

**PROJECT REPORT**  
**On**

**“LIBRARY MANAGEMENT SYSTEM”**

Submitted by,

**Sudhir Baliram Kadam**

**July Batch 2023**

IN PARTIAL FULFILMENT OF  
**JAVA FULL STACK COURSE**  
**GIRIS TECH HUB, PUNE**  
BACTH JULY 2023

## INDEX

Sr. No.	Title of Chapter	Page No.
1	<b>Introduction</b>	4
1.1	Project Aims and Objectives	4
1.2	Background of Project	5
1.3	Operation Environment	6
2	<b>System Analysis</b>	7
2.1	Software Requirement Specification	7
2.2	Existing vs. Proposed	8
2.3	Software Tools Used	9
3	<b>System Design</b>	10
3.1	Data Flow Diagram	10
4	<b>System Implementation</b>	11
4.1	Module Description	11
4.2	Screen Shots	12
5	<b>System Testing</b>	23
5.1	Unit Testing	23
5.2	Integration Testing	24
6	<b>Conclusion and Future Scope</b>	25
7	<b>References</b>	26

## **ABSTRACT**

Library management system is a project which aims in developing a computerized system to maintain all the daily work of library .This project has many features which are generally not available in normal library management systems like facility of user login. It has also a facility where student after logging in their accounts can see list of books issued and its issue date and return date. Overall this project of ours is being developed to help the students as well as staff of library to maintain the library in the best way possible and also reduce the human efforts.

# CHAPTER 1

## INTRODUCTION



### 1.1 PROJECT AIMS AND OBJECTIVES.

- Online book issue
- Student login page where student can find books issued by him/her and date of return.
- Search books, view all books
- This library management system is used to reduce the paper work, man power and stationery.
- This system makes staff of the library to work efficiently with library management.

## **1.2 BACKGROUND OF PROJECT**

Library Management System is an application which refers to library systems which are generally small or medium in size. It is used by librarian to manage the library using a computerized system where he/she can record various transactions like issue of books, return of books, addition of new books, addition of new students etc.

Books and student maintenance modules are also included in this system which would keep track of the students using the library and also a detailed description about the books a library contains. With this computerized system there will be no loss of book record or member record which generally happens when a non-computerized system is used.

All these modules are able to help librarian to manage the library with more convenience and in a more efficient way as compared to library systems which are not computerized.

## 1.3 OPERATION ENVIRONMENT

PROCESSOR	INTEL CORE PROCESSOR OR BETTER PERFORMANCE
OPERATING SYSTEM	WINDOWS 10
MEMORY	1 GB RAM OR MORE
DATE BASE	TEXT FILES

## **CHAPTER 2**

### **SYSTEM ANALYSIS**

#### **2.1 SOFTWARE REQUIREMENT SPECIFICATION**

##### **2.1.1 GENERAL DESCRIPTION**

###### **PRODUCT DESCRIPTION:**

Library Management System is a computerized system which helps user(librarian) to manage the library daily activity in electronic format. It reduces the risk of paper work such as file lost, file damaged and time consuming. It can help user to manage the transaction or record more effectively and time-saving.

###### **ROBLEM STATEMENT:**

The problem occurred before having computerized system includes:

- File lost

When computerized system is not implemented file is always lost because of human environment. Sometimes due to some human error there may be a loss of records.

- File damaged

When a computerized system is not there file is always lost due to some accident like spilling of water by some member on file accidentally. Besides some natural disaster like floods or fires may also damage the file

- Difficult to search record

When there is no computerized system there is always a difficulty in searching of records if the records are large in number .

- Space consuming

After the number of records become large the space for physical storage of file and records also increases if no computerized system is implemented.

##### **2.1.2 SYSTEM OBJECTIVES**

- Improvement in control and performance

The system is developed to cope up with the current issues and problems of library. The system can add user, validate user and is also bug free.

- Save cost

After computerized system is implemented less human force will be required to maintain the library thus reducing the overall cost

- Save time

Librarian is able to search record by using few clicks of mouse and few search keywords thus saving his valuable time.

##### **2.1.3 SYSTEM REQUIREMENTS**

###### **2.1.3.1 NON FUNCTIONAL REQUIREMENTS**

- **EFFICIENCY REQUIREMENT**

When a library management system will be implemented librarian and user will easily access library as searching and book transaction will be very faster .

- **RELIABILITY REQUIREMENT**

The system should accurately performs member login and search books and keep track of issued books.

- **USABILITY REQUIREMENT**

The system is designed for a user friendly environment so that student and staff of library can perform the various tasks easily and in an effective way.

### 2.1.3.2 FUNCTIONAL REQUIREMENTS

- **USER LOGIN**

This feature used by the user to login into system. They are required to enter user id and password before they are allowed to enter the system .The user id and password will be verified and if invalid id is there user is allowed to not enter the system.

- **MAINMENU**

This function is used to display the main menu of this project.

- **ADD BOOKD**

This function adds books in a file. For that, you need to mention the department to which you want to add the book. The record is saved in a file. And, it is similar for the following functions as well.

- **DELETE BOOKS**

This function can be used to delete the book from the file.

- **EDIT BOOKS**

This function can be used to edit the existing book.

- **SEARCH BOOK**

This function can be used to search particular book from file. .

- **VIEW BOOKS**

This function can be used for to display book info.

- **ISSUE RECORD**

With this function, you can keep record of the student to whom the book has been issued.

- **CLOSEAPPLICATION**

This function is for closing the application.

## 2.2 EXISTING VS PROPOSED SYSTEM

- Existing system does not have any facility of teachers login or student login where as proposed system will have a facility of student login as well as teacher's login
- Existing system does not have a facility of online reservation of books whereas proposed system has a facility of online reservation of books
- Existing system does not have any facility of online notice board where description of workshops happening in our college as well as nearby colleges is being provided.



- Existing system does not have any facility to generate student reports as well book issue reports whereas proposed system provides librarian with a tool to generate reports
- Existing system does not have any facility for book request and suggestions where as in proposed system after logging in to their accounts student can request books as well as provide suggestions to improve library.

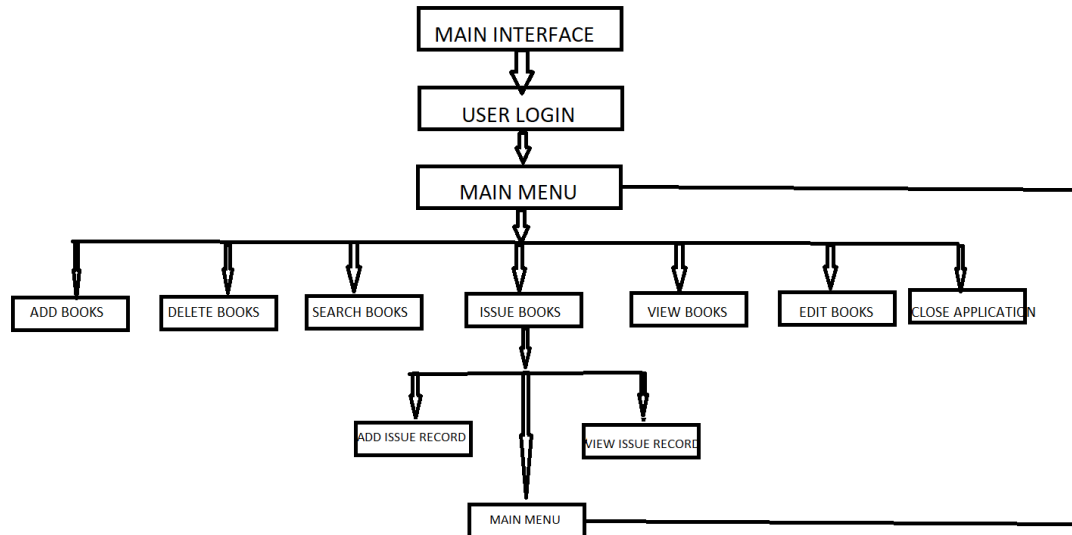
## **2.3 SOFTWARE TOOLS USED**

- This is open source project.
- Dev cpp code editor.
- In this project c programming language is used for development of project.
- File handling is used to store data .

## CHAPTER 3

### SYSTEM DESIGN

#### 3.1 DATA FLOW DIAGRAM



## CHAPTER 4

### SYSTEM IMPLEMENTAION

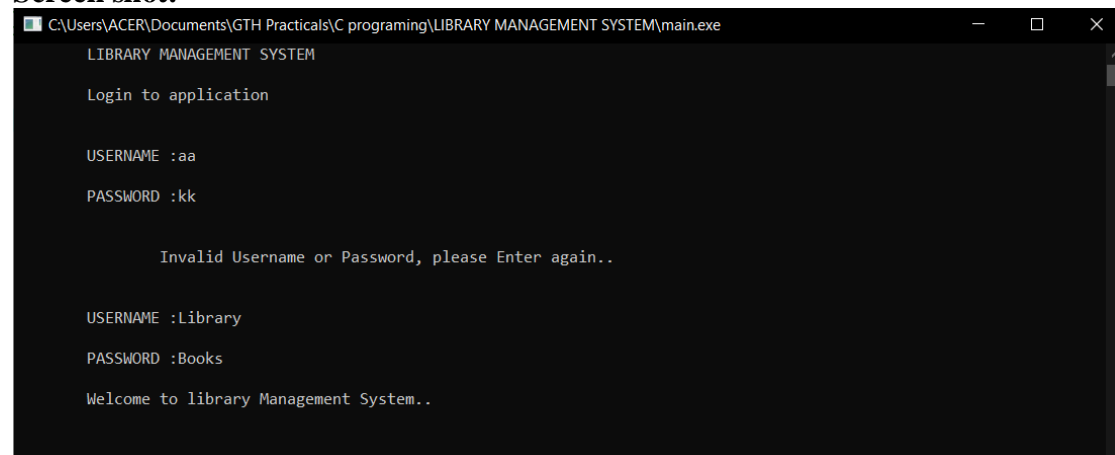
#### 4.1. MODULE DESCRIPTION

##### 4.1.1 LOGIN MODULE

function-void password()

In this module user login designed with username and password user can login into the application.

**Screen shot:**



##### CODE FOR LOGIN MODULE:

```
//#include "Loginmodel.h";
#include "Loginmodel.h"
#include "mainapplication.h"
void password(){
loginlab();
int cnt=1;
char username[30];
char password[30];
fp=fopen("C:\\Users\\ACER\\Documents\\GTH Practicals\\C programing\\LIBRARY
MANAGEMENT SYSTEM\\login.txt","r");
if(fp==NULL){
    printf("\tFile path not found..!");
}
printf("\tLIBRARY MANAGEMENT SYSTEM\n\n");
printf("\tLogin to application\n\n");
do{
    printf("\n\tUSERNAME :");
    scanf("%s",username);
    printf("\n\tPASSWORD :");
    scanf("%s",password);
    if((!strcmp(username,l.user))&&(!strcmp(password,l.pass))){
        mainmenu();
    }else{
```

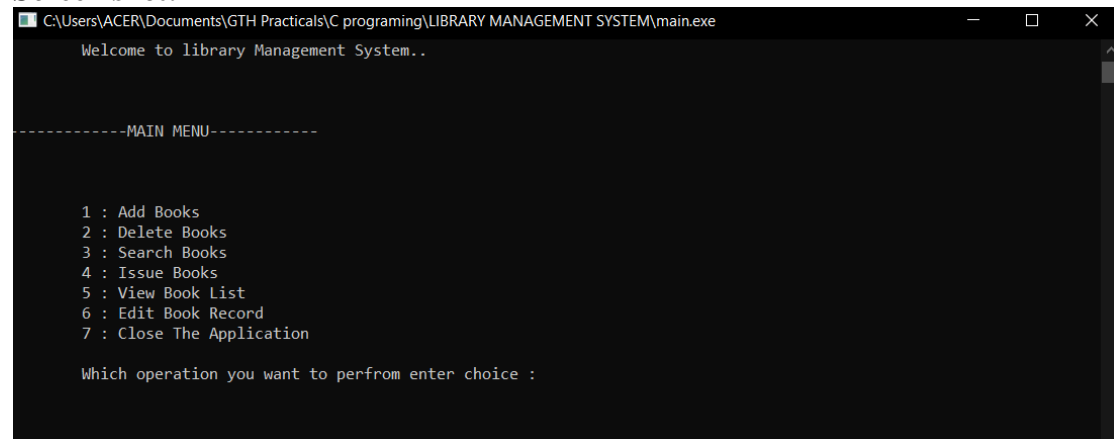
```
printf("\n\n\tInvalid Username or Password, please Enter  
again..\n\n");  
    cnt++;  
}  
} while(cnt<=3);  
if(cnt>3){  
    printf("\t\tLogin Faild..Unknown user..!\n");  
    exit(1);  
}  
fclose(fp);  
}
```

#### 4.1.2 MAINMENU MODULE

function- void mainmenu()

This module is used to show the main menu the library management system.

##### Screen shot:



##### CODE FOR MAINMENU MODULE:

```
//#include<stdio.h>  
#include<string.h>  
#include "bookmodel.h"  
#include "filepath.h"  
#include "Searchbook.h"  
#include "viewbooks.h"  
#include "addBook.h"  
//#include "Editbooks.h"  
#include "Editbookrecords.h"  
#include "Deletebooks.h"  
#include "issueBooksmodel.h"  
#include "issuerecords.h"  
#include "CloseApplication.h"  
void mainmenu(){ //this is main menu of the library management system  
    int choice;  
    do{ printf("\n\n\n-----MAIN MENU-----\n\n\n");  
        printf("\n\t1 : Add Books");  
        printf("\n\t2 : Delete Books");  
        printf("\n\t3 : Search Books");  
        printf("\n\t4 : Issue Books");  
        printf("\n\t5 : View Book List");
```

```
printf("\n\t6 : Edit Book Record");
printf("\n\t7 : Close The Application");
printf("\n\n\tWhich operation you want to perform enter choice :");
scanf("%d",&choice);
switch(choice){           // choice to that we want to perform
    case 1:
        addbook(b);
        break;
    case 2:
        deletebooks();
        break;
    case 3:
        searchbooks(b);
        break;
    case 4:
        issuebooks();
        break;
    case 5:
        viewbooks();
        break;
    case 6:
        editbook();
        break;
    case 7:
        closeapplication();
        break;
}

}while(choice!=8);

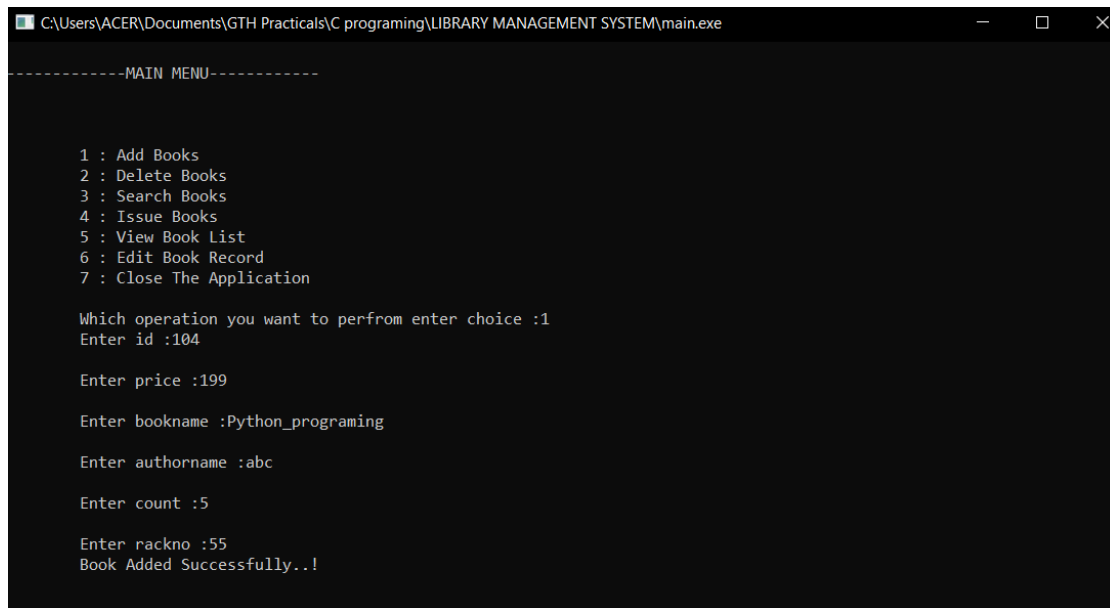
}
```

#### **4.1.3. ADD BOOK MODULE**

function- void addbooks()

This module is used to add books in the file. Book has different fields like id, book name, author name, price, Quantity, rackno, etc.

**Screen shot:**



```
C:\Users\ACER\Documents\GTH Practicals\C programing\LIBRARY MANAGEMENT SYSTEM\main.exe

-----MAIN MENU-----

1 : Add Books
2 : Delete Books
3 : Search Books
4 : Issue Books
5 : View Book List
6 : Edit Book Record
7 : Close The Application

Which operation you want to perfrom enter choice :1
Enter id :104

Enter price :199

Enter bookname :Python_programing

Enter authorname :abc

Enter count :5

Enter rackno :55
Book Added Successfully..!
```

### CODE FOR ADD BOOK MODULE:

```
//this function is used to add or write book records in text file
void addbook(struct Book b){
    fptr=fopen("C:\\Users\\ACER\\Documents\\GTH Practicals\\C
programing\\LIBRARY MANAGEMENT SYSTEM\\demo.txt","a");
    if(fptr==NULL){
        printf("\tFile path not found..!");
    }else{
        printf("\tEnter id :");
        scanf("%d",&b.id);
        printf("\n\tEnter price :");
        scanf("%d",&b.price);
        printf("\n\tEnter bookname :");
        scanf("%s",&b.name);
        printf("\n\tEnter authorname :");
        scanf("%s",&b.author);
        printf("\n\tEnter count :");
        scanf("%d",&b.count);
        printf("\n\tEnter rackno :");
        scanf("%d",&b.rackno);
        //write all data in text file

        fprintf(fptr,"%d\t%d\t%s\t%s\t%d\t%d",b.id,b.price,b.name,b.author,b.count,b.r
ackno);

        fputs("\n",fptr);
        printf("\tBook Added Successfully..!");
        fclose(fptr);
    }
}
```

### 4.1.4. DELETE BOOK MODULE

function- void deletebooks()

This module is designed to delete the specific book record by book id.

### Screen shot:

```
-----MAIN MENU-----
1 : Add Books
2 : Delete Books
3 : Search Books
4 : Issue Books
5 : View Book List
6 : Edit Book Record
7 : Close The Application

Which operation you want to perform enter choice :2
Enter the id to delete book record :104
Deleted record successfully..!
```

### CODE FOR DELETE MODULE

```
void deletebooks(){
    FILE *fptr1;
    int bid;
    char str[256];
    fptr=fopen("C:\\Users\\ACER\\Documents\\GTH Practicals\\C
programming\\LIBRARY MANAGEMENT SYSTEM\\demo.txt","r");
    if(fptr==NULL){
        printf("\tRecords not found..");
    }
    //createing temp file to store book data
    fptr1=fopen("C:\\Users\\ACER\\Documents\\GTH Practicals\\C
programming\\LIBRARY MANAGEMENT SYSTEM\\temp.txt","w");
    if(fptr==NULL){
        printf("\tRecords not found..");
    }
    printf("\tEnter the id to delete book record :");
    scanf("%d",&bid);
    //this loop read all records until eof

    while(fscanf(fptr,"%d%d%s%s%d%d",&b.id,&b.price,&b.name,&b.author,&b.count,
    &b.rackno)!=-1){
        if(bid!=b.id){ //this condition ture until the book id is not found
            //this line of code is used to write all records in temp file excpet
            matched book id

            fprintf(fptr1,"%d\t%d\t%s\t%s\t%d\t%d",b.id,b.price,b.name,b.author,b.count,b.
            rackno);

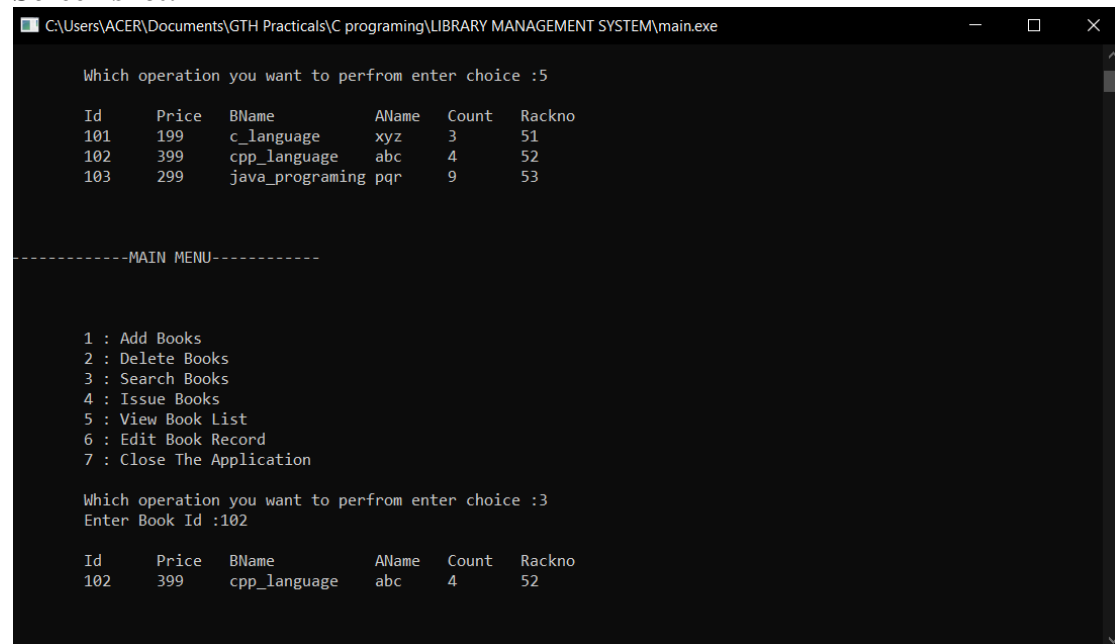
            fputs("\n",fptr1);
        }
    }
    fclose(fptr);
    fclose(fptr1);
    remove("demo.txt"); //remove original file
    rename("temp.txt","demo.txt"); //rename the temp file to original file
    printf("\tDeleted record successfully..!");
}
```

#### 4.1.5. SEARCH MODULE

function – void searchbooks()

This module is used to search a book record by the book id.

**Screen shot:**



#### CODE FOR SEARCH MODULE

```
void searchbooks(struct Book b){
    int bookid,flag;
    fptr=fopen("C:\\Users\\ACER\\Documents\\GTH Practicals\\C
programing\\LIBRARY MANAGEMENT SYSTEM\\demo.txt","r");
    if(fptr==NULL){
        printf("\tRecords not found..");
    }else{
        flag=0;
        printf("\tEnter Book Id :"); //searching books by their book id
        scanf("%d",&bookid);
        //this loop will read all records until end of file

        while(fscanf(fptr,"%d%d%s%s%d%d",&b.id,&b.price,&b.name,&b.author,&b.count,
        &b.rackno)!=-1){
            if(b.id==bookid) //if book found set flag to 1 and break loop
            {
                flag=1;
                break;
            }
        }

        if(flag){ //if flag is 1 then show list of issued books

            printf("\n\tId\tPrice\tBName\t\tAName\tCount\tRackno\n");

            printf("\t%d\t%d\t%s\t%s\t%d\t%d\n",b.id,b.price,b.name,b.author,b.count,b.rac
```



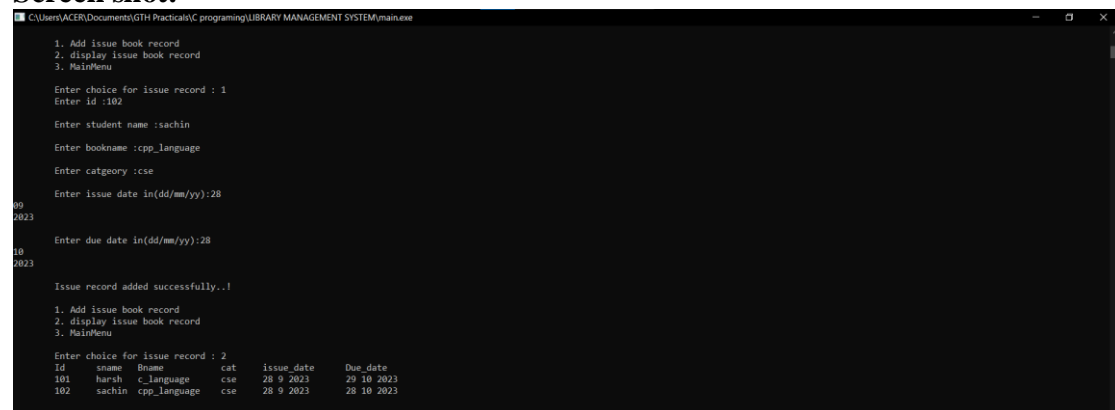
```
kno);
    }else{
        printf("\n\tBook not found..");
    }
    fclose(fp);
}
```

### 4.1.6. ISSUE BOOK MODULE

function- void issuebooks()

This module is used to keep records of issued books with fields like book id, book name, student name, category, issue date and due date etc. in this we have two choices add issue record and view issue record.

#### Screen shot:



```
C:\Users\ACER\Documents\GTH Practicals\C programming\LIBRARY MANAGEMENT SYSTEM\main.exe
1. Add issue book record
2. display issue book record
3. MainMenu
Enter choice for issue record : 1
Enter id :102
Enter student name :sachin
Enter bookname :cpp_language
Enter category :cse
Enter issue date in(dd/mm/yy):28/09/2023
Enter due date in(dd/mm/yy):28/10/2023
Issue record added successfully..!
1. Add issue book record
2. display issue book record
3. MainMenu
Enter choice for issue record : 2
ID      Sname  Bname   cat    issue_date    Due_date
101     harsh   c_language  cse     28 9 2023     29 10 2023
102     sachin  cpp_language cse     28 9 2023     28 10 2023
```

### CODE FOR ISSUE BOOK MODULE

```
#include<time.h>
//created structure to store data about issue book records
struct IssueBooks{
    int id;
    char stname[30];
    char name[30];
    char cat[30];
    struct tm issue;
    struct tm duedate;
};
struct IssueBooks ib;

void issuebooks(){ //this function is used to keep record of isseued books
//    time_t t;
//    t = time(NULL);
int choice;
do{
    printf("\n\n\t1. Add issue book record");
    printf("\n\t2. display issue book record");
    printf("\n\t3. MainMenu");
    printf("\n\n\tEnter choice for issue record : ");
    scanf("%d",&choice);
    switch(choice){
        case 1: //here we can add issue book record
```

```
        fptr2=fopen("C:\\Users\\ACER\\Documents\\GTH Practicals\\C
programming\\LIBRARY MANAGEMENT SYSTEM\\issuerecord.txt","a");
        if(fptr2==NULL){
            printf("\tFile path not found..!");
        }else{
            printf("\tEnter id :");
            scanf("%d",&ib.id);
            printf("\n\tEnter student name :");
            scanf("%s",&ib.stname);
            printf("\n\tEnter bookname :");
            scanf("%s",&ib.name);
            printf("\n\tEnter catgeory :");
            scanf("%s",&ib.cat);
            printf("\n\tEnter issue date in(dd/mm/yy):");

            scanf("%d%d%d",&ib.issue.tm_mday,&ib.issue.tm_mon,&ib.issue.tm_year);
            printf("\n\tEnter due date in(dd/mm/yy):");

            scanf("%d%d%d",&ib.duedate.tm_mday,&ib.duedate.tm_mon,&ib.duedate.tm_
year);

            //writing all reocords in file
            fprintf(fptr2,"%d\t%s\t%s\t%s\t%d %d %d\t%d
%d %d",ib.id,ib.stname,ib.name,ib.cat,

            ib.issue.tm_mday,ib.issue.tm_mon,ib.issue.tm_year,ib.duedate.tm_mday,ib.due
date.tm_mon,ib.duedate.tm_year);

            fputs("\n",fptr2);
            printf("\n\tIssue record added successfully..!");
            fclose(fptr2);
            //fflush(stdin);
        }

        break;
    case 2:
        // this case is used to show list of issued book their issue dates and due
        dates

        fptr2=fopen("C:\\Users\\ACER\\Documents\\GTH
Practicals\\C programing\\LIBRARY MANAGEMENT
SYSTEM\\issuerecord.txt","r");
        if(fptr2==NULL){
            printf("\tFile path not found..!");
        }else{

            printf("\tId\tstname\tBname\ttcat\ttissue_date\ttDue_date\n");
            //this loop fetch all records form issue book list

            while(fscanf(fptr2,"%d%s%s%s%d%d%d%d",&ib.id,&ib.stname,&ib.na
me,&ib.cat,&ib.issue.tm_mday,&ib.issue.tm_mon,&ib.issue.tm_year,&ib.duedate.tm_
mday,&ib.duedate.tm_mon,&ib.duedate.tm_year)!=-1){
                printf("\t%d\t%s\t%s\t%s\t%d %d %d\t%d %d
%d %d
```

```
%d\n",ib.id,ib.stname,ib.name,ib.cat,ib.issue.tm_mday,ib.issue.tm_mon,ib.issue.tm_yea
r,ib.duedate.tm_mday,ib.duedate.tm_mon,ib.duedate.tm_year);
    }
    fclose(fp2);
}

    break;
case 3:
    mainmenu();
    break;
}
}while(choice!=4);

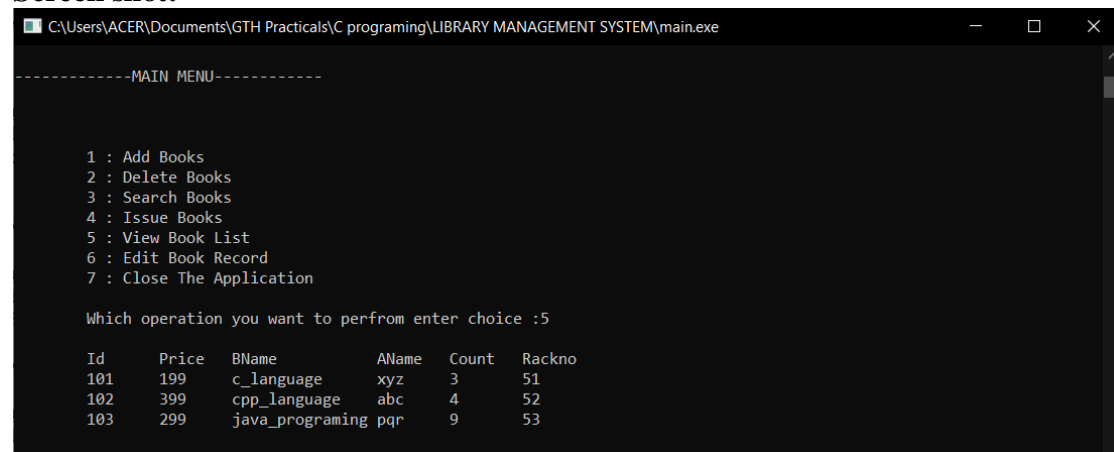
}
```

#### 4.1.7 VIEW BOOK MODULE

function – void viewbooks()

This module is used to show all records list.

**Screen shot:**



```
C:\Users\ACER\Documents\GTH Practicals\C programing\LIBRARY MANAGEMENT SYSTEM\main.exe
-----MAIN MENU-----

1 : Add Books
2 : Delete Books
3 : Search Books
4 : Issue Books
5 : View Book List
6 : Edit Book Record
7 : Close The Application

Which operation you want to perfrom enter choice :5

Id      Price  BName      AName    Count  Rackno
101     199    c_language  xyz       3      51
102     399    cpp_language abc        4      52
103     299    java_programing pqr       9      53
```

#### CODE FOR VIEW MODULE

```
void viewbooks(){ //this function is used to get the list of all records
    fptr=fopen("C:\\Users\\ACER\\Documents\\GTH Practicals\\C
programming\\LIBRARY MANAGEMENT SYSTEM\\demo.txt","r");
    if(fptr==NULL){ //if file is not found then show msg;
        printf("\tRecords not found..");
    }
    else
    {   printf("\n\tId\tPrice\tBName\tAName\tCount\tRackno\n");
        //this loop read all records until eof

        while(fscanf(fptr,"%d%d%s%s%d",&b.id,&b.price,&b.name,&b.author,&b.
count,&b.rackno)!=-1){

            //printing all records

            printf("\t%d\t%d\t%s\t%s\t%d\t%d\n",b.id,b.price,b.name,b.author,b.count,b.rackno);

        }
    }
}
```

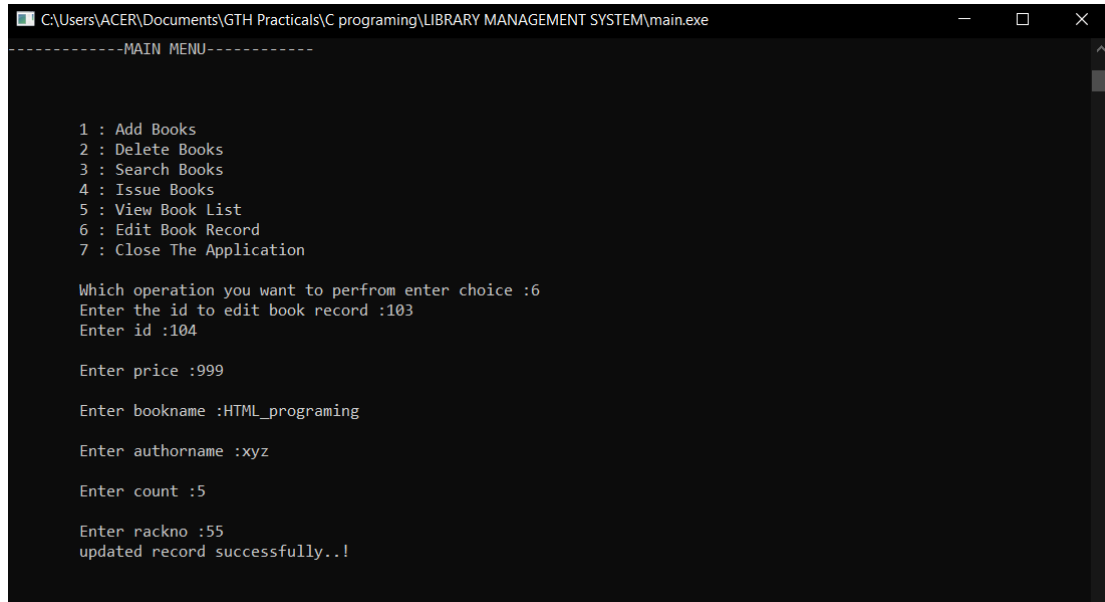
```
        fclose(fptr);    //closing file pointer
    }
}
```

#### 4.1.8. EDIT BOOK MODULE

function- void editbooks()

This module is used to edit the records by the book id. All the data related to book is edited in this module.

**Screen shot:**

A screenshot of a Windows command prompt window titled "C:\Users\ACER\Documents\GTH Practicals\C programing\LIBRARY MANAGEMENT SYSTEM\main.exe". The window displays the "MAIN MENU" with a list of options: 1: Add Books, 2: Delete Books, 3: Search Books, 4: Issue Books, 5: View Book List, 6: Edit Book Record, and 7: Close The Application. The user has selected option 6. The program then prompts for the operation choice (6), the book ID to edit (103), and the new ID (104). It then prompts for the price (999), book name (HTML\_programing), author name (xyz), count (5), and rack number (55). Finally, it displays "updated record successfully..!".

```
C:\Users\ACER\Documents\GTH Practicals\C programing\LIBRARY MANAGEMENT SYSTEM\main.exe
-----MAIN MENU-----

1 : Add Books
2 : Delete Books
3 : Search Books
4 : Issue Books
5 : View Book List
6 : Edit Book Record
7 : Close The Application

Which operation you want to perfrom enter choice :6
Enter the id to edit book record :103
Enter id :104

Enter price :999

Enter bookname :HTML_programing

Enter authorname :xyz

Enter count :5

Enter rackno :55
updated record successfully..!
```

#### CODE FOR EDIT BOOK MODULE

```
void editbook(){
    FILE *fptr1;
    int bid;
    char str[256];
    fptr=fopen("C:\\Users\\ACER\\Documents\\GTH Practicals\\C
programing\\LIBRARY MANAGEMENT SYSTEM\\demo.txt","r");
    if(fptr==NULL){
        printf("\tRecords not found..");
    }
    //createing temp file to store book data
    fptr1=fopen("C:\\Users\\ACER\\Documents\\GTH Practicals\\C
programing\\LIBRARY MANAGEMENT SYSTEM\\temp.txt","w");
    if(fptr==NULL){
        printf("\tRecords not found..");
    }
    printf("\tEnter the id to edit book record :");
    scanf("%d",&bid);
    //this loop read all records until eof

while(fscanf(fptr,"%d%d%s%s%d%d",&b.id,&b.price,&b.name,&b.author,&b.count,
&b.rackno)!=-1){

        if(bid!=b.id){ //this condition ture until the book id is not found
```

```

//this line of code is used to write all records in temp file
excpet matched book id

    fprintf(fptr1,"%d\t%d\t%s\t%s\t%d\t%d",b.id,b.price,b.name,b.author,b.count,b.
rackno);

        fputs("\n",fptr1);
    }else{
        //this block execute only when the book id matched
which is we are editiong
        printf("\tEnter id :");
        scanf("%d",&b.id);
        printf("\n\tEnter price :");
        scanf("%d",&b.price);
        printf("\n\tEnter bookname :");
        scanf("%s",&b.name);
        printf("\n\tEnter authorname :");
        scanf("%s",&b.author);
        printf("\n\tEnter count :");
        scanf("%d",&b.count);
        printf("\n\tEnter rackno :");
        scanf("%d",&b.rackno);
        //write all updated data in text file

        fprintf(fptr1,"%d\t%d\t%s\t%s\t%d\t%d",b.id,b.price,b.name,b.author,b.count,b.
rackno);

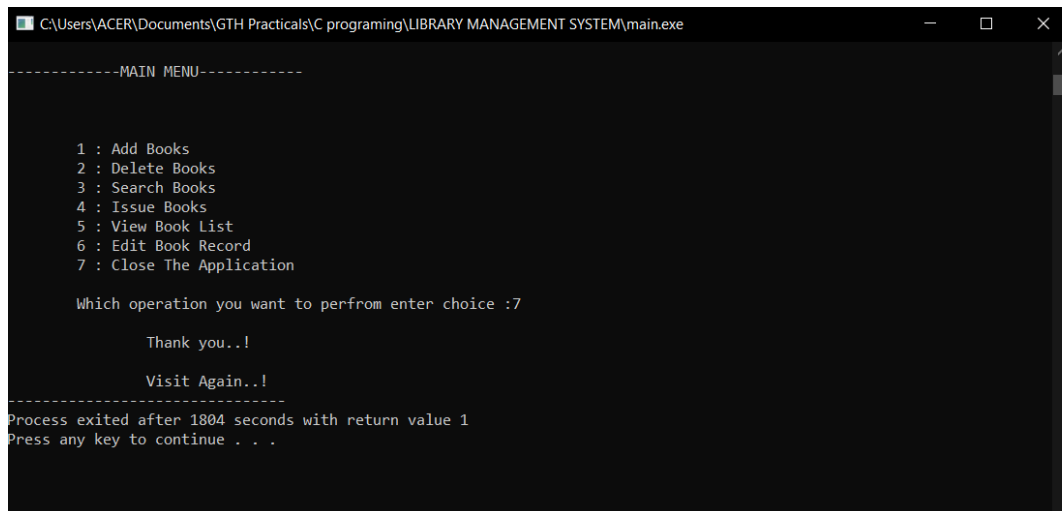
        fputs("\n",fptr1);
    }
}
fclose(fptr);
fclose(fptr1);
remove("demo.txt");
rename("temp.txt","demo.txt");
printf("\tupdated record successfully..!");
}
```

#### 4.1.9. CLOSE APPLICATION MODULE

function- void closeAppliction()

This module is used for close the application exit() function is used to close the application.

**Screen shot:**



```
C:\Users\ACER\Documents\GTH Practicals\C programming\LIBRARY MANAGEMENT SYSTEM\main.exe

-----MAIN MENU-----

1 : Add Books
2 : Delete Books
3 : Search Books
4 : Issue Books
5 : View Book List
6 : Edit Book Record
7 : Close The Application

Which operation you want to perform enter choice :7

Thank you..!

Visit Again..!

-----
Process exited after 1804 seconds with return value 1
Press any key to continue . . .
```

### CODE FOR CLOSEAPPLICATION MODULE

```
void closeapplication(){ //this function is used to close the application
    printf("\n\t\tThank you..!\n");
    printf("\n\t\tVisit Again..!");
    exit(1);
}
```

## CHAPTER 5

# SYSTEM TESTING

### 5.1 UNIT TESTING

Unit testing is undertaken when a module has been created and successfully reviewed. In order to test a single module we need to provide a complete environment i.e. besides the module we would require

- The procedures belonging to other modules that the module under test calls
- A procedure to call the functions of the module under test with appropriate parameters

Unit testing was done on each and every module that is described under module description of chapter

Testing main module:

it will start the application and call login module

Testing login module:

function-void password()

In this module user login designed with username and password user can login into the application.

Testing mainmenu module:

function- void mainmenu()

This module is used to show the main menu the library management system.

Testing add book module:

function- void addbooks()

This module is used to add books in the file. Book has different fields like id, book name, author name, price, Quantity, rackno, etc.

Testing delete book module:

function- void deletebooks()

This module is designed to delete the specific book record by book id.

Testing search book module:

function – void searchbooks()

This module is used to search a book record by the book id.

Testing issue book record module:

function- void issuebooks()

This module is used to keep records of issued books with fields like book id, book name, student name, category, issue date and due date etc. in this we have two choices add issue record and view issue record.

Testing view books module:

function – void viewbooks()

This module is used to show all records list.

Testing edit book module:

function- void editbooks()

This module is used to edit the records by the book id. All the data related to book is

edited in this module.

Testing close application module:

function- void closeAppliction()

This module is used for close the application exit() function is used to close the application

## **5.2 INTEGRATION TESTING**

The primary objective is to test the module interfaces in order to ensure that no errors are occurring when one module invokes the other module.



## **CHAPTER 6**

### **CONCLUSION AND FUTURE SCOPE**

This library management system provides a computerized version of library management which will benefit the students as well as the staff of the library. It makes entire process online where all data of books is stored digitally. It also has a login facility so its secure unauthorized users cannot use it.

There is a future scope of this facility that many more features can be added, such as books record can be stored as department wise, books can be search by book name, etc.

There are drawbacks using file handling so we can develop this library management system in other programming language (ex. Java, Python, etc.) and use database to keep book records in backend thus making it more interactive more users friendly and project which fulfills each users need in the best way possible.

## **CHAPTER 7**

### **REFERENCES**

- [GeekforGeeks-file handling](#)
- [Javatpoint- file handling](#)
- [W3resourse](#)