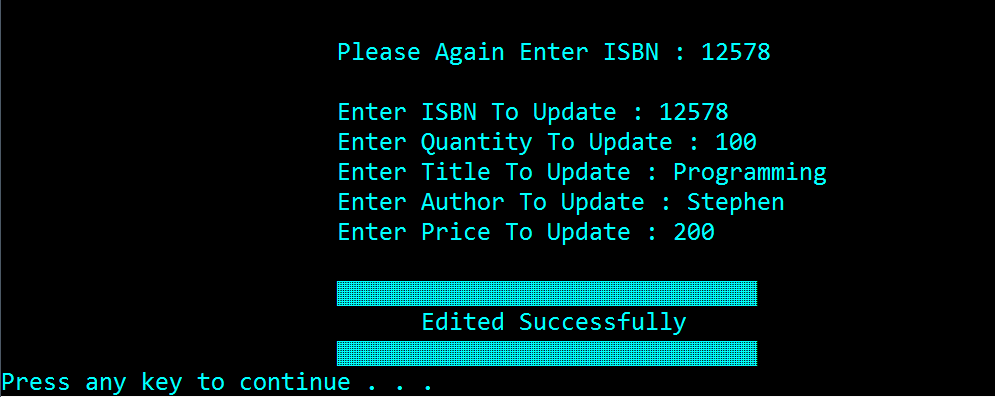
}

}

}

}

The front end of the void edit\_book function:

**delete\_book ():**

The functionality of delete\_book () function is to delete any of the book record stored in the stock. The code for delete\_book () is:

**void delete\_book()**

**{**

system("cls");

header();

ofstream fout;

ifstream fin;

fin.open("Book.txt",ios::in);

int isbn,quantity,price;

string title, author;

bool x=0;

if(!fin)

{

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

for(int i=0;i<30;i++){ cout<<"\xB2";}

cout<<"\n \t\t\t Error With Opening File \n\t\t\t";

for(int i=0;i<30;i++){ cout<<"\xB2";}

getch();

}

else

{x=search\_book() ;

if(x==1)

{

int user\_isbn;

cout<<"\n\n\t\t\tPlease Again Enter ISBN : ";

cin>>user\_isbn;

fout.open("NewBook.txt",ios::out);

fin>>isbn>>quantity>>title>>author>>price;

while(!fin.eof())

{

if(user\_isbn!=isbn)

{

fout<<isbn<<" "<<quantity<<" "<<title<<" "<<author<<" "<<"\n";

}fin>>isbn>>quantity>>title>>author>>price;

}

fin.close();

fout.close();

remove("Book.txt");

rename("NewBook.txt","Book.txt");

}

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

for(int i=0;i<30;i++){ cout<<"\xB2";}

cout<<"\n \t\t\t Deleted Successfully \n\t\t\t";

for(int i=0;i<30;i++){ cout<<"\xB2";}

getch();}**}**

The front end of the delete\_book () function:



**view\_stock ():**

The functionality of the view\_stock () function is that, it shows all of the books present in the stock. The stock represents book with its ISBN number, author, quantity and price. The code for view\_stock () function is:

void view\_stock()

{

system("cls");

header();

cout<<"\n\n\t\t\t\*\*\*\* ALL BOOKs DETAIL \*\*\*\*\n\n";

ifstream fin;

fin.open("Book.txt",ios::in);

if(!fin)

{

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

for(int i=0;i<30;i++){ cout<<"\xB2";}

cout<<"\n \t\t\t Error With Opening File \n\t\t\t";

for(int i=0;i<30;i++){ cout<<"\xB2";}

getch();

}

else

{

int isbn,quantity,price;

string title, author;

while (fin >> isbn >> quantity >> title >> author >> price)

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

cout <<"\t\t\tISBN of Book : " <<isbn<<endl;

cout <<"\t\t\tQuantity in Stock : " <<quantity<<endl;

cout <<"\t\t\tTitle of Book : " <<title<<endl;

cout <<"\t\t\tAuthor of Book : " <<author<<endl;

cout <<"\t\t\tPrice of Book : " <<price<<endl;

}

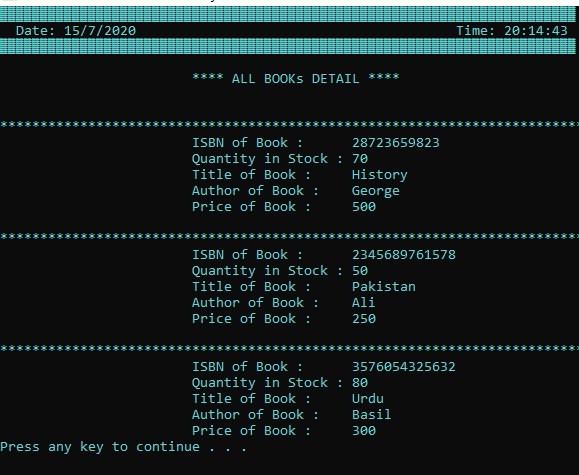
system("pause");

}

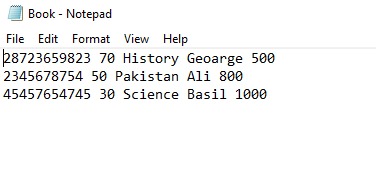
fin.close();

}

The front end of the view\_stock function:



The stock stored in txt file:



**search\_book ():**

The functionality of search\_book () is to search any of the book stored in the stock. It provides a facility to search the by ISBN number or by Author or by book title. The code for search\_book () function is:

bool search\_book()

{

system("cls");

header();

cout<<"\n\n\t\t \*\*\*\* SEARCH BOOK FROM STOCK \*\*\*\*\n\n";

int choice;

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

cout << "\n\n\t\t\t1: SEARCH BY ISBN NO ";

cout << "\n\t\t\t2: SEARCH BY Title ";

cout << "\n\t\t\t3: SEARCH BY Author Name ";

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

cout << "\n\n\t\t\tEnter Your Choice : ";

cin >> choice ;

node\* data = new node;

ifstream fin;

fin.open("Book.txt",ios::in);

if(!fin)

{

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

for(int i=0;i<30;i++){ cout<<"\xB2";}

cout<<"\n \t\t\t Error With Opening File \n\t\t\t";

for(int i=0;i<30;i++){ cout<<"\xB2";}

getch();

}

else

{

int isbn,quantity,price;

string title, author;

switch(choice)

{

case 1:

{

bool x=true;

int searched\_isbn;

cout<<"\n\n\t\t\tEnter ISBN To Search : ";

cin>>searched\_isbn;

while(!fin.eof())

{

fin>>isbn >> quantity >> title >> author >> price;

if (searched\_isbn==isbn )

{

data->isbn=isbn; data->quantity=quantity; data->title=title; data->author=author; data->price=price;

print\_searched(data);

x=false;

return true;

break;

}

}

if(x==true)

{

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

for(int i=0;i<30;i++){ cout<<"\xB2";}

cout<<"\n \t\t\t NO RECORD TO DISPLAY \n\t\t\t";

for(int i=0;i<30;i++){ cout<<"\xB2";}

getch();

search\_book();

}

break;

}

case 2:

{

bool x=true;

string searched\_title;

cout<<"\n\n\t\t\tEnter Title To Search : ";

cin.ignore();

getline(cin,searched\_title);

while(!fin.eof())

{

fin>>isbn >> quantity >> title >> author >> price;

if (searched\_title==title )

{

data->isbn=isbn; data->quantity=quantity; data->title=title; data->author=author; data->price=price;

print\_searched(data);

x=false;

return true;

break;

}

}

if(x==true)

{

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

for(int i=0;i<30;i++){ cout<<"\xB2";}

cout<<"\n \t\t\t NO RECORD TO DISPLAY \n\t\t\t";

for(int i=0;i<30;i++){ cout<<"\xB2";}

getch();

search\_book();

}

break;

}

case 3:

{

bool x=true;

string searched\_name;

cout<<"\n\n\t\t\tEnter Author Name To Search : ";

cin.ignore();

getline(cin, searched\_name);

while(!fin.eof())

{

fin>>isbn >> quantity >> title >> author >> price;

if (searched\_name==author)

{

data->isbn=isbn; data->quantity=quantity; data->title=title; data->author=author; data->price=price;

print\_searched(data);

x=false;

return true;

break;

}

}

if(x==true)

{

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

for(int i=0;i<30;i++){ cout<<"\xB2";}

cout<<"\n \t\t\t NO RECORD TO DISPLAY \n\t\t\t";

for(int i=0;i<30;i++){ cout<<"\xB2";}

getch();

search\_book();

}

break;

}

default :

{

print\_select\_valid();

search\_book();

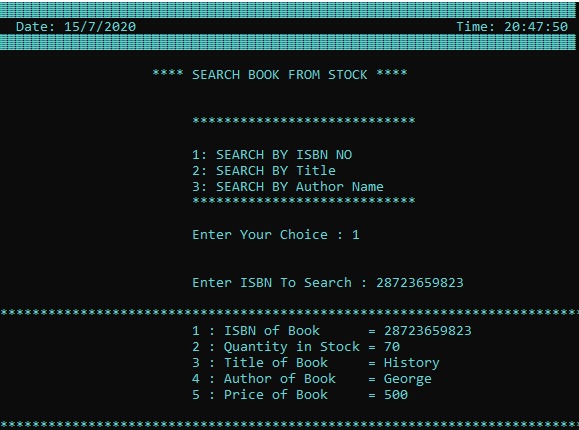
}

}

}

}

The front of the given code is:



**Add\_Staff\_Details ():**

The functionality of Add\_Staff\_Details () is to add the details of the staff of book shop. The function enables to add the staff’s name, CNIC number, age and date of birth as staff details. The code for Add\_Staff\_Details () is:

void Add\_Staff\_Details()

{

system("cls");

header();

cout<<"\n\n\t\t\t\*\*\*\* ADD STAFF DETAILS \*\*\*\*\n\n";

cout <<"\N\t\t\t\nEnter Name : ";

cin.ignore();

getline(cin, Staff\_name);

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

getline(cin,Staff\_CNIC );

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

cin >> Staff\_age;

cout << "\t\t\tEnter Date Of Birth : ";

cin.ignore();

getline(cin,Staff\_DOB);

ofstream fout;

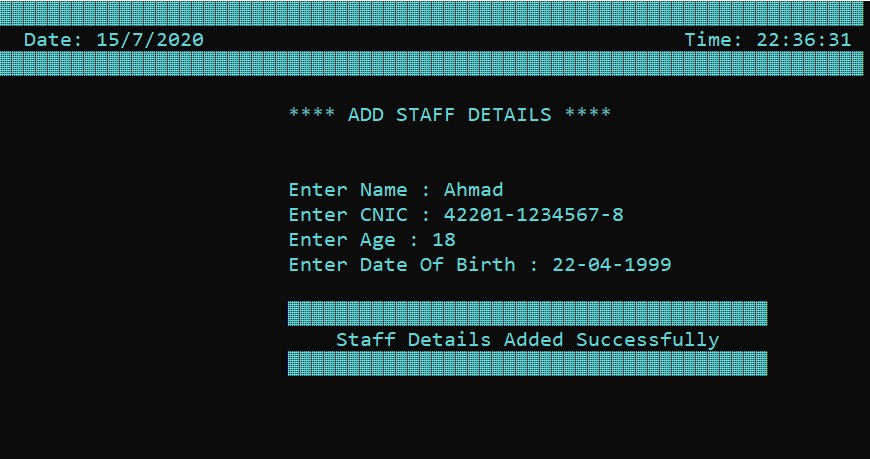
fout.open("Staff\_Details.txt");

fout<<Staff\_name<<" "<<Staff\_CNIC<<" "<<Staff\_age<<" "<<Staff\_DOB<<"\n";

fout.close();

}

The front end of the Add\_Staff\_Details is:



**view\_Staff\_Details ():**

The functionality of view\_Staff\_Details () function is to show the details of the staff of book shop. The details includes the staff’s name, CNIC number, age and date of birth. The code for view\_Staff\_Details is:

void view\_Staff\_Details()

{

system("cls");

header();

cout<<"\n\n\t\t\t\*\*\*\* STAFF DETAIL \*\*\*\*\n\n";

ifstream fin;

fin.open("Staff\_Details.txt");

if(!fin)

{

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

for(int i=0;i<30;i++){ cout<<"\xB2";}

cout<<"\n \t\t\t Error With Opening File \n\t\t\t";

for(int i=0;i<30;i++){ cout<<"\xB2";}

getch();

}

else

{

while (fin >> Staff\_name >> Staff\_CNIC >> Staff\_DOB >> Staff\_age)

{

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

cout <<"\t\t\t1. Name : " <<Staff\_name<<endl;

cout <<"\t\t\t2. CNIC : " <<Staff\_CNIC<<endl;

cout <<"\t\t\t3. Date Of Birth : " <<Staff\_DOB<<endl;

cout <<"\t\t\t4. Age : " <<Staff\_age<<endl;

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

}

}

fin.close();

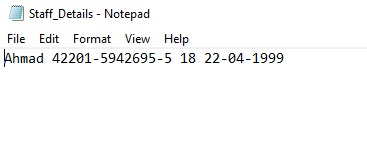
system("pause");

}

The front end of the view\_Staff\_Details is:

****

The staff details stored in file:



**generate\_bill ():**

The functionality of the generate\_bill function is to generate the bill of the customers. The bill includes the book ISBN number, title, author, quantity and price according to the quantity of the books selected by customer. At last it prints the total amount of all the books selected by the customer. The code for generate\_bill is:

void generate\_bill()

{

system("cls");

header();

cout<<"\n\n\t\t\t\*\* GENERATE BILL \*\*\n\n";

int c;

ifstream fin;

ofstream fout,fout1,fout\_up;

fout.open("print.txt",ios::app);

fout1.open("All\_Bill.txt",ios::app);

time\_t t = time(NULL);

tm\* tPtr = localtime(&t);

int day=(tPtr->tm\_mday),month= (tPtr->tm\_mon)+1 , year=(tPtr->tm\_year)+1900;

int hour=(tPtr->tm\_hour), minute=(tPtr->tm\_min) , sec= (tPtr->tm\_sec);

if (!fout||!fout1)

{

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

for (int i = 0; i < 30; i++) { cout << "\xB2"; }

cout << "\n \t\t\t Error With Opening File \n\t\t\t";

for (int i = 0; i < 30; i++) { cout << "\xB2"; }

getch();

}

do {

int cust\_quantity;

string cust\_isbn;

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

cin >> cust\_isbn;

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

cin >> cust\_quantity;

fin.open("Book.txt", ios::in);

if (!fin)

{

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

for (int i = 0; i < 30; i++) { cout << "\xB2"; }

cout << "\n \t\t\t Error With Opening File \n\t\t\t";

for (int i = 0; i < 30; i++) { cout << "\xB2"; }

getch();

}

else

{

int quantity, price;

string title, author,isbn;

while (!fin.eof())

{

fin >> isbn >> quantity >> title >> author >> price;

if (cust\_isbn == isbn)

{

fout <<"\n\tBook Shop Management System"<<"\nBook ISBN : "<< cust\_isbn << "\nBook Title: " << title << "\nAuthor: " << author << "\nBook Quantity: "<< cust\_quantity<< "\nBook Price: "

<< price \* cust\_quantity <<"\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*";

fout1 << "\nBook ISBN : "<< cust\_isbn << "\nBook Title" << title << "\nAuthor" << author << "\nBook Quantity"<< cust\_quantity<< "\nBook Price : "

<< price \* cust\_quantity <<"\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*";

break;

}

}

}

fout\_up.open("NewBook.txt",ios::out);

int quantity, price;

string title, author,isbn;

fin>>isbn >> quantity >> title >> author >> price;

while(!fin.eof())

{

if(cust\_isbn!=isbn)

{

fout<<isbn<<" "<<quantity<<" "<<title<<" "<<author<<" "<<price<<"\n";

fin>>isbn >> quantity >> title >> author >> price;

}

else

{

int temp = quantity;

quantity=quantity-cust\_quantity;

if(quantity<0)

{

cout<<"\n\t\tWe Don't Have Enough Quantity\n";

cout<<"\t\tNumber of Book in Stock : "<<temp;

}

fout<<isbn<<" "<<quantity<<" "<<title<<" "<<author<<" "<<price<<"\n";

fin>>isbn>>quantity>>title>>author>>price;

}

}

fout\_up.close();

remove("Book.txt");

rename("NewBook.txt","Book.txt");

again:

cout << "\n";

cout << "\n\t1 : Do you wish for another product ";

cout << "\n\t2 : Exit ";

cout << "\n\t Enter Your Choice : ";

cin >> c;

if(c>2)

{

print\_select\_valid();

goto again;

}

} while (c==1);

fout << "\nDate : "<< day << "/" << month << "/" << year << "\nTime : " << hour << ":" << minute << ":" << sec << "\n" ;

fout1 << "\nDate : "<< day << "/" << month << "/" << year << "\nTime : " << hour << ":" << minute << ":" << sec << "\n" ;

fin.close();

fout.close();

fout1.close();

fin.open("print.txt");

int p\_quantity,p\_price,p\_day, p\_month, p\_year, p\_hour , p\_minute,p\_sec;

string p\_title, p\_author, p\_isbn;

while (!fin.eof())

{

fin >> p\_isbn >> p\_title >> p\_author >> p\_quantity >> p\_price >> p\_day >> p\_month >> p\_year >> p\_hour >> p\_minute >> p\_sec;

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

cout <<"\t\t\tISBN of Book : " <<p\_isbn<<endl;

cout <<"\t\t\tQuantity Buyed : " <<p\_quantity<<endl;

cout <<"\t\t\tTitle of Book : " <<p\_title<<endl;

cout <<"\t\t\tAuthor of Book : " <<p\_author<<endl;

cout <<"\t\t\tTotal Price of Bill : " <<p\_price<<endl;

cout <<"\t\t\tTime : " << p\_hour <<":"<<p\_minute << ":" << p\_sec <<endl;

cout <<"\t\t\tDate : " << p\_day << "/" <<p\_month << "/" << p\_year <<endl;

}

system("pause");

system("notepad /p print.txt");

cout<<"\n\t\t\tPrinting Slip....";

if (remove("print.txt") != 0)

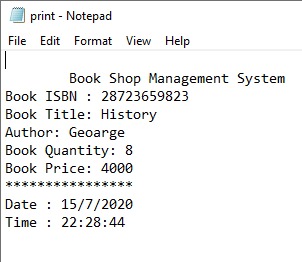
perror("File deletion failed");

else

cout << "File deleted successfully";

}

The bill generated of the generate\_bill function in a txt file is:



**All\_bill ():**

The functionality of All\_bill () function is to show the overall record of the customers that bought books from our shop. The code for All\_bill () function is:

void All\_bill()

{

ifstream fin;

fin.open("All\_Bill.txt", ios::in);

if (!fin)

{

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

for (int i = 0; i < 30; i++) { cout << "\xB2"; }

cout << "\n \t\t\t Error With Opening File \n\t\t\t";

for (int i = 0; i < 30; i++) { cout << "\xB2"; }

getch();

}

else

{

int isbn,quantity,price,day, month, year, hour , minute,sec;

string title, author;

while (fin >> isbn >> title >> author >> quantity >> price >> day >> month >> year >> hour >> minute >> sec )

{

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

cout <<"\t\t\tISBN of Book : " <<isbn<<endl;

cout <<"\t\t\tQuantity Buyed : " <<quantity<<endl;

cout <<"\t\t\tTitle of Book : " <<title<<endl;

cout <<"\t\t\tAuthor of Book : " <<author<<endl;

cout <<"\t\t\tTotal Price of Bill : " <<price<<endl;

cout <<"\t\t\tTime : " << hour <<":"<< minute << ":" << sec <<endl;

cout <<"\t\t\tDate : " << day << "/" << month << "/" << year <<endl;

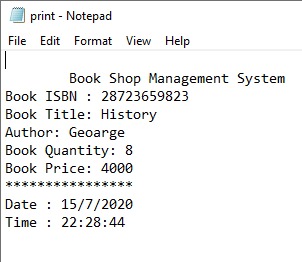
}

system("pause");

}

fin.close();

}



**Additional Functionalities**

Below are given sum additional functionalities which are used to minimize the complexity of the code.

**print\_select\_one ():**

The functionality of print\_select\_one is to represent the heading of “Select Any One Option” in the code. This function is further used in other functions. The code of the function is:

void print\_select\_one()

{

cout<<"\n ";

for(int i=0;i<65;i++){ cout<<"\xB2";}

cout<<"\n\n\t\t\t Select Any One Option \n\n ";

for(int i=0;i<65;i++){ cout<<"\xB2";}

}

**print\_select\_valid ():**

The functionality of print\_select\_valid is to print the command of “Please Select Valid Option” in the code. This function is further used in other functions. The code of the function is:

void print\_select\_valid()

{

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

for(int i=0;i<30;i++){ cout<<"\xB2";}

cout<<"\n \t\t\t Please Select Valid Option \n\t\t\t";

for(int i=0;i<30;i++){ cout<<"\xB2";}

system("pause");

}

**Time ():**

The functionality of the time () function is to show the current time where it is needed in the code. The code of the function is:

void time()

{

time\_t t = time(NULL);

tm\* tPtr = localtime(&t);

cout << "Date: " <<(tPtr->tm\_mday)<<"/"<< (tPtr->tm\_mon)+1 <<"/"<< (tPtr->tm\_year)+1900;

cout << "\t\t\t\t\t Time: " << (tPtr->tm\_hour)<<":"<< (tPtr->tm\_min)<<":"<< (tPtr->tm\_sec) << endl;

}

**Header ():**

The functionality of the header () function is to provide the header in different functions of the code. The code of the function is:

void header()

{

for(int i=0;i<72;i++){ cout<<"\xB2";}

cout<<"\n ";

time();

for(int i=0;i<72;i++){ cout<<"\xB2";}

}

**print\_searched**:

The functionality of print\_searched function is to print the details of the book in each and every function where it is needed. The code of the function is:

void print\_searched (node \*data)

{

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

cout <<"\t\t\t1 : ISBN of Book = " <<data->isbn<<endl;

cout <<"\t\t\t2 : Quantity in Stock = " <<data->quantity<<endl;

cout <<"\t\t\t3 : Title of Book = " <<data->title<<endl;

cout <<"\t\t\t4 : Author of Book = " <<data->author<<endl;

cout <<"\t\t\t5 : Price of Book = " <<data->price<<endl;

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

system("pause");

}

**edited\_successfully ():**

The functionality of edited\_successfully () function is to print the command of “Edited Successfully” wherever it is needed in the code. The code of the function is:

void edited\_successfully ()

{

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

for(int i=0;i<30;i++){ cout<<"\xB2";}

cout<<"\n \t\t\t Edited Successfully \n\t\t\t";

for(int i=0;i<30;i++){ cout<<"\xB2";}

cout<<"\n";

system("pause");

}

**Fontsize ():**

The functionality of fontsize () is to alter the font size of throughout the program running in the console. The code of the function is:

void fontsize(int a, int b) {

PCONSOLE\_FONT\_INFOEX lpConsoleCurrentFontEx = new CONSOLE\_FONT\_INFOEX();

lpConsoleCurrentFontEx->cbSize = sizeof(CONSOLE\_FONT\_INFOEX);

GetCurrentConsoleFontEx(out, 0, lpConsoleCurrentFontEx);

lpConsoleCurrentFontEx->dwFontSize.X = a;

lpConsoleCurrentFontEx->dwFontSize.Y = b;

SetCurrentConsoleFontEx(out, 0, lpConsoleCurrentFontEx);

}