LIBRARY MANAGEMENT SYSTEM

CSE-2003

AYUSH SAXENA-19BCE0892
ayush.saxena2019@vitstudent.ac.in

ABSTRACT

This project aims to develop a computerized system to maintain all the records of the diurnal operations carried out in libraries. Our project is equipped with portals for both, the student and the librarian for an easier working. Introducing the concept of a student portal gives our project the edge it requires. Most library management systems are as they are only limited to the with librarian portal with various function like addition of books, deletion of books, searching a book, modification of book details and issuing books. Adding to all these features, a student can rate the books, provide reviews for the same, suggest books and search for books according to their rating and reviews.
Students are informed upon updation of the library under the recently added section.
Admin account allows the addition and deletion of student IDs. The current student login lands to Student main menu only if the student is existing member of class.
Existing records can be edited only by an authorized person. New members can only be registered by the library administrator or authorized person.
Overall this project of ours is being enveloped to help the students as well as staff of library to maintain the library in the best way possible and reduce the human efforts.

Keywords: Library management system, login portal, add books, delete books, modify book details, issue books, suggest books, provide review, view review, add user.

INTRODUCTION

In this age of budding IT sector and e-science, manual library systems are on the brink of being replaced with digital library systems. To cope up with the rapid development, dynamic and efficient alternatives are being opted for. The existing manual library system requires more time for processing and adding a new feature. Library Management system reduces manual activities and paperwork. With this computerized system the issues faced by the existing manual method of library management are automatically solved such as registration of users and difficulties in maintaining a proper record of books as well as time and energy wasted in searching a material all can be solved easily and efficiently.

Books and student maintenance modules are also included in this system which would monitor the library. With this electronic framework there will be no loss of book record or part record which by and large happens when a non-automated framework is utilized.

The main objective is to eliminate paper work, maintain a record of every transaction in a computerized way to avoid issues such as missing files, to provide request column for librarian for providing new books, and to provide a search column to search the availability of a book.

LMS will help increase efficiency and the systemise the working of the library by reducing man power and manual activities subsequently improve productivity.

LITERATURE REVIEW

A Library Management System (LMS) is a tool to overcome the manual labour and reduce the time required to search and issue books. Our project is capable of managing the work of a library. Book transactions including book searching, reviewing and rating a book, details and appearance of the book. These operations can be easily handled by this system. Our system is suitable libraries ranging from small to big libraries for colleges, schools, universities and other academic resource centres. However, our focus is LMS for colleges or universities.

LMS has consists of two portals:

1.Student

2.Librarian

Students portal allows students to search books by title, name/author, subject etc. It also allow students to find recently added material. For the librarian portal, staff can issue book, receive book, change book details, register new users and have access to altering the database backup/restore. Overall this system can be very helpful.

Overall this technique is being enveloped to help the students as well as staff of library to maintain the library in the best way possible and reduce the human efforts and can be terribly useful, and it will build things easier.

PROPOSED WORK

LIBRARIAN PORTAL:

The librarian portal is encrypted with a password allowing access only to the staff. The password of the librarian portal is “codechamp”. As discussed earlier the librarian portal has the following operations:

- Add books
- Delete books
- Search books
- Issue books
- View book list
- View suggested book list
- Edit book’s record
- Add a new student to library
- Change login
- Close application

Add books:
This function allows the librarian to add n number of books to the student login. Books can be added to the various categories like computer, electronics, electrical, civil, mechanical and

architecture. The function requires the following information to be given beforehand such as the unique book ID of the book, book name, author of the book, number of books to be added, price of each book and the rack where the book is to be placed.

Pseudocode

Begin

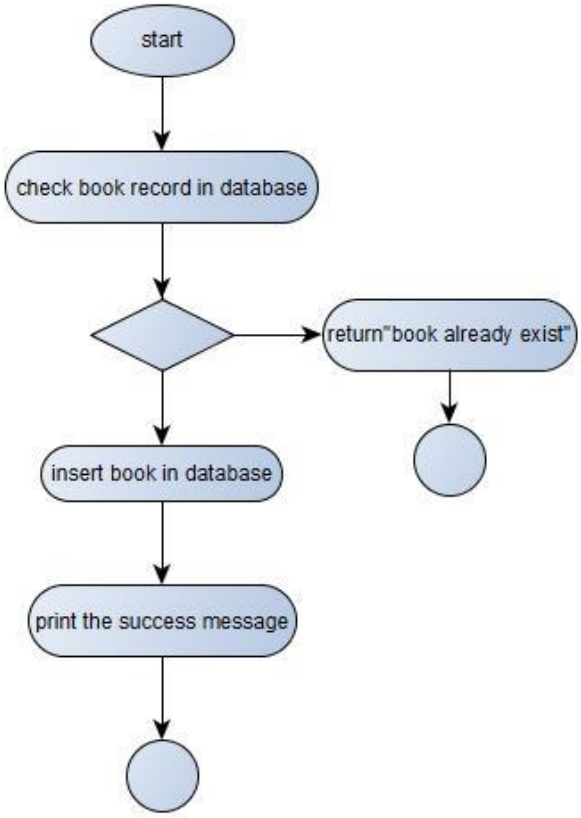
If pointer = null (list is overflow)

Pointer == info = d2

Pointer == next = head

Head = p

End



Delete books:

This function allows the librarian to delete a particular book or number of books from the library. Every book has unique book ID, upon entering this particular ID we can access the book to be deleted. Hence the book will be deleted from the database directly.

Pseudocode:

Begin

pointer = head

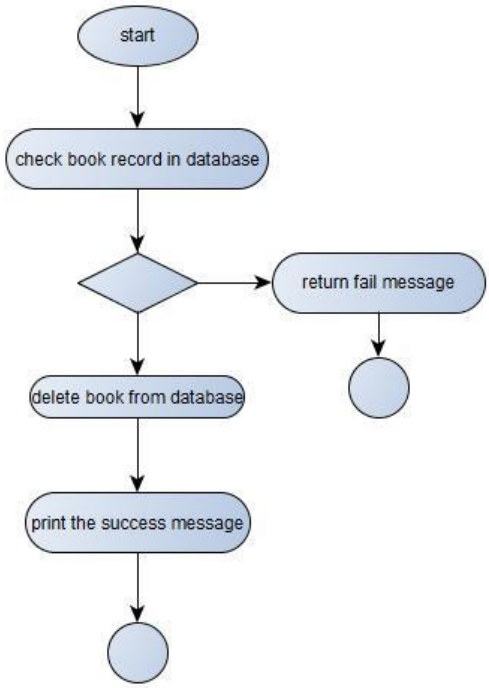
loop (pointer == next! = null)

loc = pointer

```
pointer = pointer == next
    end loop
loc == next = null
free (pointer)
End
```

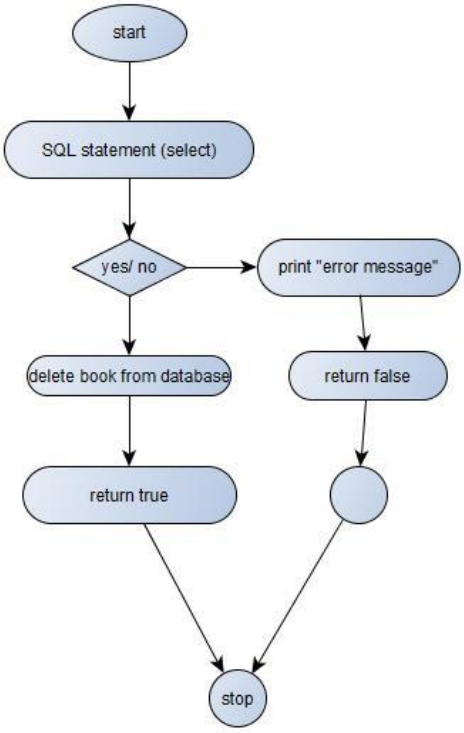
Search books:

This function allows the librarian to search for the book required. Searching can be done in two ways, using ID of the book or the name. Searching a book using its ID, finds the unique ID of the books and then displays the book. Searching a book using the name reads the complete characters of the book name and then if found it displays the details of the book.



Issue book:

This function is used to issue the book to the user, view the list of issued books, search the issued book or remove the issued book. For the user to issue a book, he's required to enter the unique book ID of the book, if the book is available on the rack, the user has the option to issue the book. The user is required to input the following details for him to issue the book. User must give his registration number and name. This function displays the issue date and when the book is to be returned, and the number of issued books under that particular account. 'View issued book' displays the all the books issued from the library in a tabular form. 'Search issued book' checks which is book is issued under which ID. 'Remove issued book' removes the book from the issued list of the user once if he returns.



View book list:
This function allows the librarian to view the number of books available in the library with the respective details such as the rack number, category, book name, author, and the price. It also displays the unissued books on the rack.

View books:
This function gives the names of the books that are persisting in the library, along its details. This function calls the data entered by the librarian, through add books function.

Pseudocode:

Begin
Visit the list where the data given by the librarian is stored.
Visit each node in the list.
If the nodes are empty
Display “There are No books available”
else
Display ”category; Id; name; author; quantity, rack number of the books available”

Edit books:
If the librarian wants to edit the book details he can use this function to rename/edit the author name, book id, book name, quantity of the books remaining in the library. The main benefit through this automated librarian system is it makes easier to maintain a perfect record by using this function.

Pseudocode:

Begin
Enter the book id to be edited.
Check whether

If valid: change/ edit the author, name, quantity, rack number or price and print
"The record is modified"
Else print "No Record Found"
End

Add a New Student:
This Function allows the librarian to add a new student to the library by creating new ID for the student and allowing him access to avail the features of LMS.

Pseudocode:
Start
Enter the details of the student to take as input.
print" student login created successfully"
End

Change Login
This function is used to change the student login to Librarian Login or from the librarian student to the student login.

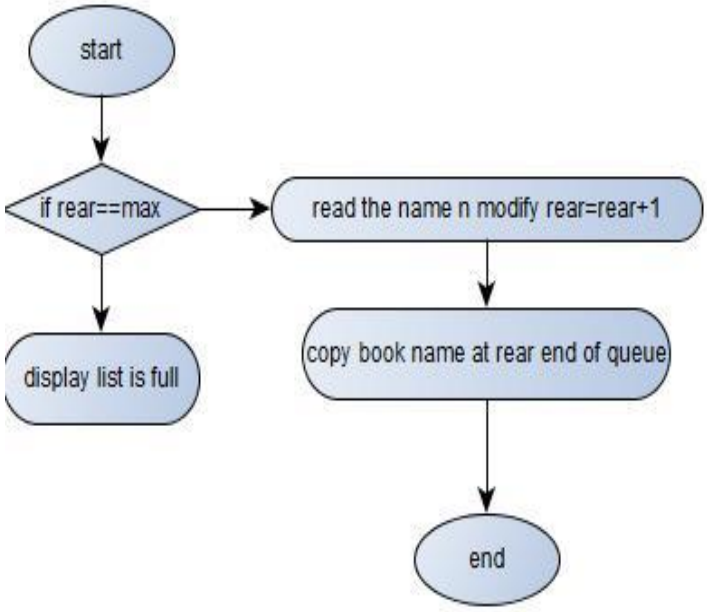
Pseudocode:
Start
The function returns to the login page.
Enter the id and check whether is valid.
If the Id is valid (check whether the Password is Valid).
Then provide the menu of operations.
Else print "Input details are not Incorrect."
End

STUDENT PORTAL
Suggest books
User suggests some books that he wishes to get added in library. Suggested books can be viewed by librarian in order in which they were suggested. Pseudocode:

1. Begin
2. Design an enqueue function.
3. Check if (rear == MAX)
4. If true print "List is full"
5. Else Read name of book and modify rear as rear+1.
6. Copy book name at the rear of queue.

7. Check if (front=0)
8. If true modify front=1 and rear=1.
9. Ask user if he wishes to continue and recall the function.
10. End

DATAFLOW DIAGRAM



Provide review for the books

Student can provide ratings for books in each of the department which will help other students in book selection as they can check the experience of other students from the ratings they provided.

Pseudocode:

1) Begin
2) Use Switch Case statement as we must separate the book ratings by department
3) Provide unique case number for each department.
4)
5) Define an array of integer for rating and an array of character for book name in each department.
6) Read the name of book followed by its rating.
7) Ask user if wishes to add more and recall the same function if he wishes to else return to main menu.
8) END

DATAFLOW DIAGRAM

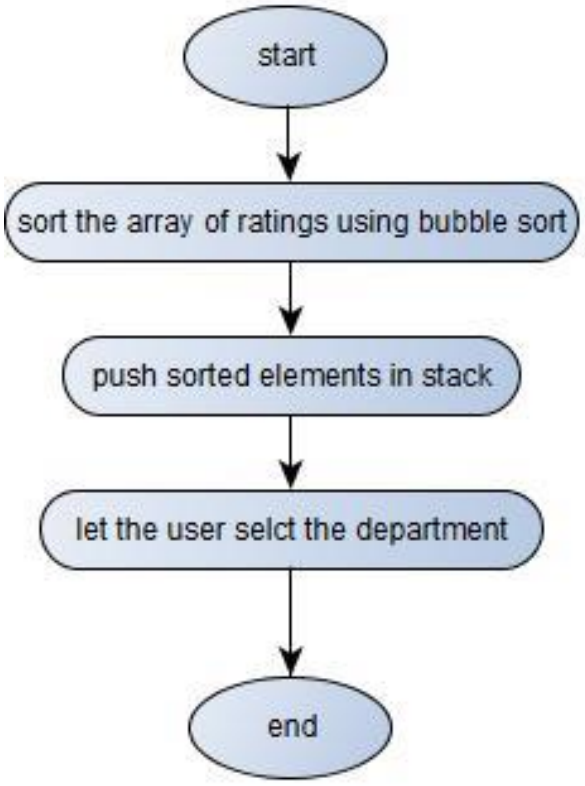
Check the best books on the basics of reviews:

You can check best books in each department as books appear in descending order of their

rating with best rated books at top.
Pseudocode:

1) Begin
2) Sort the ratings provided using Bubble Sort technique in each department.
3) Push the results in a Stack.
4) Pop the elements of Stack corresponding department for which user wishes see ratings
5) Books are displayed with their name and rating corresponding to them with highest rating at top in each department.
6) END

DATAFLOW DIAGRAMS



Recently added books:
Student gets to see the list of recently added books in each department i.e. mechanical, computer, electrical, electronics etc.
Pseudocode:

1) Visit the list where the data given by the librarian is stored.
2) visit each node in the list.
3) If the nodes are empty
   print (“There are No books available”)
4) else
   print (category; Id; name; author; quantity, rack number of the books available)

CHANGE LOGIN

Already defined in Librarian Portal this feature is used to switch login and the same window that u used to come at the start of program appears so that you can login as librarian or any other student.

Pseudocode:

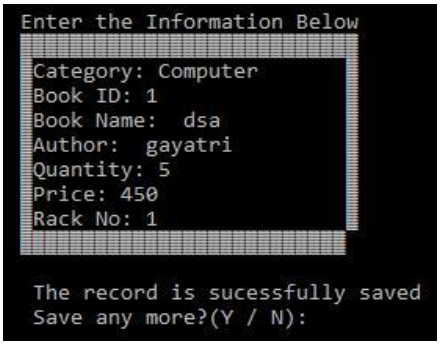
1) Begin
2) Read from user whether he wishes to login as librarian or Student.
3) For Librarian read each character and put '*' on screen so no one else can see password.
4) Check if password matches with preset password.
5) If it matches Land him to librarian main menu.
6) For student he is asked to enter unique number assigned to him.
7) If that number which is registration number in this case is present in file maintained by librarian
8) then land on to student main menu.
9) END

RESULTS AND DISCUSSION

DISCUSSION ABOUT OUTPUTS OBTAINED

1) LIBRARIAN PORTAL:

   When you add books to the database. Adding books to the database can be done by using different kinds of data structures like stacks, queues and linked lists. But the more efficient way of adding books can be done by linked lists after calculating the run time complexity.



To delete books , by using unique book Id:

```
Enter the Book ID to  delete:1

The book record is available
Book name is dsa
Do you want to delete it?(Y/N):
The record is sucessfully deleted
Delete another record?(Y/N)
```

Searching a book by its name:

```
      ****Search Books By Name****
Enter Book Name:dsa

The Book is available
 ID:12345
 Name:dsa
 Author:amulya
 Qantity:3
 Price:Rs.345.00
 Rack No:12

Try another search?(Y/N)
```

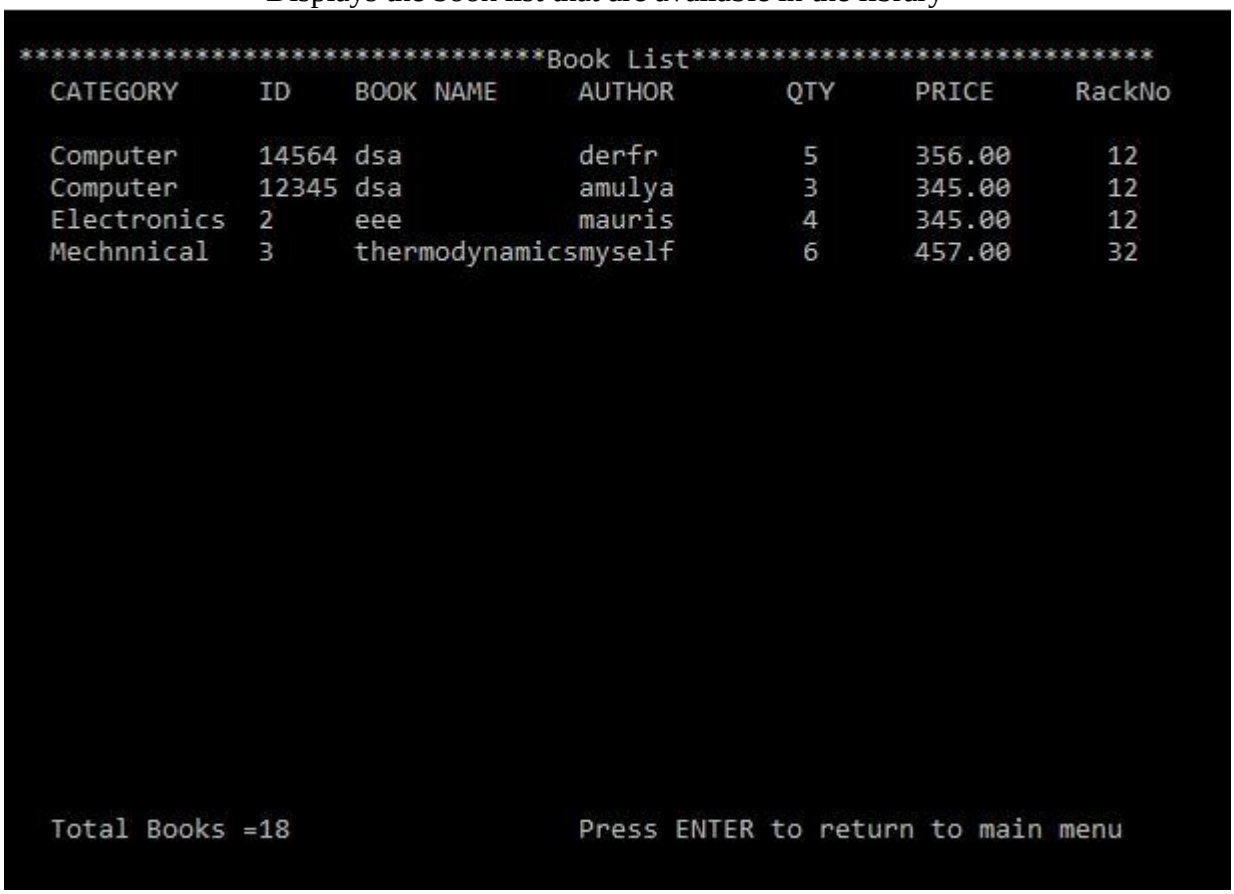
To issue the book. Its gives the in time and out time when the book is to be returned.

```
      ***Issue Book section***

Enter the Book Id:2

The book record is available
There are 4 unissued books in library
The name of book is eee
Enter student name:amulya
Issued date=0-0-0
The BOOK of ID 2 is issued
To be return:15-0-0
Issue any more(Y/N):
```

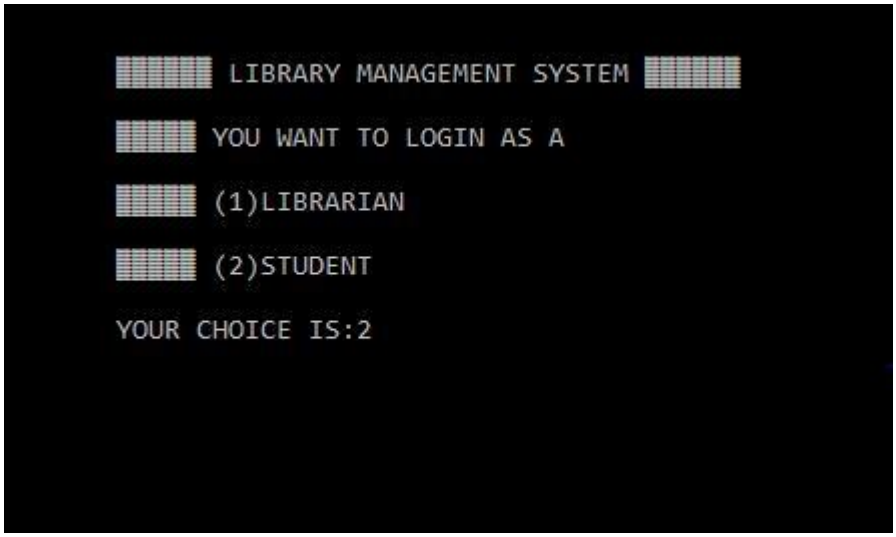
View list of the books issued which can be completely viewed by the librarian:

```
**********************************Issued book list**********************************

STUDENT NAME    CATEGORY    ID    BOOK NAME    ISSUED DATE    RETURN DATE

amulya          Computer    12345 dsa            0-0-0          15-0-0
amulya          Electronics 2     eee            0-0-0          15-0-0
```

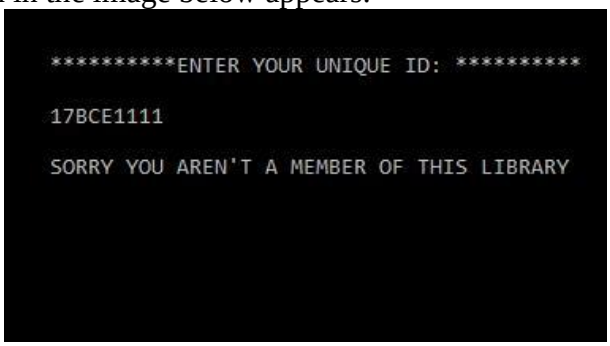
Displays the book list that are available in the library



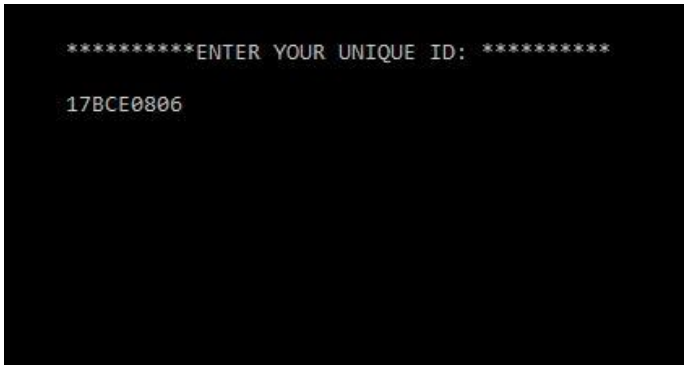
PASSWORD VALIDATION



Here it can be seen that if a registration number belonging to that is class is entered then only it takes to student portal else a prompt message appears.
For example, student with this registration doesn't exist in this class therefore we find that prompt message shown in the image below appears.



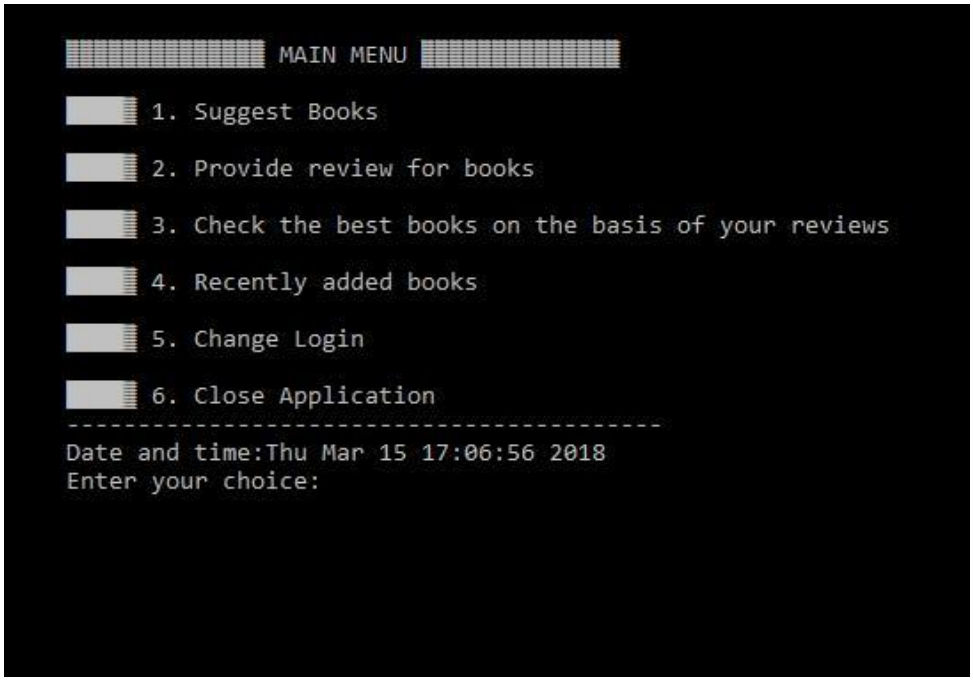
Now consider the example when I entered my registration number we find that Student menu gets opened.



STUDENT PORTAL

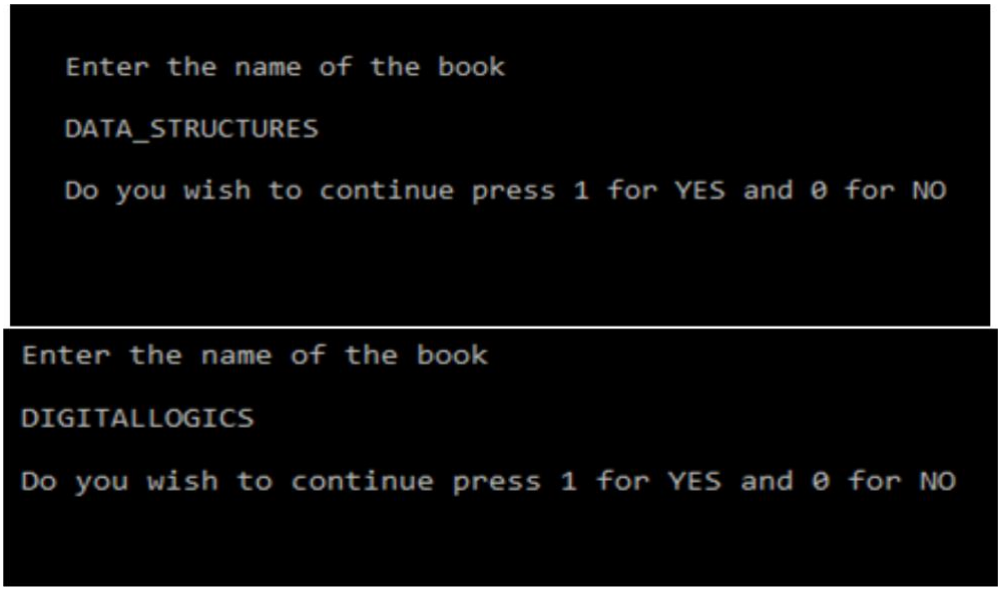
STUDENT MAIN MENU

Here you can see the various modules to which student has access to.



SUGGESTING BOOKS

Student can suggest books that wishes to get added into library. We have made use of **Queues** here and they are best to use among all the data structures we have because priority should be given first to the student who suggested book first and as queues work on the principal of FIFO they tend to be the best among all.

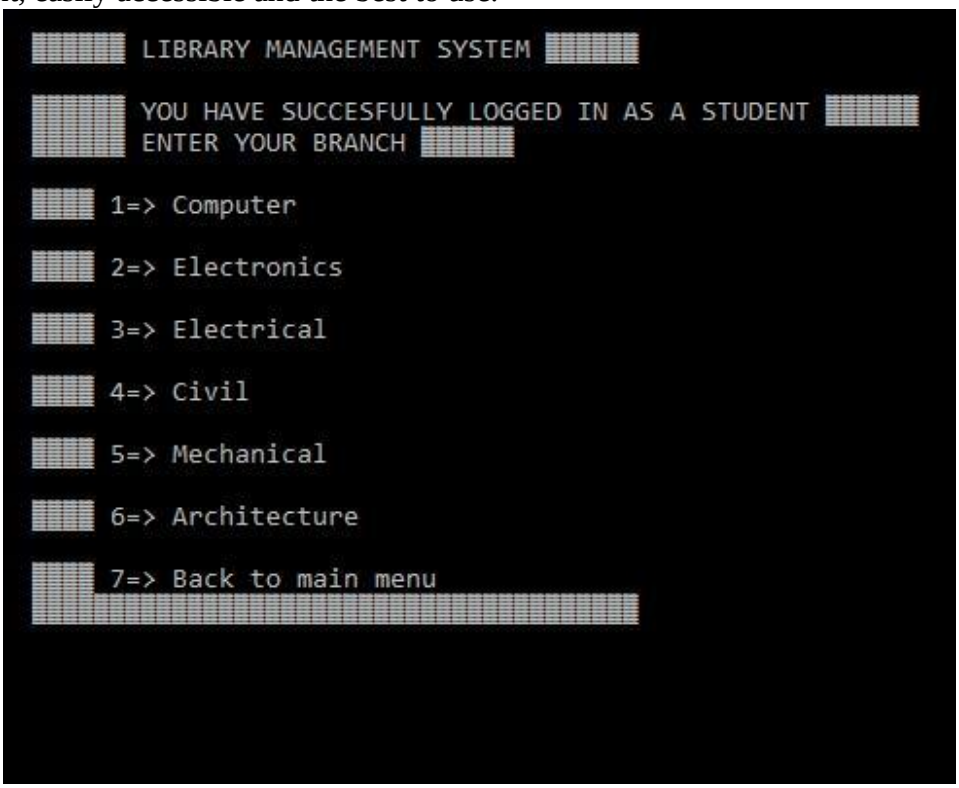


These books can be further viewed in librarian portal as a list of suggested books.

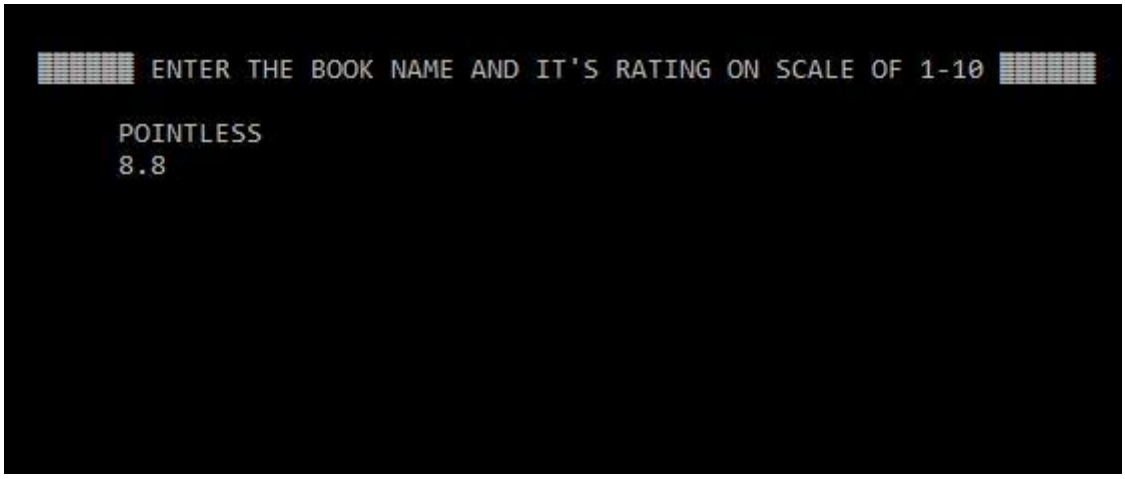
PROVIDING RATING FOR BOOKS

Suppose we select computer. Now we can provide review for the books in department of computer science based on which our next important function check books based on review will work. We have made use of **2-D Dimensional Arrays** here for storing the Book name and

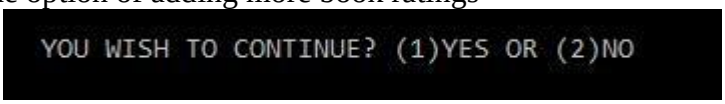
its ratings as we aren't required to perform any special operation therefore they are simple, convenient, easily accessible and the best to use.



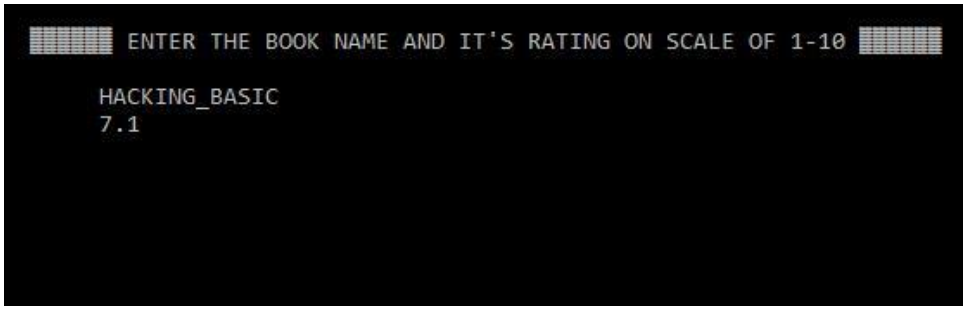
Suppose we select Computer and provide a rating for random books in computer science field Later in check best books based on your review we will find that books are sorted in descending order of their rating thus giving list of best books at the top. Visual example given below will clear functioning of this module.



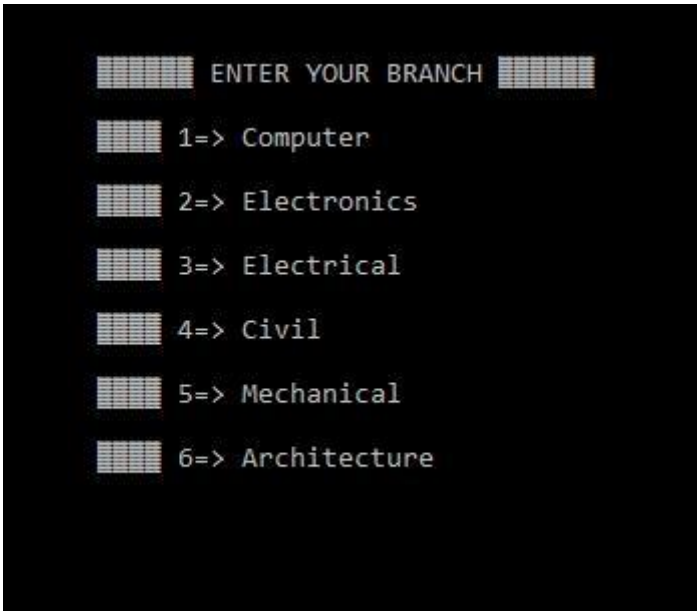
Further you find the option of adding more book ratings



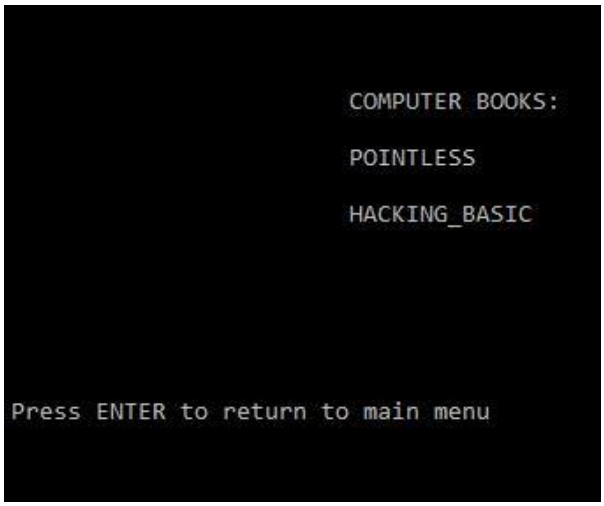
Providing more ratings in computer will make functioning of this module more clearer.



CHECKING BEST BOOKS ON THE BASIS OF THEIR RATINGS

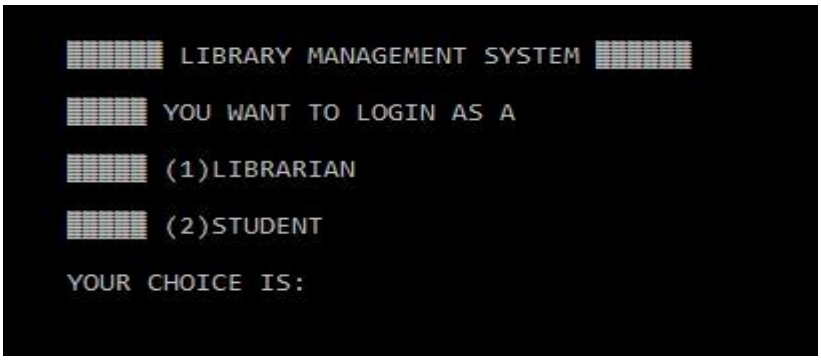


It can be easily seen that books are displayed in the order of their rating in descending order such that topnotch books in that field are at the top. We have **Bubble Sorted** the Array and have pushed the sorted elements into a **Stack.** Now the book which has least rating would be placed first in Stack and we know that Stack is based on principle of LIFO therefore topmost element would be book with best rating and popping them out would display Book names in descending order of their rating thus again we find that Stacks were the most suitable Data Structure available for this module.



CHANGE LOGIN

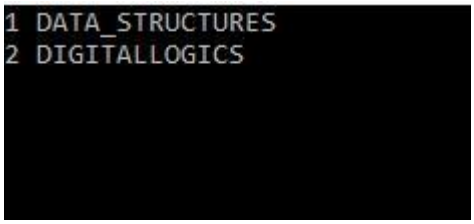
It again brings us the same window which we got at starting and we have already discussed about the working behind it which includes **File Handling** as id is to verified from a big database.



LIBRARIAN PORTAL

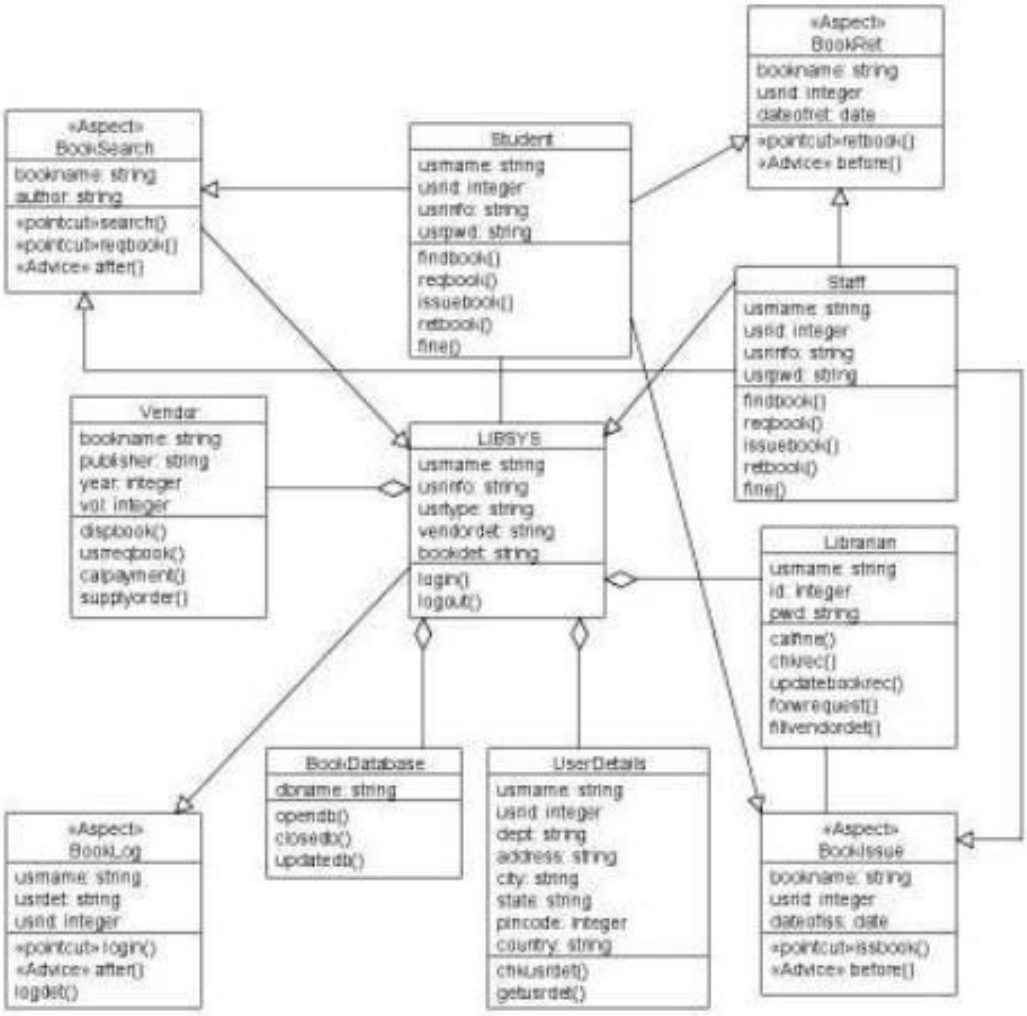
SUGGESTED BOOKS

We suggested some books in student portal these books can be viewed under librarian portal. We know the functioning behind it , it's just dequeuing of the queue elements and displaying them.



This UML class diagram shows the Library Management System. There is a total of 7 classes as Libsys, student, vendor, librarian, staff, book database and user details, and 4 aspects as book search, book log, booklet and book issue. This system describes the management of library books and the operation related to issue and return. The book search aspect is used to find any book from the catalogue. Book ret and Book issue aspects are used for issuing and returning of the book respectively. The main users of this system are librarian, students and staff. The vendor class is used to supply any book demanded by the system and records the payment against that order.

UML DIAGRAM FOR LIBRARY MANAGEMENT SYSTEM


«Aspect»
BookRet
bookname: string
usrid: integer
dateofret: date
«pointcut»retbook()
«Advice» before()
«Aspect»
BookSearch
bookname: string
author: string
«pointcut»search()
«pointcut»reqbook()
«Advice» after()
Student
usrname: string
usrid: integer
usrinfo: string
usrpwd: string
findbook()
reqbook()
issuebook()
retbook()
fine()
Staff
usrname: string
usrid: integer
usrinfo: string
usrpwd: string
findbook()
reqbook()
issuebook()
retbook()
fine()
Vendor
bookname: string
publisher: string
year: integer
vol: integer
dispbook()
usreqbook()
callpayment()
supplyorder()
LIBSYS
usrname: string
usrinfo: string
usrtype: string
vendordet: string
bookdet: string
login()
logout()
Librarian
usrname: string
id: integer
pwd: string
callfine()
chkrec()
updatebookrec()
forwrequest()
fillvendordet()
BookDatabase
dbname: string
opendb()
closedb()
updatedb()
UserDetails
usrname: string
usrid: integer
dept: string
address: string
city: string
state: string
pincode: integer
country: string
chkusrdet()
getusrdet()
«Aspect»
BookLog
usrname: string
usrdet: string
usrid: integer
«pointcut»login()
«Advice» after()
logdet()
«Aspect»
BookIssue
bookname: string
usrid: integer
dateofiss: date
«pointcut»issbook()
«Advice» before()

CONCLUSION

In recent years,modernization has made necessary for any sphere of life to adapt to the new and easier technology especially in the institutions. Library management systems process made computerized to reduce human errors and to increase the efficiency. The main focus of this project is to lessen human efforts.

This project work is aimed at making the institution library available to students and library staff at a click of mouse in respective areas of choice. It makes entire process online where students can search books, staff can generate reports and do book transactions. It also has a facility for student login where student can see status of books issued as well give some suggestions and can search for a book of a particular subject. It has a facility of library staff's login where they can view list of books,edit books record, view suggested books list,issue the books to the students and maintain the record of the issued books and can also check how many books are issued and stock available in the library. In this, we can maintain the late fine of students who returns the issued books after the due date.

It is expected that this project will go a long way in satisfying users requirements.

There is a future scope of this facility that many more features such as online lectures video tutorials can be added by library staff, latest learning facilities such as articles,journals,projects,thesis etc. with the computer systems without going through the rigorous steps and routine in the conventional institution libraries.

Code

```
#include <windows.h>0

#include<stdio.h>

#include<conio.h>              //contains delay(),getch(),gotoxy(),etc.


#include <stdlib.h>

#include<string.h>             //contains strcmp(),strcpy(),strlen(),etc

#include<ctype.h>              //contains toupper(), tolower(),etc

#include<dos.h>                //contains _dos_getdate

#include<time.h>

#define RETURNTIME 15

//list of function prototype

#define MAX 50

char qu[MAX][30],item[30];
```

```
int rear=0,front=0;
char
catagories[][15]={"Computer","Electronics","Electrical","Civil","Mechnnical","Architecture"};
void returnfunc(void);
void addit();
void mainmenu(void);
void addbooks(void);
void deletebooks(void);
void editbooks(void);
void searchbooks(void);
void issuebooks(void);
void viewbooks(void);
void closeapplication(void);
int getdata();
int checkid(int);
int t(void);
void Password();
void issuerecord();
void loaderanim();
void suggestbooks();
void deq();
void enq();
void mainmenu1();
void option();
void Check();
void display();
void stumenu(void);
void sort(float,char);
void schoice(void);
```

```
void displaybooks(int);


COORD coord = {0, 0}; // sets coordinates to 0,0
//COORD max_buffer_size = GetLargestConsoleWindowSize(hOut);
COORD max_res,cursor_size;
void gotoxy (int x, int y)
{
      coord.X = x; coord.Y = y; // X and Y coordinates
      SetConsoleCursorPosition(GetStdHandle(STD_OUTPUT_HANDLE), coord);
}
void delay(unsigned int mseconds)
{
   clock_t goal = mseconds + clock();
   while (goal > clock());
}
//list of global files that can be accessed form anywhere in
program FILE *fp,*ft,*fs;


//list of global variable
int s;
char findbook;
char password[10]={"codechamp"};


struct meroDate
{
   int mm,dd,yy;
};
struct books
{
```

```
    int id;
    char stname[20];
    char name[20];
    char Author[20];
    int quantity;
    float Price;
    int count;
    int rackno;
    char *cat;
    struct meroDate issued;
    struct meroDate duedate;
};
struct books a;
int Check(char *fname, char *str)
{
FILE *file;
int i = 1,j=1;
int r = 0;
char temp[10];
if((file = fopen(fname, "r")) == NULL) {
    return(-1);
}
while(fgets(temp,10, file) != NULL) {
            if((strcmp(temp, str))==0)
            {
            gotoxy(15,9);
            printf("VALIDATION SUCCESSFUL");
            mainmenu1();
```

```
                r++;
   }
   i++;
}
if(r == 0) {
        gotoxy(15,9);
   printf("\aSORRY YOU AREN'T A MEMBER OF THIS LIBRARY ");
}

if(file) {
   fclose(file);
}
return(0);
}
void option(){
system("cls");
int ch;
gotoxy(20,3);
printf("\xB2\xB2\xB2\xB2\xB2\xB2 LIBRARY MANAGEMENT SYSTEM
\xB2\xB2\xB2\xB2\xB2\xB2");
gotoxy(20,5);
printf("\xB2\xB2\xB2\xB2\xB2 YOU WANT TO LOGIN AS A \n");
gotoxy(20,7);
printf("\xB2\xB2\xB2\xB2\xB2 (1)LIBRARIAN \n");
gotoxy(20,9);
printf("\xB2\xB2\xB2\xB2\xB2 (2)STUDENT \n");
gotoxy(20,11);
printf("YOUR CHOICE IS:");
gotoxy(35,11);
```

```
scanf("%d",&ch);
switch(ch){
case 1:
        Password();
   break;
case 2:
   system("cls");
   gotoxy(15,5);
        printf("**********ENTER YOUR UNIQUE ID: ********** \n");
        char s[11];
        gotoxy(15,7);
        scanf("%s",&s);
        Check("a.txt",s);
   break;
default:
      gotoxy(10,23);
      printf("\aWrong Entry!!Please re-entered correct
      option"); if(getch())
         option();

}
}
int main()

{
     option();
     getch();
     return 0;
```

```
}
void mainmenu1()
{
   system("cls");
   int i;
   gotoxy(20,3);
   printf("\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2 MAIN MENU");
   printf(" \xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2");
   gotoxy(20,5);
   printf("\xDB\xDB\xDB\xDB\xB2 1. Suggest
   Books"); gotoxy(20,7);
   printf("\xDB\xDB\xDB\xDB\xB2 2. Provide review for books");
   gotoxy(20,9);
   printf("\xDB\xDB\xDB\xDB\xB2 3. Check the best books on the basis of your
   reviews"); gotoxy(20,11);
   printf("\xDB\xDB\xDB\xDB\xB2 4. Recently added
   books"); gotoxy(20,13);
   printf("\xDB\xDB\xDB\xDB\xB2 5. Change Login");
   gotoxy(20,15);
   printf("\xDB\xDB\xDB\xDB\xB2 6. Close Application");
   gotoxy(20,16);
   printf("----------------------------------------");
   gotoxy(20,17);
   t();
   gotoxy(20,18);
   printf("Enter your choice:");
   switch(getch())
   {
      case '1':
```

```
suggestbooks();
break;
    case '2':
stumenu();
break;
case '3':
schoice();
break;
    case '4':
option();
break;
case '5':
option();
break;
    case '6':
{
system("cls");
gotoxy(16,3);
printf("Thanks for using the Program..");
gotoxy(10,7);
printf("Exiting in 5 second.....>");
//flushall();
delay(5000);
exit(0);
}
break;

default:
{
```

```
        gotoxy(10,23);
        printf("\aWrong Entry!!Please re-entered correct option");
        if(getch())
        mainmenu1();
        }

    }
}
void suggestbooks()
{
        system("cls");
        enq();
}
void addit()
{
        system("cls");
        char q[11];
        gotoxy(10,5);
        scanf("%s",q);
        FILE *out = fopen("a.txt", "a");
        fprintf(out, "%s", q);
        gotoxy(10,7);
        printf("RECORD ADDED SUCCESSFULLY");
        delay(2000);
        fclose(out);
        mainmenu();
}
void enq()
{
```

```
system("cls");
if (rear == MAX)
{printf("List is full");}
else
{
int c;
char a[30];
gotoxy(5,5);
printf("Enter the name of the book \n");
gotoxy(5,7);
//scanf ("%[^\n]%*c", a);
scanf("%s",&a);
//fgets(a,30,stdin);
rear = rear + 1;
strcpy(qu[rear],a);
if(front==0)
{front=1;}
gotoxy(5,9);
printf("Do you wish to continue press 1 for YES and 0 for NO \n");
scanf("%d",&c);
switch(c){
case 1:
        enq();
        break;
case 0:
        mainmenu1();
        break;
default:
{printf("Enter the correct choice ");
```

```
enq();
break;}
}}}
void deq()
{
if (front == 0)
{printf("That's all folks\ ");}
else if(front == rear)
front = rear = 0;
else
{
strcpy(item,qu[front]);
strcpy(qu[front],0);
front = front + 1;
printf("%d",item);
}}
void display()
{
int e;
system("cls");
int i;
if (front == 0)
printf("Queue is empty");
else
{
for (i = front; i <= rear; i++)
{
printf("%d %s \n",i,qu[i]);
}
```

```
}
gotoxy(50,20);
printf("Press 1 to go back to Mainmenu ");
scanf("%d",&e);
switch(e){
case 1:
        mainmenu();
        break;
}
}
void mainmenu()
{
   system("cls");
   int i;
   gotoxy(20,3);
   printf("\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2 MAIN MENU");
   printf(" \xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2");
   gotoxy(20,5);
   printf("\xDB\xDB\xDB\xDB\xB2 1. Add Books  ");
   gotoxy(20,7);
   printf("\xDB\xDB\xDB\xDB\xB2 2. Delete books");
   gotoxy(20,9);
   printf("\xDB\xDB\xDB\xDB\xB2 3. Search Books");
   gotoxy(20,11);
   printf("\xDB\xDB\xDB\xDB\xB2 4. Issue Books");
   gotoxy(20,13);
   printf("\xDB\xDB\xDB\xDB\xB2 5. View Book list");
   gotoxy(20,15);
   printf("\xDB\xDB\xDB\xDB\xB2 6. View Suggested Book List ");
```

```
gotoxy(20,17);
printf("\xDB\xDB\xDB\xDB\xB2 7. Edit Book's
Record"); gotoxy(20,19);
printf("\xDB\xDB\xDB\xDB\xB2 8. Add a new student to
library"); gotoxy(20,21);
printf("\xDB\xDB\xDB\xDB\xB2 9. Change Login");
gotoxy(20,23);
printf("\xDB\xDB\xDB\xDB\xB2 10. Close Application");
      gotoxy(20,25);
printf("----------------------------------------");
gotoxy(20,26);
t();
gotoxy(20,27);
printf("Enter your choice:");
switch(getch())
{
   case '1':
   addbooks();
   break;
      case '2':
   deletebooks();
   break;
   case '3':
   searchbooks();
   break;
               case '4':
   issuebooks();
   break;
   case '5':
```

```
viewbooks();
break;
case '6':
display();
break;
case '7':
editbooks();
break;
case '8':
addit();
break;
case '9':
option();
break;
case '10':
{
system("cls");
gotoxy(16,3);
printf("Thanks for using the Program..");
gotoxy(10,7);
printf("Exiting in 5 second.....>");
//flushall();
delay(5000);
exit(0);
}
break;
default:
{
gotoxy(10,23);
```

```
        printf("\aWrong Entry!!Please re-entered correct option");
        if(getch())
        mainmenu();
        }

    }
}
void addbooks(void)    //funtion that add books
{
    system("cls");
    int i;
    gotoxy(20,5);
    printf("\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2SELECT CATEGOIES");
    printf("\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2");
    gotoxy(20,7);
    printf("\xDB\xDB\xDB\xDB\xB2 1. Computer");
    gotoxy(20,9);
    printf("\xDB\xDB\xDB\xDB\xB2 2. Electronics");
    gotoxy(20,11);
    printf("\xDB\xDB\xDB\xDB\xB2 3. Electrical");
    gotoxy(20,13);
    printf("\xDB\xDB\xDB\xDB\xB2 4. Civil");
    gotoxy(20,15);
    printf("\xDB\xDB\xDB\xDB\xB2 5. Mechanical");
    gotoxy(20,17);
    printf("\xDB\xDB\xDB\xDB\xB2 6. Architecture");
    gotoxy(20,19);
    printf("\xDB\xDB\xDB\xDB\xB2 7. Back to main
    menu"); gotoxy(20,21);
```

```
printf("\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\\xB2\xB2\xB2" );


printf("\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2");
   gotoxy(20,22);
   printf("Enter your choice:");
   scanf("%d",&s);
   if(s==7)

   mainmenu() ;
   system("cls");
   fp=fopen("Bibek.dat","ab+");
   if(getdata()==1)
   {
   a.cat=catagories[s-1];
   fseek(fp,0,SEEK_END);
   fwrite(&a,sizeof(a),1,fp);
   fclose(fp);
   gotoxy(21,14);
   printf("The record is sucessfully saved");
   gotoxy(21,15);
   printf("Save any more?(Y / N):");
   if(getch()=='n')
      mainmenu();
   else
      system("cls");
      addbooks();
   }
```

```
}
void deletebooks()    //function that delete items from file fp
{
   system("cls");
   int d;
   char another='y';
   while(another=='y') //infinite loop
   {
   system("cls");
   gotoxy(10,5);
   printf("Enter the Book ID to delete:");
   scanf("%d",&d);
   fp=fopen("Bibek.dat","rb+");
   rewind(fp);
   while(fread(&a,sizeof(a),1,fp)==1)
   {
      if(a.id==d)
      {

      gotoxy(10,7);
      printf("The book record is available");
      gotoxy(10,8);
      printf("Book name is %s",a.name);
      gotoxy(10,9);
      printf("Rack No. is %d",a.rackno);
      findbook='t';
      }
   }
   if(findbook!='t')
```

```
{
   gotoxy(10,10);
   printf("No record is found modify the search");
   if(getch())
   mainmenu();
}
if(findbook=='t' )
{
   gotoxy(10,9);
   printf("Do you want to delete it?(Y/N):");
   if(getch()=='y')
   {
   ft=fopen("test.dat","wb+"); //temporary file for delete
   rewind(fp);
   while(fread(&a,sizeof(a),1,fp)==1)
   {
      if(a.id!=d)
      {
      fseek(ft,0,SEEK_CUR);
      fwrite(&a,sizeof(a),1,ft); //write all in tempory file except that
      }                          //we want to delete
   }
   fclose(ft);
   fclose(fp);
   remove("Bibek.dat");
   rename("test.dat","Bibek.dat"); //copy all item from temporary file to fp except
   that fp=fopen("Bibek.dat","rb+"); //we want to delete if(findbook=='t')

   {
```

```
            gotoxy(10,10);
            printf("The record is sucessfully deleted");
            gotoxy(10,11);
            printf("Delete another record?(Y/N)");
        }
        }
    else
    mainmenu();
    fflush(stdin);
    another=getch();
    }
    }
    gotoxy(10,15);
    mainmenu();
}
void searchbooks()
{
    system("cls");
    int d;
    printf("*****************************Search
Books*********************************");
    gotoxy(20,10);
    printf("\xDB\xDB\xDB\xB2 1. Search By ID");
    gotoxy(20,14);
    printf("\xDB\xDB\xDB\xB2 2. Search By Name");
    gotoxy( 15,20);
    printf("Enter Your Choice");
    fp=fopen("Bibek.dat","rb+"); //open file for reading propose
    rewind(fp);     //move pointer at the begining of file
```

```
switch(getch())
{
 case '1':
{
   system("cls");
   gotoxy(25,4);
   printf("****Search Books By Id****");
   gotoxy(20,5);
   printf("Enter the book id:");
   scanf("%d",&d);
   gotoxy(20,7);
   printf("Searching.....");
   while(fread(&a,sizeof(a),1,fp)==1)
   {
   if(a.id==d)
   {
      delay(2);
      gotoxy(20,7);
      printf("The Book is available");
      gotoxy(20,8);
      printf("\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2");
      printf("\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2");
      gotoxy(20,9);
      printf("\xB2 ID:%d",a.id);gotoxy(47,9);printf("\xB2");
      gotoxy(20,10);
      printf("\xB2 Name:%s",a.name);gotoxy(47,10);printf("\xB2");
      gotoxy(20,11);
      printf("\xB2 Author:%s
      ",a.Author);gotoxy(47,11);printf("\xB2"); gotoxy(20,12);
```

```
        printf("\xB2 Quantity:%d ",a.quantity);gotoxy(47,12);printf("\xB2");
gotoxy(47,11);printf("\xB2");

        gotoxy(20,13);

        printf("\xB2 Price:Rs.%.2f",a.Price);gotoxy(47,13);printf("\xB2");

        gotoxy(20,14);

        printf("\xB2 Rack No:%d ",a.rackno);gotoxy(47,14);printf("\xB2");

        gotoxy(20,15);

        printf("\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2");

        printf("\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2");

        findbook='t';

    }


    }

    if(findbook!='t') //checks whether conditiion enters inside loop or not

    {

    gotoxy(20,8);


printf("\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\x
B2");

    gotoxy(20,9);printf("\xB2"); gotoxy(38,9);printf("\xB2");

    gotoxy(20,10);


printf("\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\x
B2");

    gotoxy(22,9);printf("\aNo Record Found");

    }

    gotoxy(20,17);

    printf("Try another search?(Y/N)");

    if(getch()=='y')

    searchbooks();

    else
```

```
    mainmenu();
    break;
}
case '2':
{
    char s[15];
    system("cls");
    gotoxy(25,4);
    printf("****Search Books By Name****");
    gotoxy(20,5);
    printf("Enter Book Name:");
    scanf("%s",s);
    int d=0;
    while(fread(&a,sizeof(a),1,fp)==1)
    {
    if(strcmp(a.name,(s))==0) //checks whether a.name is equal to s or not
    {
        gotoxy(20,7);
        printf("The Book is available");
        gotoxy(20,8);
        printf("\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2");
        printf("\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2");
        gotoxy(20,9);
        printf("\xB2 ID:%d",a.id);gotoxy(47,9);printf("\xB2");
        gotoxy(20,10);
        printf("\xB2 Name:%s",a.name);gotoxy(47,10);printf("\xB2");
        gotoxy(20,11);
        printf("\xB2 Author:%s",a.Author);gotoxy(47,11);printf("\xB2");
        gotoxy(20,12);
```

```
        printf("\xB2 Qantity:%d",a.quantity);gotoxy(47,12);printf("\xB2");
        gotoxy(20,13);
        printf("\xB2 Price:Rs.%.2f",a.Price);gotoxy(47,13);printf("\xB2");
        gotoxy(20,14);
        printf("\xB2 Rack No:%d ",a.rackno);gotoxy(47,14);printf("\xB2");
        gotoxy(20,15);
        printf("\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2");
        printf("\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2");
        d++;
    }

    }
    if(d==0)
    {
    gotoxy(20,8);
printf("\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2");
    gotoxy(20,9);printf("\xB2"); gotoxy(38,9);printf("\xB2");
    gotoxy(20,10);
printf("\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2");
    gotoxy(22,9);printf("\aNo Record Found");
    }
    gotoxy(20,17);
    printf("Try another search?(Y/N)");
    if(getch()=='y')
    searchbooks();
    else
```

```
      mainmenu();
      break;
   }
   default :
   getch();
   searchbooks();
   }
   fclose(fp);
}
void issuebooks(void) //function that issue books from library
{
   int t;

   system("cls");
   printf("*********************************ISSUE
SECTION**************************");
   gotoxy(10,5);
   printf("\xDB\xDB\xDB\xDb\xB2 1. Issue a
   Book"); gotoxy(10,7);
   printf("\xDB\xDB\xDB\xDb\xB2 2. View Issued
   Book"); gotoxy(10,9);
   printf("\xDB\xDB\xDB\xDb\xB2 3. Search Issued Book");
   gotoxy(10,11);
   printf("\xDB\xDB\xDB\xDb\xB2 4. Remove Issued Book");
   gotoxy(10,14);
   printf("Enter a Choice:");
   switch(getch())
   {
   case '1': //issue book
```

```
{
    system("cls");
    int c=0;
    char another='y';
    while(another=='y')
    {
        system("cls");
        gotoxy(15,4);
        printf("***Issue Book section***");
        gotoxy(10,6);
        printf("Enter the Book Id:");
        scanf("%d",&t);
        fp=fopen("Bibek.dat","rb");
        fs=fopen("Issue.dat","ab+");
        if(checkid(t)==0) //issues those which are present in library
        {
            gotoxy(10,8);
            printf("The book record is available");
            gotoxy(10,9);
            printf("There are %d unissued books in library ",a.quantity);
            gotoxy(10,10);
            printf("The name of book is %s",a.name);
            gotoxy(10,11);
            printf("Enter student name:");
            scanf("%s",a.stname);
           // struct dosdate_t d; //for current date
           // _dos_getdate(&d);
           // a.issued.dd=d.day;
           // a.issued.mm=d.month;
```

```
    //a.issued.yy=d.year;
    gotoxy(10,12);
    printf("Issued
    date=%d-%d-%d",a.issued.dd,a.issued.mm,a.issued.yy);
    gotoxy(10,13);
    printf("The BOOK of ID %d is issued",a.id);
    a.duedate.dd=a.issued.dd+RETURNTIME;  //for return date
    a.duedate.mm=a.issued.mm;
    a.duedate.yy=a.issued.yy;
    if(a.duedate.dd>30)
    {
    a.duedate.mm+=a.duedate.dd/30;
    a.duedate.dd-=30;

    }
    if(a.duedate.mm>12)
    {
   a.duedate.yy+=a.duedate.mm/12;
   a.duedate.mm-=12;

    }
    gotoxy(10,14);
    printf("To be return:%d-%d-%d",a.duedate.dd,a.duedate.mm,a.duedate.yy);
    fseek(fs,sizeof(a),SEEK_END);
    fwrite(&a,sizeof(a),1,fs);
    fclose(fs);
    c=1;
}
if(c==0)
{
```

```
            gotoxy(10,11);
            printf("No record found");
            }
            gotoxy(10,15);
            printf("Issue any more(Y/N):");
            fflush(stdin);
            another=getche();
            fclose(fp);
        }

        break;
    }
    case '2': //show issued book list
    {
        system("cls");
        int j=4;
        printf("*******************************Issued book
list*******************************\n");
        gotoxy(2,2);
        printf("STUDENT NAME CATEGORY ID BOOK NAME ISSUED DATE RETURN
DATE");
          fs=fopen("Issue.dat","rb");
        while(fread(&a,sizeof(a),1,fs)==1)
        {

            gotoxy(2,j);
            printf("%s",a.stname);
            gotoxy(18,j);
            printf("%s",a.cat);
            gotoxy(30,j);
```

```
            printf("%d",a.id);
            gotoxy(36,j);
            printf("%s",a.name);
            gotoxy(51,j);
            printf("%d-%d-%d",a.issued.dd,a.issued.mm,a.issued.yy );
            gotoxy(65,j);
            printf("%d-%d-%d",a.duedate.dd,a.duedate.mm,a.duedate.yy);
            //struct dosdate_t d;
            //_dos_getdate(&d);
            gotoxy(50,25);
//          printf("Current date=%d-%d-%d",d.day,d.month,d.year);
            j++;

        }
        fclose(fs);
        gotoxy(1,25);
        returnfunc();
    }
    break;
    case '3':  //search issued books by id
    {
        system("cls");
        gotoxy(10,6);
        printf("Enter Book ID:");
        int p,c=0;
        char another='y';
        while(another=='y')
        {
```

```
        scanf("%d",&p);
        fs=fopen("Issue.dat","rb");
        while(fread(&a,sizeof(a),1,fs)==1)
        {
           if(a.id==p)
           {
              issuerecord();
              gotoxy(10,12);
              printf("Press any key.....");
              getch();
              issuerecord();
              c=1;
           }

        }
        fflush(stdin);
        fclose(fs);
        if(c==0)
        {
           gotoxy(10,8);
           printf("No Record Found");
        }
        gotoxy(10,13);
        printf("Try Another Search?(Y/N)");
        another=getch();
     }
  }
  break;
  case '4': //remove issued books from list
```

```
{
   system("cls");
   int b;
   FILE *fg; //declaration of temporary file for delete
   char another='y';
   while(another=='y')
   {
      gotoxy(10,5);
      printf("Enter book id to remove:");
      scanf("%d",&b);
      fs=fopen("Issue.dat","rb+");
      while(fread(&a,sizeof(a),1,fs)==1)
      {
         if(a.id==b)
         {
            issuerecord();
            findbook='t';
         }
         if(findbook=='t')
         {
            gotoxy(10,12);
            printf("Do You Want to Remove it?(Y/N)");
            if(getch()=='y')
            {
               fg=fopen("record.dat","wb+");
               rewind(fs);
               while(fread(&a,sizeof(a),1,fs)==1)
               {
                  if(a.id!=b)
```

```
                {
                fseek(fs,0,SEEK_CUR);
                fwrite(&a,sizeof(a),1,fg);
                }
            }
            fclose(fs);
            fclose(fg);
            remove("Issue.dat");
            rename("record.dat","Issue.dat");
            gotoxy(10,14);
            printf("The issued book is removed from list");

            }

        }
        if(findbook!='t')
        {
            gotoxy(10,15);
            printf("No Record Found");
        }
    }
    gotoxy(10,16);
    printf("Delete any more?(Y/N)");
    another=getch();
  }
}
default:
gotoxy(10,18);
printf("\aWrong Entry!!");
```

```
    getch();
    issuebooks();
    break;
     }
     gotoxy(1,30);
     returnfunc();
}
void viewbooks(void) //show the list of book persists in library
{
    int i=0,j;
    system("cls");
    gotoxy(1,1);
    printf("*********************************Book
List*****************************");
    gotoxy(2,2);
    printf(" CATEGORY    ID    BOOK NAME     AUTHOR      QTY     PRICE     RackNo ");
    j=4;
    fp=fopen("Bibek.dat","rb");
    while(fread(&a,sizeof(a),1,fp)==1)
    {
    gotoxy(3,j);
    printf("%s",a.cat);
    gotoxy(16,j);
    printf("%d",a.id);
    gotoxy(22,j);
    printf("%s",a.name);
    gotoxy(36,j);
    printf("%s",a.Author);
    gotoxy(50,j);
```

```
    printf("%d",a.quantity);
    gotoxy(57,j);
    printf("%.2f",a.Price);
    gotoxy(69,j);
    printf("%d",a.rackno);
    printf("\n\n");
    j++;
    i=i+a.quantity;
     }
     gotoxy(3,25);
     printf("Total Books =%d",i);
     fclose(fp);
     gotoxy(35,25);
     returnfunc();
}
void editbooks(void) //edit information about book
{
    system("cls");
    int c=0;
    int d,e;
    gotoxy(20,4);
    printf("****Edit Books Section****");
    char another='y';
    while(another=='y')
    {
        system("cls");
        gotoxy(15,6);
        printf("Enter Book Id to be edited:");
        scanf("%d",&d);
```

```
fp=fopen("Bibek.dat","rb+");
while(fread(&a,sizeof(a),1,fp)==1)
{
   if(checkid(d)==0)
   {
      gotoxy(15,7);
      printf("The book is availble");
      gotoxy(15,8);
      printf("The Book ID:%d",a.id);
      gotoxy(15,9);
      printf("Enter new
      name:");scanf("%s",a.name); gotoxy(15,10);
      printf("Enter new
      Author:");scanf("%s",a.Author); gotoxy(15,11);
      printf("Enter new
      quantity:");scanf("%d",&a.quantity); gotoxy(15,12);
      printf("Enter new
      price:");scanf("%f",&a.Price); gotoxy(15,13);
      printf("Enter new
      rackno:");scanf("%d",&a.rackno); gotoxy(15,14);
      printf("The record is modified");
      fseek(fp,ftell(fp)-sizeof(a),0);
      fwrite(&a,sizeof(a),1,fp);
      fclose(fp);
      c=1;
   }
   if(c==0)
   {
      gotoxy(15,9);
```

```
            printf("No record found");
          }
       }
       gotoxy(15,16);
       printf("Modify another Record?(Y/N)");
       fflush(stdin);
       another=getch() ;
   }
       returnfunc();
}
void returnfunc(void)
{
   {
   printf(" Press ENTER to return to main menu");
   }
   a:
   if(getch()==13) //allow only use of enter
   mainmenu();
   else
   goto a;
}
int getdata()
{
   int t;
   gotoxy(20,3);
   printf("Enter the Information Below");
   gotoxy(20,4);
   printf("\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2");
   printf("\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2");
```

```
gotoxy(20,5);
printf("\xB2");gotoxy(46,5);printf("\xB2");
gotoxy(20,6);
printf("\xB2");gotoxy(46,6);printf("\xB2");
gotoxy(20,7);
printf("\xB2");gotoxy(46,7);printf("\xB2");
gotoxy(20,8);
printf("\xB2");gotoxy(46,8);printf("\xB2");
gotoxy(20,9);
printf("\xB2");gotoxy(46,9);printf("\xB2");
gotoxy(20,10);
printf("\xB2");gotoxy(46,10);printf("\xB2");
gotoxy(20,11);
printf("\xB2");gotoxy(46,11);printf("\xB2");
gotoxy(20,12);
printf("\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2");
printf("\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2");
gotoxy(21,5);
printf("Category:");
gotoxy(31,5);
printf("%s",catagories[s-1]);
gotoxy(21,6);
printf("Book ID:\t");
gotoxy(30,6);
scanf("%d",&t);
if(checkid(t) == 0)
{
   gotoxy(21,13);
   printf("\aThe book id already exists\a");
```

```
        getch();
        mainmenu();
        return 0;
    }
    a.id=t;
    gotoxy(21,7);
    printf("Book Name:");gotoxy(33,7);
    scanf("%s",a.name);
    gotoxy(21,8);
    printf("Author:");gotoxy(30,8);
    scanf("%s",a.Author);
    gotoxy(21,9);
    printf("Quantity:");gotoxy(31,9);
    scanf("%d",&a.quantity);
    gotoxy(21,10);
    printf("Price:");gotoxy(28,10);
    scanf("%f",&a.Price);
    gotoxy(21,11);
    printf("Rack No:");gotoxy(30,11);
    scanf("%d",&a.rackno);
    return 1;
}
int checkid(int t) //check whether the book is exist in library or not
{
    rewind(fp);
    while(fread(&a,sizeof(a),1,fp)==1)
    if(a.id==t)
    return 0; //returns 0 if book exits
     return 1; //return 1 if it not
```

```
}
int t(void) //for time
{
time_t t;
time(&t);
printf("Date and time:%s\n",ctime(&t));

return 0 ;
}

void Password(void) //for password option
{

  system("cls");
  char d[25]="Password Protected";
  char ch,pass[10];
  int i=0,j;
  //textbackground(WHITE);
  //textcolor(RED);
   gotoxy(10,4);
   for(j=0;j<20;j++)
   {
   delay(50);
   printf("*");
   }
   for(j=0;j<20;j++)
  {
  delay(50);
  printf("%c",d[j]);
```

```
}
for(j=0;j<20;j++)
{
delay(50);
printf("*");
}
gotoxy(10,10);
gotoxy(15,7);
printf("Enter Password:");

while(ch!=13)
{
 ch=getch();

 if(ch!=13 && ch!=8){
 putch('*');
 pass[i] = ch;
 i++;
 }
}
pass[i] = '\0';
if(strcmp(pass,password)==0)
{

 gotoxy(15,9);
 //textcolor(BLINK);
 printf("Password match");
 gotoxy(17,10);
 printf("Press any key to countinue.....");
```

```
   getch();
   mainmenu();
  }
  else
  {
    gotoxy(15,16);
    printf("\aWarning!! Incorrect Password");
    getch();
    Password();
  }
}
void issuerecord() //display issued book's information
{
        system("cls");
        gotoxy(10,8);
        printf("The Book has taken by
        Mr. %s",a.stname); gotoxy(10,9);
        printf("Issued
        Date:%d-%d-%d",a.issued.dd,a.issued.mm,a.issued.yy);
        gotoxy(10,10);
        printf("Returning Date:%d-%d-%d",a.duedate.dd,a.duedate.mm,a.duedate.yy);
}
void loaderanim()
{
int loader;
system("cls");
gotoxy(20,10);
printf("LOADING.....");
printf("\n\n");
gotoxy(22,11);
```

```
for(loader=1;loader<20;loader++)
{
delay(100);printf("%c",219);}
}
//End of program


int i1=0,i2=0,i3=0,i4=0,i5=0,i6=0;
int top1=-1,top2=-1,top3=-1,top4=-1,top5=-1,top6=-1;
int g=0,b=0,c=0,d=0,e=0,f=0; char

toplist1[5][100],toplist2[5][100],toplist3[5][100],toplist4[5][100],toplist5[5][100],toplist6[5][
100];
float a1[5],a2[5],a3[5],a4[5],a5[5],a6[5];
char n1[5][100],n2[5][100],n3[5][100],n4[5][100],n5[5][100],n6[5][100];
void stumenu()
{
   system("cls");
   gotoxy(20,3);
printf("\xB2\xB2\xB2\xB2\xB2\xB2 LIBRARY MANAGEMENT SYSTEM
\xB2\xB2\xB2\xB2\xB2\xB2");
gotoxy(20,5);
printf("\xB2\xB2\xB2\xB2\xB2\xB2 YOU HAVE SUCCESFULLY LOGGED IN AS A STUDENT
\xB2\xB2\xB2\xB2\xB2\xB2");
gotoxy(20,6);
printf("\xB2\xB2\xB2\xB2\xB2\xB2 ENTER YOUR BRANCH \xB2\xB2\xB2\xB2\xB2\xB2");
gotoxy(20,8);
printf("\xB2\xB2\xB2\xB2\ 1=> Computer");
gotoxy(20,10);
printf("\xB2\xB2\xB2\xB2\ 2=> Electronics");
gotoxy(20,12);
```

```
printf("\xB2\xB2\xB2\xB2\ 3=> Electrical");
gotoxy(20,14);
printf("\xB2\xB2\xB2\xB2\ 4=> Civil");
gotoxy(20,16);
printf("\xB2\xB2\xB2\xB2\ 5=> Mechanical");
gotoxy(20,18);
printf("\xB2\xB2\xB2\xB2\ 6=> Architecture");
gotoxy(20,20);
printf("\xB2\xB2\xB2\xB2\ 7=> Back to main
menu"); gotoxy(20,21);
printf("\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2\xB2");
int ch;
scanf("%d",&ch);
switch(ch){
case 1:
  {
     system("cls");
     gotoxy(20,3);
     printf("\xB2\xB2\xB2\xB2\xB2\xB2 ENTER YOUR REVIEW ON A SCALE OF 1-10 AND
BOOK NAME \xB2\xB2\xB2\xB2\xB2\xB2\n");
     gotoxy(25,5);
     scanf("%s",&n1[i1]);
     gotoxy(25,6);
     scanf("%f",&a1[i1]);
     printf("\n");
     i1++;
     if(i1==10){
        printf("That's all folks");
```

```
        }



        system("cls");
        gotoxy(25,8);
        printf("YOU WISH TO CONTINUE? (1)YES OR (2)NO\n");
        gotoxy(25,10);
        int y;
        scanf("%d",&y);
        switch(y){
    case 1:
        stumenu();
        break;
    case 2:
        option();
        }
    }
    break;
case 2:
    {
         system("cls");
        gotoxy(20,3);
        printf("\xB2\xB2\xB2\xB2\xB2\xB2 ENTER YOUR REVIEW ON A SCALE OF 1-10 AND
BOOK NAME \xB2\xB2\xB2\xB2\xB2\xB2\n");
        gotoxy(25,5);
        scanf("%s",&n2[i2]);
        gotoxy(25,6);
                    scanf("%f",&a2[i2]);
        printf("\n");
```

```
      i2++;
      if(i2==10)
        {printf("That's all folks");
                              }



      system("cls");
      gotoxy(25,8);
       printf("YOU WISH TO CONTINUE? (1)YES OR (2)NO\n");
       gotoxy(25,10);
       int y;
      scanf("%d",&y);
      switch(y){
   case 1:
      stumenu();
      break;
   case 2:
      option();
      }
   }
   break;
case 3:
   {
       system("cls");
      gotoxy(20,3);
      printf("\xB2\xB2\xB2\xB2\xB2\xB2 ENTER YOUR REVIEW ON A SCALE OF 1-10 AND
BOOK NAME \xB2\xB2\xB2\xB2\xB2\xB2\n");
      gotoxy(25,5);
```

```
        scanf("%s",&n3[i3]);
        gotoxy(25,6);
                        scanf("%f",&a3[i3]);
        printf("\n");

        i3++;
        if(i3==10)
           {printf("That's all folks");
                                }


         system("cls");
         gotoxy(25,8);
         printf("YOU WISH TO CONTINUE? (1)YES OR (2)NO\n");
         gotoxy(25,10);
         int y;
        scanf("%d",&y);
        switch(y){
    case 1:
        stumenu();
        break;
    case 2:
        option();
        }
    }
    break;
case 4:
    {
         system("cls");
```

```
      gotoxy(20,3);
     printf("\xB2\xB2\xB2\xB2\xB2\xB2 ENTER YOUR REVIEW ON A SCALE OF 1-10
AND BOOK NAME \xB2\xB2\xB2\xB2\xB2\xB2\n");
     gotoxy(25,5);
      scanf("%s",&n4[i4]);
      gotoxy(25,6);
                  scanf("%f",&a4[i4]);
      printf("\n");

      i4++;
      if(i4==10)
         {printf("That's all folks");
                           }

       system("cls");
       gotoxy(25,8);
       printf("YOU WISH TO CONTINUE? (1)YES OR
       (2)NO\n"); gotoxy(25,10);
                   int y;
      scanf("%d",&y);
      switch(y){
   case 1:
      stumenu();
      break;
   case 2:
      option();
      }
   }
   break;
```

```
case 5:
   {
       system("cls");
      gotoxy(20,3);
     printf("\xB2\xB2\xB2\xB2\xB2\xB2 ENTER YOUR REVIEW ON A SCALE OF 1-10
AND BOOK NAME \xB2\xB2\xB2\xB2\xB2\xB2\n");
     gotoxy(25,5);
      scanf("%s",&n5[i5]);
      gotoxy(25,6);
                  scanf("%f",&a5[i5]);
      printf("\n");

      i5++;
      if(i5==10)
         {printf("That's all folks so chose another branch");
                           }

       system("cls");
       gotoxy(25,8);
       printf("YOU WISH TO CONTINUE? (1)YES OR (2)NO\n");
       gotoxy(25,10);
       int y;
      scanf("%d",&y);
      switch(y){
   case 1:
      stumenu();
      break;
   case 2:
      option();
```

```
      }
   }
   break;
case 6:
   {
       system("cls");
      gotoxy(20,3);
      printf("\xB2\xB2\xB2\xB2\xB2\xB2 ENTER YOUR REVIEW ON A SCALE OF 1-10 AND
BOOK NAME \xB2\xB2\xB2\xB2\xB2\xB2\n");
      gotoxy(25,5);
      scanf("%s",&n6[i6]);
      gotoxy(25,6);
                  scanf("%f",&a6[i6]);
      printf("\n");

      i6++;
      if(i6==10)
        {printf("That's all folks");
                              }

       system("cls");
       gotoxy(25,8);
       printf("YOU WISH TO CONTINUE? (1)YES OR (2)NO\n");
       gotoxy(25,10);
       int y;
      scanf("%d",&y);
      switch(y){
   case 1:
      stumenu();
```

```
        break;
    case 2:
        option();
        }
    }
    break;
case 7:
    stumenu();
}
}
void sort(float x[5],char st[5][100]){
int i,j;
float t;
char s[100];
for(i=0;i<4;i++){
    for(j=0;j<4-i;j++){
        if(x[j]>x[j+1])
        {
            t=x[j];
            x[j]=x[j+1];
            x[j+1]=t;
            strcpy(s,st[j]);
            strcpy(st[j],st[j+1]);
            strcpy(st[j+1],s);
        }

    }
}
```

```
}



void push1()
{
	if(top1==9 ){
		printf("Data limit exceeded");
	}
	else{
		strcpy(toplist1[++top1],n1[g++]);
	}
}
void push2()
{
	if(top2==9 ){
		printf("Data limit exceeded");
	}
	else{
		strcpy(toplist2[++top2],n2[b++]);
	}
}
void push3()
{
	if(top3==9 ){
		printf("Data limit exceeded");
	}
	else{
		strcpy(toplist3[++top3],n3[c++]);
	}
```

```
}
void push4()
{
	if(top4==9 ){
		printf("Data limit exceeded");
	}
	else{
		strcpy(toplist4[++top4],n4[d++]);
	}
}
void push5()
{
	if(top5==9 ){
		printf("Data limit exceeded");
	}
	else{
		strcpy(toplist5[++top5],n5[e++]);
	}
}
void push6()
{
	if(top6==9 ){
		printf("Data limit exceeded");
	}
	else{
		strcpy(toplist6[++top6],n6[f++]);
	}
}
void pop(char list[5][100],int limit){
```

```
	if(limit==-1){
		printf("List empty\n");
		exit(0);
	}
	char x[100];
	strcpy(x,list[limit]);
	printf("%s\n",x);

}
void displaybooks(int xy){
	int x1,x2,x3,x4,x5,x6,i;
	system("cls");
	gotoxy(25,5);
	switch(xy){
	case 1:
	{
			printf("COMPUTER BOOKS:\n");

		for(x1=top1,i=7;x1>=0;x1--,i+=2){
			gotoxy(25,i);
			pop(toplist1,x1);
		}
	}
	break;
	case 2:
		{

		printf("ELECTRONIC BOOKS:\n");
```

```
        for(x2=top2,i=7;x2>=0;x2--,i+=2){
                gotoxy(25,i);
                pop(toplist2,x2);
        }}
        break;
case 3:
        {

        printf("ELECTRICAL BOOKS:\n");

        for(x3=top3,i=7;x3>=0;x3--,i+=2){
                gotoxy(25,i);
                pop(toplist3,x3);
        }

        }
        break;
case 4:
        {

        printf("CIVIL BOOKS:\n");

        for(x4=top4,i=7;x4>=0;x4--,i+=2){
                gotoxy(25,i);
                pop(toplist4,x4);
        }}
        break;
case 5:
        {
```

```
                printf("MECHANICAL BOOKS:\n");

                for(x5=top5,i=7;x5>=0;x5--,i+=2){
                        gotoxy(25,i);
                        pop(toplist5,x5);
                }}
                break;
        case 6:
                {


                printf("ARCHITECTURE BOOKS:\n");

                for(x6=top6,i=7;x6>=0;x6--,i+=2){
                        gotoxy(25,i);
                        pop(toplist6,x6);
                }}}
}
void schoice(){
        int cho,i;
        system("cls");
        gotoxy(20,6);
printf("\xB2\xB2\xB2\xB2\xB2\xB2 ENTER YOUR BRANCH \xB2\xB2\xB2\xB2\xB2\xB2");
gotoxy(20,8);
printf("\xB2\xB2\xB2\xB2\ 1=> Computer");
gotoxy(20,10);
printf("\xB2\xB2\xB2\xB2\ 2=> Electronics");
gotoxy(20,12);
```

```
printf("\xB2\xB2\xB2\xB2\ 3=> Electrical");
gotoxy(20,14);
printf("\xB2\xB2\xB2\xB2\ 4=> Civil");
gotoxy(20,16);
printf("\xB2\xB2\xB2\xB2\ 5=> Mechanical");
gotoxy(20,18);
printf("\xB2\xB2\xB2\xB2\ 6=> Architecture");

        scanf("%d",&cho);
        switch(cho){
                case 1:
                {
                        sort(a1,n1);
                        for(i=0;i<5;i++){
                                        push1();
                        }
                        displaybooks(1);
                }
                        break;
                case 2:
                        {

                        sort(a2,n2);
                        for(i=0;i<5;i++){
                                        push2();
                        }
                        displaybooks(2);
                }
                        break;
```

```
case 3:
        {

        sort(a3,n3);
        for(i=0;i<5;i++){
                        push3();
        }
        displaybooks(3);
        }
        break;
case 4:
        {

        sort(a4,n4);
        for(i=0;i<5;i++){
                        push4();
        }
        displaybooks(4);
}
        break;
case 5:
        {

        sort(a5,n6);
        for(i=0;i<5;i++){
                        push5();
        }
        displaybooks(5);
}
```

```
                    break;
                case 6:
                    {


                    sort(a6,n6);
                    for(i=0;i<5;i++){
                                push6();
                    }
                    displaybooks(6);
                }
        }
}
```

REFERENCES

- Ian, H., Witten, Rodger J. McNab, Stefan J. Boddie and David Bainbridge, 2000. Greenstone: A comprehensive open-source digital library software system. Proceedings of the 5'b ACM Conference on Digital Libraries.
- Serge Abiteboul, 1997. Querying semi-structured data. Proceedings of the 6th International Conference on Database Theory.
- Sihem AmerYahia, Chavdar Botev and Jayavel, 2004. TeXQuery: A fulltext search extension to Xquery. Proceedings of the 13'h Conference on World Wide Web.
- Brian F., Cooper, Neal Sample, Michael J. Franklin, Gfsli R. Hjaltasonl and Moshe Shadmonl, 2001. A fast index for semistructured data. Proceedings of the 27'h VLDB Conference.
- Dirk Bahle Hugh E. and Williams Justin Zobel, 2002. Efficient phrase querying with an auxiliary index. Proceedings of the 25d' Annual International ACM SIGIR Conference on Research and Development in Information Retrieval.
- Dongwook Shin, Hyuncheol Jang and Honglan Jin, 1998. BUS: An effective indexing and retrieval scheme in structured documents. Proceedings of the third ACM Conference on Digital Libraries.

- Christian Bohm, Bernhard Braunmuller, HansPeter Kriegel and Matthias Schubert, 2000. Efficient similarity search in digital libraries. IEEE Advances in Digital Libraries (ADL).
- Gonzalo Navarro and Ricardo Yates, 1997. Proximal nodes: A model to query document databases by content and structure. ACM Transactions on Information Systems, 15: pp 401-435.
- David Bainbridge, John Thompson and Ian H. Witten, 2003. Assembling and enriching digital library collections. Proceedings of the 3rd ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL).
- Ian, H., Witten, David Bainbridge and Stefan J. Boddie, 2001. Power to the people: End-userbuilding of digital library collections. Proceedings of the 15' ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL).
- Anoop Kumar, Ranjani Saigal, Robert Chavez and Nikolai Schwertner, 2004. Architecting an extensible Digital Repository. Proceedings of the 2004 Joint ACM/IEEE Conference on Digital Libraries (JCDL).
- Jason McHugh, Serge Abiteboul, Roy Goldman, Dallan Quass and Jennifer Widom, 1997. Lore: A database management system for semi-structured data. SIGMOD Record, 26: pp 54-66.
- Naomi Dushay, James C. French and Carl Lagoze, 1999. A characterization study of NCSTRL distributed searching. Technical Report: TR991725, Cornell University Ithaca, NY, USA.
- Li Guilin, Li Jianzhong and Yang Yan, 2003. Information extraction from PDF using plug-in. Comp. Appl., 23: pp 110-112.
- Gu Xianrui, Li Jianzhong and Yang Yan, 2002. Parallel document data loading algorithm in digital library. Comp. Sci., 29: pp 104-106.
- Ren Meirui, Li Jianzhong and Yang Yan, 2002. The implementation of an automated text categorization system based on Naive Bayes method. Comp. Sci., 29: pp 285-287.
- Yang Yan and Li Jianzhong, 2003. Cluster-based data allocation method of web servers in digital library. Comp. Eng. Appl. pp 39: 38-41.
- Yang yan and Li Jianzhong, 2005. Interest-based recommendation in digital library. J. Comp. Sci., 1: pp 40-46.
- Yang yan and Li Jianzhong, 2004. Topology-based paper recommendation in digital library. J.
- F.Cirillo, A.Cozzolino, M.De Santo, M. Marsella, S.Salerno,(2000), A metadata based distance learning platform, IEEE Systems, Man, and Cybernetics, vol. 1, pp. 44 -48.
- Brian F., Cooper, Neal Sample, Michael J. Franklin, Gfsli R. Hjaltasonl and Moshe Shadmonl, 2001