

A PROJECT REPORT
on
“LIBRARY MANAGEMENT SYSTEM”

Submitted to
KIIT UNIVERSITY

In Partial Fulfillment of the Requirement for the Award of

BACHELOR’S DEGREE IN COMPUTER
SCIENCE & ENGINEERING

BY

ANKAN MUKHERJEE	ROLL-1828049
ANUBHAV MISHRA	ROLL-1828054
ASHUTOSH SHARDA	ROLL-1828056
AYUSH PANDEY	ROLL-1828059
AYUSH JAISWAL	ROLL-1828060
DEVTANU MAJUMDAR	ROLL-1828065
HARSH CHANDRA JHA	ROLL-1828069
KESHAV KUMAR	ROLL-1828075
KUMAR SHUBHAM	ROLL-1828077



UNDER THE GUIDANCE OF PROF.
MAHENDRA KUMAR GOURISARIA

SCHOOL OF COMPUTER ENGINEERING
KALINGA INSTITUTE OF INDUSTRIAL TECHNOLOGY
BHUBANESWAE, ODISHA -751024

April 2019

KIIT Deemed to be University

School of Computer Engineering

Bhubaneswar, ODISHA 751024



CERTIFICATE

This is certify that the project entitled
**“LIBRARY MANAGEMENT
 SYSTEM”** submitted by

ANKAN MUKHERJEE	ROLL-1828049
ANUBHAV MISHRA	ROLL-1828054
ASHUTOSH SHARDA	ROLL-1828056
AYUSH PANDEY	ROLL-1828059
AYUSH JAISWAL	ROLL-1828060
DEVTANU MAJUMDAR	ROLL-1828065
HARSH CHANDRA JHA	ROLL-1828069
KESHAV KUMAR	ROLL-1828075
KUMAR SHUBHAM	ROLL-1828077

is a record of bonafide work carried out by them, in the partial fulfilment of the requirement for the award of Degree of Bachelor of Engineering (Computer Sci- ence & Engineering OR Information Technology) at KIIT Deemed to be university, Bhubaneswar. This work is done during year 2019-2020, under our guidance.

Date / /

(Prof.Co-Guide Name)

(Prof. Co -Guide Name)

Project Co-Guide

Project Guide

CONTRIBUTION

NAME	ROLL NO.	CNTRIBUTION	SIGN
ANKAN MUKHERJEE	1828049	void main() , void intro(), void book_issue()	
ANUBHAV MISHRA	1828054	void display_all(), void display_allb()	
ASHUTOSH SHARDA	1828056	void write_book(), void write_student(), void admin_menu()	
AYUSH PANDEY	1828059	void display_spb(char n[]), void display_sps(char n[])	
AYUSH JAISWAL	1828060	class student	
DEVTANU MAJUMDAR	1828065	void modify_book(), void modify_student()	
HARSH CHANDRA JHA	1828069	void delete_book(), void delete_student()	
KESHAV KUMAR	1828075	void book_deposit()	
KUMAR SHUBHAM	1828077	class book	

ACKNOWLEDGEMENT

We are profoundly grateful to Prof. MAHENDRA KUMAR GOURISARIA for his expert guidance and continuous encouragement throughout to see that this project rights its target since its commencement to its completion.

GROUP MEMBERS

ANUBHAV MISHRA	ROLL-1828054
ASHUTOSH SHARDA	ROLL-1828056
AYUSH PANDEY	ROLL-1828059
AYUSH JAISWAL	ROLL-1828060
DEVTANU MAJUMDAR	ROLL-1828065
HARSH CHANDRA JHA	ROLL-1828069
KESHAV KUMAR	ROLL-1828075
KUMAR SHUBHAM	ROLL-1828077

ABSTRACT

Library management system is a project which aims in developing a computerized system to maintain all the daily work of library .It also has a facility of admin through which the admin can monitor the whole system .It also has facility of an online notice board where teachers can student can put up information about workshops or seminars being held in our colleges or nearby colleges and librarian after proper verification from the concerned institution organizing the seminar can add it to the notice board . 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 and also the students can request the librarian to add new books by filling the book request form.

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.

CONTENT TABLE

DESCRIPTION	PAGE
1. Introduction	
1.1 PROJECT AIMS AND OBJECTIVES	7
1.2 BACKGROUND OF PROJECT	8
2. Objective	9
3. Project planning	
3.1 SOFTWARE REQUIREMENT SPECIFICATION	
3.1.1. GENERAL DESCRIPTION	10
3.1.2 NON FUNCTIONAL REQUIREMENTS	12
3.1.3 <i>FUNCTIONAL REQUIREMENTS</i>	13
3.2 <i>SOFTWARE AND HARDWARE REQUIREMENTS</i>	
3.2.1 SOFTWARE REQUIREMENTS	14
3.2.2 HARDWARE REQUIREMENTS	14
3.2.3 LANGUAGE USED	15
4. Coding part	16
5. Screen shorts of outputs	46
6. Limitations	
6.1 DISADVANTAGE OF LIBRARY MANAGEMENT SYSTEM	90
6.2 FEATURE AND ADVANTAGE OF THE LIBRARY MANAGEMENT SYSTEM	91
7. Scope of improvement	92
8. Conclusion	93
9. Bibliography/References	94

CHAPTER - 1

INTRODUCTION:

1.1 PROJECT AIMS AND OBJECTIVES :-

The project aims and objectives that will be achieved after completion of this project are discussed in this sub chapter. The aims and objectives are as follows:

- Online book issue
- Request column for librarian for providing new books
- A separate column for digital library
- Student login page where student can find books issued by him/her and date of return.
- A search column to search availability of books
- A teacher login page where teacher can add any events being organized in the college and important suggestions regarding books.
- Online notice board about the workshop.

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.

In addition, report module is also included in Library Management System. If user's position is admin, the user is able to generate different kinds of reports like lists of students registered, list of books, issue and return reports

.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.

CHAPTER - 2

OBJECTIVE:

1. The main objective of the project is to make the book handling.
2. The other things of the staffs of library digital,easy and very friendly which will save the time.
3. In this project we have made the library managemet system which will manage the all the students records and book records the main motive is to bring all the libraray management system online.

CHAPTER - 3

Project Planning:

In this chapter, we will discuss and analyze about the developing process of Library Management System including software requirement specification and comparison between existing and proposed system . The functional and non functional requirements are included in software requirement specification part to provide complete description and overview of system requirement before the developing process is carried out. Besides that, existing vs proposed provides a view of how the proposed system will be more efficient than the existing one.

3.1 SOFTWARE REQUIREMENT SPECIFICATION:-

3.1.1 GENERAL DESCRIPTION:-

i. 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.

ii. **PROBLEM 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. Some times 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 files.

❖ **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.

❖ **Cost consuming**

As there is no computerized system the to add each record paper will be needed which will increase the cost for the management of library.

ii. **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.

❖ **Option of online Notice board**

Librarian will be able to provide a detailed description of workshops going in the college as well as in nearby colleges.

❖ **Lecture Notes**

Teacher have a facility to upload lectures notes in a pdf file having size not more than 10mb

iii. **SYSTEM REQUIREMENTS :**

3.1.2 NON FUNCTIONAL REQUIREMENTS

❖ **Product Requirements**

a. 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 .

b. RELIABILITY REQUIREMENT

The system should accurately performs member registration ,member validation , report generation, book transaction and search.

c. 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.

3.1.3 FUNCTIONAL REQUIREMENTS

1. NORMAL USER

a. REGISTER NEW USER

❖ Description of feature

This feature can be performed by all users to register new user to create account.

❖ Functional requirements

- System must be able to verify information
- System must be able to delete information if information is wrong

b. REGISTER NEW BOOK

❖ Description of feature

This feature allows to add new books to the library.

❖ Functional requirements

- System must be able to verify information
- System must be able to enter number of copies into table.
- System must be able to not allow two books having same book id.

c. SEARCH BOOK

❖ DESCRIPTION OF FEATURE

This feature is found in book maintenance part . we can search book based on book id , book name , publication or by author name.

Functional requirements

- System must be able to search the database based on select search type
- System must be able to filter book based on keyword entered

- System must be able to show the filtered book in table view

d. ISSUE BOOKS AND RETURN BOOKS and DESCRIPTION OF FEATURE

This feature allows to issue and return books and also view reports of book issued.

❖ Functional requirements

- System must be able to enter issue information in database.
- System must be able to update number of books.
- System must be able to search if book is available or not before issuing books
- System should be able to enter issue and return date information

3.2 SOFTWARE AND HARDWARE REQUIREMENTS

This section describes the software and hardware requirements of the system

3.2.1 SOFTWARE REQUIREMENTS

Operating system- Linux and Windows 10 is used as the operating system as it is stable and supports more features and is more user friendly

- C++ programming language
- Turbo C++
- MinGW
- MS Word
- Web Browsers-Google Chrome,Mozilla Firefox
- Code blocks,Visual studio code

3.2.2 HARDWARE REQUIREMENTS

- HP Probook 360 laptop
- Ram 2 GB
- Hard disk 20 GB
- Processor- intel core i5
- Keyboard
- Mouse
- Printer(to print the report)

3.2.3 LANGUAGE USED

C++:C++ is a general-purpose object-oriented programming (OOP) language, developed by Bjarne Stroustrup, and is an extension of the C language. It is therefore possible to code C++ in a "C style" or "object-oriented style." In certain scenarios, it can be coded in either way and is thus an effective example of a hybrid language.

C++ is considered to be an intermediate-level language, as it encapsulates both high- and low-level language features. Initially, the language was called "C with classes" as it had all the properties of the C language with an additional concept of "classes." However, it was renamed C++ in 1983.

C++ is one of the most popular languages primarily utilized with system/application software, drivers, client-server applications and embedded firmware.

The main highlight of C++ is a collection of predefined classes, which are data types that can be instantiated multiple times. The language also facilitates declaration of user-defined classes. Classes can further accommodate member functions to implement specific functionality.

Multiple objects of a particular class can be defined to implement the functions within the class. Objects can be defined as instances created at run time. These classes can also be inherited by other new classes which take in the public and protected functionalities by default.

C++ includes several operators such as comparison, arithmetic, bit manipulation and logical operators. One of the most attractive features of C++ is that it enables the overloading of certain operators such as addition.

A few of the essential concepts within the C++ programming language include polymorphism, virtual and friend functions, templates, namespaces and pointers.

CHAPTER - 4

CODING PART :

```
#include<fstream.h>
#include<conio.h>
#include<stdio.h>
#include<process.h>
#include<string.h>
#include<iomanip.h>//use for setw
class book
{
    char bno[6];

    char bname[50];

    char aname[20];

public:

    void create_book()

    {

        cout<<"\nNEW BOOK ENTRY...\n";

        cout<<"\nEnter The book no.";

        cin>>bno;

        cout<<"\n\nEnter The Name of The Book ";

        gets(bname);
```



```
cout<<"\n\nEnter The Author's Name ";
```

```
gets(aname);
```

```
cout<<"\n\nBook Created..";
```

```
}
```

```
void show_book()
```

```
{
```

```
cout<<"\nBook no. : "<<bno;
```

```
cout<<"\nBook Name : ";
```

```
puts(bname);
```

```
cout<<"Author Name : ";
```

```
puts(aname);
```

```
}
```

```
void modify_book()
```

```
{
```

```
cout<<"\nBook no. : "<<bno;
```

```
cout<<"\nModify Book Name : ";
```

```
gets(bname);
```

```
cout<<"\nModify Author's Name of Book : ";
```

```
    gets(aname);  
  
}  
  
char* retbno()  
  
{  
    return bno;  
}  
  
void report()  
  
{  
    cout<<bno<<setw(30)<<bname<<setw(30)<<aname<<endl;  
}  
  
};
```

```
class student  
  
{  
  
    char admno[6];  
  
    char name[20];  
  
    char stbno[6];
```

```
int token;
```

```
public:
```

```
void create_student()
```

```
{
```

```
clrscr();
```

```
cout<<"\nNEW STUDENT ENTRY...\n";
```

```
cout<<"\nEnter The admission no. ";
```

```
cin>>admno;
```

```
cout<<"\n\nEnter The Name of The Student ";
```

```
gets(name);
```

```
token=0;
```

```
stbno[0]='\0';
```

```
cout<<"\n\nStudent Record Created..";
```

```
}
```

```
void show_student()
```

```
{
```

```
cout<<"\nAdmission no. : "<<admno;
```

```
cout<<"\nStudent Name : ";
```

```
puts(name);
```

```
cout<<"\nNo of Book issued : "<<token;
```

```
if(token==1)
```

```
    cout<<"\nBook No "<<stbno;
```

```
}
```

```
void modify_student()
```

```
{
```

```
    cout<<"\nAdmission no. : "<<admno;
```

```
    cout<<"\nModify Student Name : ";
```

```
    gets(name);
```

```
}
```

```
char* retadmno()
```

```
{
```

```
    return admno;
```

```
}
```

```
char* retstbno()
```

```
{
```

```
    return stbno;
```

```
}
```

```
int rettoken()
```

```
{
```

```
    return token;
```

```
}
```

```
void addtoken()
```

```
{
```

```
token=1;
```

```
}
```

```
void resettoken()
```

```
{
```

```
token=0;
```

```
}
```

```
void getstbno(char t[])
```

```
{
```

```
strcpy(stbno,t);
```

```
}
```

```
void report()
```

```
{
```

```
cout<<"\t"<<admno<<setw(20)<<name<<setw(10)<<token<<endl
```

```
;
```

```
}
```

```
};
```

```
fstream fp,fp1;
```

```
book bk;
```

```
student st;
```

```
// function to write in file
```

```
// ios::allows output (write operations) to a stream.
```

```
//ios::in allows input (read operations) from a stream.
```

```
void write_book()
```

```
{
```

```
    char ch;
```

```
        fp.open("C:\\Users\\KIIT\\Desktop\\book.txt",ios::out|ios::app);
```

```
    do
```

```
    {
```

```
        clrscr();
```

```
        bk.create_book();
```

```
        fp.write((char*)&bk,sizeof(book));
```

```
        cout<<"\n\nDo you want to add more record..(y/n?)";
```

```
        cin>>ch;
```

```
    }while(ch=='y'||ch=='Y');
```

```
    fp.close();

}

//app seek to end before each write

//ate open and seek to end immediately after opening

void write_student()

{
    char ch;
    fp.open("C:\\Users\\KIIT\\Desktop\\student.txt",ios::out|ios::app);

    do

    {
        st.create_student();

        fp.write((char*)&st,sizeof(student));

        cout<<"\n\ndo you want to add more record..(y/n?)";

        cin>>ch;

        }while(ch=='y'||ch=='Y');

    fp.close();
}

//      function to read specific record from file

void display_spb(char n[])
```

```
{

cout<<"\nBOOK DETAILS\n";
int flag=0;
fp.open("C:\\Users\\KIIT\\Desktop\\book.txt",ios::in);
    while(fp.read((char*)&bk,sizeof(book)))

    {
        if(strcmpi(bk.retbn(),n)==0)

        {

            bk.show_book();

            flag=1;

        }
    }

    fp.close();

    if(flag==0)

        cout<<"\n\nBook does not exist";

        getch();

}

void display_sps(char n[])

{

    cout<<"\nSTUDENT DETAILS\n";

    int flag=0;
```



```
fp.open("C:\\Users\\KIIT\\Desktop\\student.txt",ios::in);

while(fp.read((char*)&st,sizeof(student)))
{

    if((strcmpi(st.retadmno(),n)==0))//comparing strings
    {

        st.show_student();

        flag=1;

    }

}

fp.close();

if(flag==0)

    cout<<"\n\nStudent does not exist";

getch();
}

//      function to modify record of file

void modify_book()
{

    char n[6];

    int found=0;
```

```
clrscr();

cout<<"\n\n\tMODIFY BOOK REOCORD.... ";

cout<<"\n\n\tEnter The book no. of The book";

cin>>n;
fp.open("C:\\Users\\KIIT\\Desktop\\book.txt",ios::in|ios::out);

while(fp.read((char*)&bk,sizeof(book)) && found==0)

{

if(strcmpi(bk.retbn(),n)==0)

{

bk.show_book();

cout<<"\nEnter The New Details of book"<<endl;

bk.modify_book();

int pos=-1*sizeof(bk);
fp.seekp(pos,ios::cur);

fp.write((char*)&bk,sizeof(book));

cout<<"\n\n\tRecord Updated";

found=1;

}

}
```

```
fp.close();

if(found==0)

cout<<"\n\n Record Not Found ";

getch();

}

void modify_student()

{
    char n[6];

    int found=0;

    clrscr();

    cout<<"\n\n\tMODIFY STUDENT RECORD... ";
    cout<<"\n\n\tEnter The admission no. of The student";

    cin>>n;
    fp.open("C:\\Users\\KIIT\\Desktop\\student.txt",ios::in|ios::out);

    while(fp.read((char*)&st,sizeof(student)) && found==0)

    {

        if(strcmpi(st.retadmno(),n)==0)

        {
            st.show_student();
```

```

    cout<<"\nEnter The New Details of student"<<endl;

    st.modify_student();

    int pos=-1*sizeof(st);

    fp.seekp(pos,ios::cur);

    fp.write((char*)&st,sizeof(student));

    cout<<"\n\n\t Record Updated";

    found=1;

}
}

fp.close();

if(found==0)

cout<<"\n\n Record Not Found ";

getch();

}
//      function to delete record of file

void delete_student()

{

    char n[6];

```

```
int flag=0;
```

```
clrscr();
```

```
cout<<"\n\n\n\tDELETE STUDENT...";
```

```
cout<<"\n\nEnter The admission no. of the Student You Want To  
Delete : ";
```

```
cin>>n;
```

```
fp.open("C:\\Users\\KIIT\\Desktop\\student.txt",ios::in|ios::out);
```

```
fstream fp2;
```

```
fp2.open("C:\\Users\\KIIT\\Desktop\\Temp.txt",ios::out);
```

```
fp.seekg(0,ios::beg);
```

```
while(fp.read((char*)&st,sizeof(student)))
```

```
{
```

```
    if(strcmpi(st.retadmno(),n)!=0)
```

```
        fp2.write((char*)&st,sizeof(student));
```

```
    else
```

```
        flag=1;
```

```
}
```

```
fp2.close();
```

```
fp.close();
```

```
remove("C:\\Users\\KIIT\\Desktop\\student.txt");
```

```
rename("C:\\Users\\KIIT\\Desktop\\Temp.txt","C:\\Users\\KIIT\\Desktop\\student.txt");
```

```
    if(flag==1)
```

```
        cout<<"\n\n\tRecord Deleted ..";
```

```
    Else
```

```
        cout<<"\n\nRecord not found";
```

```
    getch();
```

```
}
```

```
void delete_book()
```

```
{
```

```
    char n[6];
```

```
    clrscr();
```

```
    cout<<"\n\n\n\tDELETE BOOK ...";
```

```
    cin>>n;
```

```
    fp.open("C:\\Users\\KIIT\\Desktop\\book.txt",ios::in|ios::out);
```

```
    fstream fp2;
```

```
    fp2.open("C:\\Users\\KIIT\\Desktop\\Temp.txt",ios::out);
```

```
    fp.seekg(0,ios::beg);
```

```
    while(fp.read((char*)&bk,sizeof(book)))
```

```

{
    if(strcmpi(bk.retbn(),n)!=0)
    {
        fp2.write((char*)&bk,sizeof(book));
    }
}

fp2.close();

fp.close();

remove("C:\\Users\\KIIT\\Desktop\\book.txt");

rename("C:\\Users\\KIIT\\Desktop\\Temp.txt","C:\\Users\\KIIT\\Desкто
p\\book.txt");

cout<<"\n\n\tRecord Deleted ..";

getch();
}

//      function to display all students list

void display_all()
{

```

```

clrscr();
fp.open("C:\\Users\\KIIT\\Desktop\\student.txt",ios::in);

if(!fp)
{
    cout<<"ERROR!!! FILE COULD NOT BE OPEN ";

    getch();

    return;

}

cout<<"\n\n\t\tSTUDENT LIST\n\n";

cout<<"=====
=====\\n";

cout<<"\tAdmission
No."<<setw(10)<<"Name"<<setw(20)<<"Book Issued\\n";

cout<<"=====
=====\\n";

while(fp.read((char*)&st,sizeof(student)))
{
    st.report();
}

fp.close();

```



```

    getch();

}

//      function to display Books list

void display_allb()

{

    clrscr();
    fp.open("C:\\Users\\KIIT\\Desktop\\book.txt",ios::in);

    if(!fp)

    {

cout<<"ERROR!!! FILE COULD NOT BE OPEN ";

        getch();

        return;

    }

    cout<<"\n\n\t\tBook LIST\n\n";

    cout<<"=====
=====\\n";

    cout<<"Book Number"<<setw(20)<<"Book
Name"<<setw(25)<<"Author\\n";

```

```
cout<<"=====
=====\\n";
```

```
while(fp.read((char*)&bk,sizeof(book)))
{
    bk.report();
}
fp.close();

getch();
}
```

```
//    function to issue book
```

```
void book_issue()
```

```
{

    char sn[6],bn[6];

    int found=0,flag=0;

    clrscr();

    cout<<"\\n\\nBOOK ISSUE ...";

    cout<<"\\n\\n\\tEnter The student's admission no.";
```

```

cin>>sn;
fp.open("C:\\Users\\KIIT\\Desktop\\student.txt",ios::in|ios::out);
    fp1.open("C:\\Users\\KIIT\\Desktop\\book.txt",ios::in|ios::out);

while(fp.read((char*)&st,sizeof(student)) && found==0)

    {

        if(strcmpi(st.retadmno(),sn)==0)
        {

            found=1;

            if(st.rettoken()==0)

                {

                    cout<<"\n\n\tEnter the book no. ";

                    cin>>bn;
                    while(fp1.read((char*)&bk,sizeof(book))&& flag==0)
                {

                    if(strcmpi(bk.retbnno(),bn)==0)

                        {

                            bk.show_book();

                            flag=1;

                            st.addtoken();

                            st.getstbnno(bk.retbnno());

                            int pos=-1*sizeof(st);

                            fp.seekp(pos,ios::cur);

```

```
fp.write((char*)&st,sizeof(student));
```

```
cout<<"\n\n\t Book issued successfully\n\nPlease Note: Write current
date in backside of book and submit within 15 days fine Rs. 1 for each
day after 15 days period";
```

```
    }
```

```
    }
```

```
    if(flag==0)
```

```
        cout<<"Book no does not exist";
```

```
    }
```

```
else
```

```
    cout<<"You have not returned the last book ";
```

```
    }
```

```
}
```

```
    if(found==0)
```

```
        cout<<"Student record not exist...";
```

```
        getch();
```

```
    fp.close();
```

```
    fp1.close();
```

```
}
```

```
//      function to deposit book
```

```

void book_deposit()
{
    char sn[6],bn[6];

    int found=0,flag=0,day,fine;

    clrscr();

    cout<<"\n\nBOOK DEPOSIT ...";

    cout<<"\n\n\tEnter The student's admission no.";

    cin>>sn;

    fp.open("C:\\Users\\KIIT\\Desktop\\student.txt",ios::in|ios::out);
    fp1.open("C:\\Users\\KIIT\\Desktop\\book.txt",ios::in|ios::out);
    while(fp.read((char*)&st,sizeof(student)) && found==0)
    {
        if(strcmpi(st.retadmno(),sn)==0)
        {
            found=1;

            if(st.rettoken()==1)
            {
                while(fp1.read((char*)&bk,sizeof(book))&& flag==0)

```

```

{
    if(strcmpi(bk.retbnno(),st.retstbnno())==0)

    {

        bk.show_book();

        flag=1;

        cout<<"\n\nBook deposited in no. of days";

        cin>>day;

        if(day>15)

        {
            fine=(day-15)*1;

            cout<<"\n\nFine has to deposited Rs. "<<fine;
        }
        st.resettoken();

        int pos=-1*sizeof(st);

        fp.seekp(pos,ios::cur);

        fp.write((char*)&st,sizeof(student));

        cout<<"\n\n\t Book deposited successfully";

    }

}

if(flag==0)

```

```
cout<<"Book no does not exist";
```

```
}
```

```
else
```

```
cout<<"No book is issued..please check!!";
```

```
}
```

```
}
```

```
if(found==0)
```

```
cout<<"Student record not exist...";
```

```
getch();
```

```
fp.close();
```

```
fp1.close();
```

```
}
```

```
//      INTRODUCTION FUNCTION
```

```
void intro()
```

```
{
```

```
clrscr();
```

```
gotoxy(24,2);

cout<<"OBJECT ORIENTED PROGRAMMING PROJECT";

gotoxy(30,4);

cout<<"LIBRARY MANAGEMENT SYSTEM";

cout<<"\n\n\nMADE BY GROUP 1";

cout<<"\n\n\tNAME \t\t\t ROLL";

cout<<"\n\nAnkan Mukherjee\t\t-->\t\t1828049";

cout<<"\n\nAnubhav Mishra\t\t-->\t\t1828054";

cout<<"\n\nAshutosh Sharda\t\t-->\t\t1828056";

cout<<"\n\nAyush Pandey\t\t-->\t\t1828059";

cout<<"\n\nAyush Jaiswal\t\t-->\t\t1828060";

cout<<"\n\nDevtanu Majumder\t\t-->\t\t1828065";

cout<<"\n\nHarsh Chandra Jha\t\t-->\t\t1828069";

cout<<"\n\nKeshav Kumar\t\t-->\t\t1828075";

cout<<"\n\nKumar Shubham\t\t-->\t\t1828077";

cout<<"\n\nUNIVERSITY: KALINGA INSTITUTE OF
INDUSTRIAL TECHNOLOGY";
```



```
cout<<"\n\nSCHOOL OF COMPUTER ENGINEERING";
```

```
cout<<"\n\nUNDER THE GUIDANCE OF MAHENDRA  
KUMAR GOURISARIA";
```

```
getch();
```

```
}
```

```
//      ADMINISTRATOR MENU FUNCTION
```

```
void admin_menu()
```

```
{
```

```
clrscr();
```

```
int ch2;
```

```
cout<<"\n\n\n\tADMINISTRATOR MENU";
```

```
cout<<"\n\n\t1.CREATE STUDENT RECORD";
```

```
cout<<"\n\n\t2.DISPLAY ALL STUDENTS RECORD";
```

```
cout<<"\n\n\t3.DISPLAY SPECIFIC STUDENT RECORD ";
```

```
cout<<"\n\n\t4.MODIFY STUDENT RECORD";
```

```
cout<<"\n\n\t5.DELETE STUDENT RECORD";
```

```
cout<<"\n\n\t6.CREATE BOOK ";

cout<<"\n\n\t7.DISPLAY ALL BOOKS ";

cout<<"\n\n\t8.DISPLAY SPECIFIC BOOK ";

cout<<"\n\n\t9.MODIFY BOOK ";

cout<<"\n\n\t10.DELETE BOOK ";

cout<<"\n\n\t11.BACK TO MAIN MENU";

cout<<"\n\n\tPlease Enter Your Choice (1-11) ";

cin>>ch2;

switch(ch2)

{
    case 1: clrscr();

        write_student();break;

    case 2: display_all();break;

    case 3:
        char num[6];

        clrscr();
        cout<<"\n\n\tPlease Enter The Admission No. ";

        cin>>num;

        display_sps(num);

        break;
```

```
case 4: modify_student();break;

case 5: delete_student();break;

case 6: clrscr();

write_book();break;

case 7: display_allb();break;

case 8:
    {
        char num[6];

        clrscr();

        cout<<"\n\n\tPlease Enter The book No. ";
        cin>>num;

        display_spb(num);

        break;
    }

case 9: modify_book();break;

case 10: delete_book();break;

case 11: return;

default:cout<<"\a";

}
```

```
    admin_menu();
}

//      THE MAIN FUNCTION OF PROGRAM

void main()
{
    char ch;

    intro();

    do
    {
        clrscr();
        cout<<"\n\n\tMAIN MENU";

        cout<<"\n\n\t01. BOOK ISSUE";

        cout<<"\n\n\t02. BOOK DEPOSIT";

        cout<<"\n\n\t03. ADMINISTRATOR MENU";
        cout<<"\n\n\t04. EXIT";

        cout<<"\n\n\tPlease Select Your Option (1-4) ";
        ch=getche();

        switch(ch)
        {
```

```
case '1':clrscr();  
  
    book_issue();  
  
    break;  
  
case '2':book_deposit();  
  
    break;  
  
case '3':admin_menu();  
  
    break;  
  
case '4':exit(0);  
default :cout<<"\a";  
  
}
```

```
}while(ch!='4');
```

```
}
```

```
/
```

```
//
```

END OF PROJECT THANK YOU

CHAPTER - 5

SCREENSHOTS OF

OUTPUTS :

After running the code

The front page:

```
OBJECT ORIENTED PROGRAMMING PROJECT
LIBRARY MANAGEMENT SYSTEM
```

```
MADE BY : GROUP 1
```

NAME		ROLL
Ankan Mukherjee	-->	1828049
Anubhav Mishra	-->	1828054
Ashutosh Sharda	-->	1828056
Ayush Pandey	-->	1828059
Ayush Jaiswal	-->	1828060
Devtanu Majumder	-->	1828065
Harsh Chandra Jha	-->	1828069
Keshav Kumar	-->	1828075
Kumar Subham	-->	1828077

```
UNIVERSITY: KALINGA INSTITUTE OF INDUSTRIAL TECHNOLOGY
```

```
SCHOOL OF COMPUTER ENGINEERING
```

```
UNDER THE GUIDANCE OF MAHENDRA KUMAR GOURISARIA _
```

After pressing ENTER KEY

The main menu:

MAIN MENU

01. BOOK ISSUE

02. BOOK DEPOSIT

03. ADMINISTRATOR MENU

04. EXIT

Please Select Your Option (1-4)

After pressing 3(3 rd option) --> ENTER

ADMINISTRATOR MENU:

ADMINISTRATOR MENU

1.CREATE STUDENT RECORD

2.DISPLAY ALL STUDENTS RECORD

3.DISPLAY SPECIFIC STUDENT RECORD

4.MODIFY STUDENT RECORD

5.DELETE STUDENT RECORD

6.CREATE BOOK

7.DISPLAY ALL BOOKS

8.DISPLAY SPECIFIC BOOK

9.MODIFY BOOK

10.DELETE BOOK

11.BACK TO MAIN MENU

Please Enter Your Choice (1-11)

After pressing “1 ”-->ENTER

CREATE STUDENT RECORD MENU:

We have to enter student's admission no and name from keyboard

```
NEW STUDENT ENTRY...
```

```
Enter The admission no. 49
```

```
Enter The Name of The Student ankan
```

```
Student Record Created..
```

```
do you want to add more record..(y/n?)
```

After pressing “y” --> ENTER

We have to enter student's admission no and name from keyboard

Another student record will be created

```
NEW STUDENT ENTRY...
```

```
Enter The admission no. 54
```

```
Enter The Name of The Student anubhav
```

```
Student Record Created..
```

```
do you want to add more record..(y/n?)_
```

After pressing “y” --> ENTER

We have to enter student's admission no and name from keyboard

Another student record will be created

```
NEW STUDENT ENTRY...
```

```
Enter The admission no. 69
```

```
Enter The Name of The Student harsh
```

```
Student Record Created..
```

```
do you want to add more record..(y/n?)_
```

After pressing “n” --> ENTER

```
NEW STUDENT ENTRY...
```

```
Enter The admission no. 69
```

```
Enter The Name of The Student harsh
```

```
Student Record Created..
```

```
do you want to add more record..(y/n?)n
```

The console will back to the ADMINSTRATOR MENU

ADMINISTRATOR MENU

1.CREATE STUDENT RECORD

2.DISPLAY ALL STUDENTS RECORD

3.DISPLAY SPECIFIC STUDENT RECORD

4.MODIFY STUDENT RECORD

5.DELETE STUDENT RECORD

6.CREATE BOOK

7.DISPLAY ALL BOOKS

8.DISPLAY SPECIFIC BOOK

9.MODIFY BOOK

10.DELETE BOOK

11.BACK TO MAIN MENU

Please Enter Your Choice (1-11) _

After pressing “2” --> ENTER

DISPLAY ALL STUDENTS RECORDS MENU:

STUDENT LIST		
=====		
Admission No.	Name	Book Issued
=====		
49	ankan	0
54	anubhav	0
69	harsh	0

After pressing ENTER

The ADMINSTRATOR MENU

ADMINISTRATOR MENU

1.CREATE STUDENT RECORD

2.DISPLAY ALL STUDENTS RECORD

3.DISPLAY SPECIFIC STUDENT RECORD

4.MODIFY STUDENT RECORD

5.DELETE STUDENT RECORD

6.CREATE BOOK

7.DISPLAY ALL BOOKS

8.DISPLAY SPECIFIC BOOK

9.MODIFY BOOK

10.DELETE BOOK

11.BACK TO MAIN MENU

Please Enter Your Choice (1-11) _

After pressing “3” --> ENTER

DISPLAY SPECIFIC STUDENT RECORD MENU:

We have to enter student's admission no

```
      Please Enter The Admission No. 49

STUDENT DETAILS
Admission no. : 49
Student Name : ankan
No of Book issued : 0_
```


After pressing ENTER

The ADMINSTRATOR MENU

ADMINISTRATOR MENU

1.CREATE STUDENT RECORD

2.DISPLAY ALL STUDENTS RECORD

3.DISPLAY SPECIFIC STUDENT RECORD

4.MODIFY STUDENT RECORD

5.DELETE STUDENT RECORD

6.CREATE BOOK

7.DISPLAY ALL BOOKS

8.DISPLAY SPECIFIC BOOK

9.MODIFY BOOK

10.DELETE BOOK

11.BACK TO MAIN MENU

Please Enter Your Choice (1-11) _

After pressing “4” --> ENTER

MODIFY STUDENT RECORD MENU :

After entering student's admission no , the compiler will ask for the new details of the student

*Then we have to enter the new name of the student
--> ENTER*

Record will be updated

```
MODIFY STUDENT RECORD...

Enter The admission no. of The student54

Admission no. : 54
Student Name : anubhav

No of Book issued : 0
Enter The New Details of student

Admission no. : 54
Modify Student Name : mishra

Record Updated_
```

After pressing ENTER

The ADMINSTRATOR MENU

ADMINISTRATOR MENU

1.CREATE STUDENT RECORD

2.DISPLAY ALL STUDENTS RECORD

3.DISPLAY SPECIFIC STUDENT RECORD

4.MODIFY STUDENT RECORD

5.DELETE STUDENT RECORD

6.CREATE BOOK

7.DISPLAY ALL BOOKS

8.DISPLAY SPECIFIC BOOK

9.MODIFY BOOK

10.DELETE BOOK

11.BACK TO MAIN MENU

Please Enter Your Choice (1-11) _

Now we can check the new modified student record

After pressing “2” --> ENTER

DISPLAY ALL STUDENTS RECORD MENU:

STUDENT LIST		
Admission No.	Name	Book Issued
49	ankan	0
54	mishra	0
69	harsh	0

After pressing ENTER

The ADMINSTRATOR MENU

ADMINISTRATOR MENU

1.CREATE STUDENT RECORD

2.DISPLAY ALL STUDENTS RECORD

3.DISPLAY SPECIFIC STUDENT RECORD

4.MODIFY STUDENT RECORD

5.DELETE STUDENT RECORD

6.CREATE BOOK

7.DISPLAY ALL BOOKS

8.DISPLAY SPECIFIC BOOK

9.MODIFY BOOK

10.DELETE BOOK

11.BACK TO MAIN MENU

Please Enter Your Choice (1-11) _

Now we can delete any of the students record by moving the 5th menu

After pressing “5” --> ENTER

DELETE STUDENTS RECORD MENU:

The compiler will ask for the student's admission no

```
DELETE STUDENT...  
Enter The admission no. of the Student You Want To Delete : 54  
  
Record Deleted ..
```

After pressing ENTER

The ADMINSTRATOR MENU

```
ADMINISTRATOR MENU

1.CREATE STUDENT RECORD

2.DISPLAY ALL STUDENTS RECORD

3.DISPLAY SPECIFIC STUDENT RECORD

4.MODIFY STUDENT RECORD

5.DELETE STUDENT RECORD

6.CREATE BOOK

7.DISPLAY ALL BOOKS

8.DISPLAY SPECIFIC BOOK

9.MODIFY BOOK

10.DELETE BOOK

11.BACK TO MAIN MENU

Please Enter Your Choice (1-11) _
```

Now we can check DISPLAY ALL STUDENTS RECORD MENU

After pressing “2” --> ENTER

DISPLAY ALL STUDENTS RECORD MENU

STUDENT LIST		
=====		
Admission No.	Name	Book Issued
=====		
49	ankan	0
69	harsh	0

After pressing ENTER

The ADMINSTRATOR MENU

ADMINISTRATOR MENU

1.CREATE STUDENT RECORD

2.DISPLAY ALL STUDENTS RECORD

3.DISPLAY SPECIFIC STUDENT RECORD

4.MODIFY STUDENT RECORD

5.DELETE STUDENT RECORD

6.CREATE BOOK

7.DISPLAY ALL BOOKS

8.DISPLAY SPECIFIC BOOK

9.MODIFY BOOK

10.DELETE BOOK

11.BACK TO MAIN MENU

Please Enter Your Choice (1-11) _

We can create book by moving to the 6 th menu

After pressing “6” --> ENTER

The compiler will ask for the book no,book name and author's name

After entering the above data properly --> ENTER

The book record will be created and the compiler will ask for entering more record

```
NEW BOOK ENTRY...  
Enter The book no.1  
  
Enter The Name of The Book aa  
  
Enter The Author's Name AA  
  
Book Created..  
Do you want to add more record..(y/n?)
```

After pressing "y" -->ENTER

Another book record will be created

NEW BOOK ENTRY...

Enter The book no.2

Enter The Name of The Book bb

Enter The Author's Name BB

Book Created..

Do you want to add more record..(y/n?)n_



Then if we press “n” --> ENTER

The ADMINSTRATOR MENU

ADMINISTRATOR MENU

1.CREATE STUDENT RECORD

2.DISPLAY ALL STUDENTS RECORD

3.DISPLAY SPECIFIC STUDENT RECORD

4.MODIFY STUDENT RECORD

5.DELETE STUDENT RECORD

6.CREATE BOOK

7.DISPLAY ALL BOOKS

8.DISPLAY SPECIFIC BOOK

9.MODIFY BOOK

10.DELETE BOOK

11.BACK TO MAIN MENU

Please Enter Your Choice (1-11) _

Now we can display the book record by moving to the 7 th menu

After pressing “7” --> ENTER

DISPLAY ALL BOOKS MENU:

Book LIST		
Book Number	Book Name	Author
1	aa	AA
2	bb	BB

*After pressing ENTER**The ADMINSTRATOR MENU*

ADMINISTRATOR MENU

1.CREATE STUDENT RECORD

2.DISPLAY ALL STUDENTS RECORD

3.DISPLAY SPECIFIC STUDENT RECORD

4.MODIFY STUDENT RECORD

5.DELETE STUDENT RECORD

6.CREATE BOOK

7.DISPLAY ALL BOOKS

8.DISPLAY SPECIFIC BOOK

9.MODIFY BOOK

10.DELETE BOOK

11.BACK TO MAIN MENU

Please Enter Your Choice (1-11) _

We can display specific book by moving to the 8 th option.

After pressing “8” --> ENTER

DISPLAY SPECIFIC BOOK MENU:

The compiler will ask for the no of the book

After entering the book no the compiler will display the book details

```
      Please Enter The book No. 1

BOOK DETAILS
Book no. : 1
Book Name : aa
Author Name : AA
```

After pressing ENTER

The ADMINSTRATOR MENU

ADMINISTRATOR MENU

- 1.CREATE STUDENT RECORD
 - 2.DISPLAY ALL STUDENTS RECORD
 - 3.DISPLAY SPECIFIC STUDENT RECORD
 - 4.MODIFY STUDENT RECORD
 - 5.DELETE STUDENT RECORD
 - 6.CREATE BOOK
 - 7.DISPLAY ALL BOOKS
 - 8.DISPLAY SPECIFIC BOOK
 - 9.MODIFY BOOK
 - 10.DELETE BOOK
 - 11.BACK TO MAIN MENU
- Please Enter Your Choice (1-11) _

Now we can modify book record by moving to the 9 th menu

After pressing “9” -->ENTER

MODIFY BOOK MENU :

The compiler will ask for the book no.

Then it will ask for the new book details

After entering the new book details -->ENTER

```
MODIFY BOOK RECORD....

Enter The book no. of The book1

Book no. : 1
Book Name : aa
Author Name : AA

Enter The New Details of book

Book no. : 1
Modify Book Name : aaa

Modify Author's Name of Book : AAA

Record Updated_
```

After pressing ENTER

The ADMINSTRATOR MENU

ADMINISTRATOR MENU

1.CREATE STUDENT RECORD

2.DISPLAY ALL STUDENTS RECORD

3.DISPLAY SPECIFIC STUDENT RECORD

4.MODIFY STUDENT RECORD

5.DELETE STUDENT RECORD

6.CREATE BOOK

7.DISPLAY ALL BOOKS

8.DISPLAY SPECIFIC BOOK

9.MODIFY BOOK

10.DELETE BOOK

11.BACK TO MAIN MENU

Please Enter Your Choice (1-11) _

Now we can check the new book list by moving 7 th menu

DISPLAY ALL BOOKS MENU

Book LIST		
Book Number	Book Name	Author
2	bb	BB
1	aaa	AAA

After pressing ENTER

The ADMINSTRATOR MENU

ADMINISTRATOR MENU

1.CREATE STUDENT RECORD

2.DISPLAY ALL STUDENTS RECORD

3.DISPLAY SPECIFIC STUDENT RECORD

4.MODIFY STUDENT RECORD

5.DELETE STUDENT RECORD

6.CREATE BOOK

7.DISPLAY ALL BOOKS

8.DISPLAY SPECIFIC BOOK

9.MODIFY BOOK

10.DELETE BOOK

11.BACK TO MAIN MENU

Please Enter Your Choice (1-11) _

Now we can delete any of the book records from the list by moving to the 10 th menu

After pressing “10” --> ENTER

The compiler will ask for the book no.

After entering the book no --> ENTER



After pressing ENTER

The ADMINSTRATOR MENU

ADMINISTRATOR MENU

1.CREATE STUDENT RECORD

2.DISPLAY ALL STUDENTS RECORD

3.DISPLAY SPECIFIC STUDENT RECORD

4.MODIFY STUDENT RECORD

5.DELETE STUDENT RECORD

6.CREATE BOOK

7.DISPLAY ALL BOOKS

8.DISPLAY SPECIFIC BOOK

9.MODIFY BOOK

10.DELETE BOOK

11.BACK TO MAIN MENU

Please Enter Your Choice (1-11) _

Now we can check the the book list by moving to the 7 th option.

After pressing “7” --> ENTER

Book LIST		
Book Number	Book Name	Author
2	bb	BB

After pressing ENTER

The ADMINSTRATOR MENU

ADMINISTRATOR MENU

1.CREATE STUDENT RECORD

2.DISPLAY ALL STUDENTS RECORD

3.DISPLAY SPECIFIC STUDENT RECORD

4.MODIFY STUDENT RECORD

5.DELETE STUDENT RECORD

6.CREATE BOOK

7.DISPLAY ALL BOOKS

8.DISPLAY SPECIFIC BOOK

9.MODIFY BOOK

10.DELETE BOOK

11.BACK TO MAIN MENU

Please Enter Your Choice (1-11) _

After pressing “11” we can back to the main menu

After pressing “11” --> ENTER

MAIN MENU:

MAIN MENU

01. BOOK ISSUE

02. BOOK DEPOSIT

03. ADMINISTRATOR MENU

04. EXIT

Please Select Your Option (1-4) _

We can issue book by moving to the 1 st menu

After entering “1” --> ENTER

BOOK ISSUE MENU:

The compiler will ask for the student's admission no and the book no

After entering admission no and the book no --> ENTER

```
BOOK ISSUE ...  
  
Enter The student's admission no.49  
  
Enter the book no. 2  
  
Book no. : 2  
Book Name : bb  
Author Name : BB  
  
Book issued successfully  
  
Please Note: Write current date in backside of book and submit within 15 days fi  
ne Rs. 1 for each day after 15 days period
```

The compiler will display a message for deposit the book within 15 days . On the other way it will take a fine of Rs.1 per day.

After pressing ENTER

MAIN MENU:

MAIN MENU

01. BOOK ISSUE

02. BOOK DEPOSIT

03. ADMINISTRATOR MENU

04. EXIT

Please Select Your Option (1-4) _

Now we can go to the ADMINSTRATOR MENU and check for the students record. We will find that 0 is replaced by 1 in the book issue column of the particular student.

STUDENT LIST

=====		
Admission No.	Name	Book Issued
=====		
49	ankan	1
69	harsh	0

A particular student will not be able to issue another book before depositing the previous book.

BOOK ISSUE ...

Enter The student's admission no.49

You have not returned the last book _

Now after pressing "11" --> ENTER

MAIN MENU :

MAIN MENU

01. BOOK ISSUE

02. BOOK DEPOSIT

03. ADMINISTRATOR MENU

04. EXIT

Please Select Your Option (1-4) _

Now we will move for the book deposit menu

After pressing “2” --> ENTER

BOOK DEPOSIT MENU:

Now the compiler will ask for the student's admission no and the no of days after after which he/she come to deposit the book

*After entering the admission no and no of days
-->ENTER*

```
BOOK DEPOSIT ...
```

```
Enter The student's admission no.49
```

```
Book no. : 2
```

```
Book Name : bb
```

```
Author Name : BB
```

```
Book deposited in no. of days?
```

```
Book deposited successfully
```

If the no of days is greater than 15 then the compiler will mention an amount as FINE

BOOK DEPOSIT ...

Enter The student's admission no.49

Book no. : 2

Book Name : bb

Author Name : BB

Book deposited in no. of days20

Fine has to deposited Rs. 5

Book deposited successfully

After pressing ENTER

MAIN MENU:

MAIN MENU

01. BOOK ISSUE

02. BOOK DEPOSIT

03. ADMINISTRATOR MENU

04. EXIT

Please Select Your Option (1-4) _

After pressing “4” --> ENTER

The programme will be ended.

CHAPTER - 6

LIMITATIONS

Just like any other object on the earth is bounded with some restrictions, so is the case with this software. It is also true that each and every software has some limitations, and even after trying our best still some are left.

This application will run only on a standalone pc. It is designed for a small organization. It is also found that I am unable to meet all the requirements of the organization because it does not support LAN or WAN. It's also not multiuser.

6.1 DISADVANTAGE OF LIBRARY MANAGEMENT SYSTEM

- The data stored is prone to cyber hacks. Opting for a reliable online system eliminates the risk. Costly and Expensive
- Complicated to operate
- Online Systems require high-speed internet connectivity
- Risk of computer virus
- The automation feature is not available in offline/open source systems thus, requires manual action to perform operations
- Unlike online systems that utilize cloud computing, Open-source systems store data on computer hard drive. This increases the risk of data loss

6.2 FEATURE AND ADVANTAGE OF THE LIBRARY MANAGEMENT SYSTEM

- Manage the complete management of the entire library through the software's easy interface
- It removes manual process of issuing books by easy and simplified way of issuing book saving time and effort
- The librarian can issue, return and reserve book for a particular student through the software's interface
- The software automatically shows fine levied by automatically counting days from the date if Issue incase of late return of the book
- Add, update, search and view library items online
- Student can also check the availability status of a particular book online
- Generate customized report for library items, library inventory and library fine collection

CHAPTER - 7

SCOPE OF IMPROVEMENT

The school library management system is very simple yet unique idea that helps users to access data in the form of library UI which very primitive. Every project have its demerits and hence have some scope of improvement. So in this section of report we try to present some of the improvement that our team find that could have been done to make our project more user friendly. Following points ahead will present the same :

At the Finishing time of this project, one thing that came across many of our team mates was that, we could've created a better GUI or any better background at different sections of the UI. Doing this could have really helped in making the UI of this project more user friendly and attractive as well.

Another improvement that can be done in this project was that we could've used proper database to store our data (like student info or book info, etc), we can store offline (using mySQL) or online (using firebase). Acheiving this could've made our project more realistic and better.

Having said all this points for scope of improvement in our project, we as in our entire team is proud for what we have created, because there's always something more that can be done, hence possibilities are neverending.

CHAPTER - 8

CONCLUSION

Allows the user to store the book details and the customer details. This software package allows storing the details of all the data related to library. The system is strong enough to withstand regressive yearly operations under conditions where the database is maintained and cleared over a certain time of span. The implementation of the system in the organization will considerably reduce data entry, time and also provide readily calculated reports.

CHAPTER - 9

BIBLIOGRAPHY /

REFERENCES

9.1 BOOKS:

a) C++ Primer (5th Edition)

by Barbara E. Moo, Josée Lajoie, Stanley B. Lippman

Publisher: Addison-Wesley Professional

Release Date: August 2012

ISBN: 9780133053043

b) Programming: Principles and Practice Using C++

Programming: Principles and Practice Using C++

2nd Edition, Kindle Edition

by Bjarne Stroustrup

c) Object-Oriented Programming with C++

Object-Oriented Programming with C++

Paperback – 20 Sep 2017

by E Balagurusamy

9.2 WEBSITES:

- <http://www.w3schools.com>
- <http://www.geeksforgeeks.org>
- <http://www.slideshare.net>
- <http://www.github.com>