Siddaganga Institute of Technology, Tumakuru

(An Autonomous Institution affiliated to Visvesvaraya Technological University, Belagavi, Approved by AICTE, New Delhi, Accredited by NAAC and ISO 9001:2015 certified)

OPEN ENDED PROJECT

on

"LIBRARY MANAGEMENT SYSTEM"

submitted
in the partial fulfillment of the requirements IV semester
Bachelor of Engineering
In
Computer Science and Engineering

by

ADEEBA MIR 1SI20CS004
ADITYA RANJAN 1SI20CS005
AKASH RANJAN 1SI20CS007
AKRITI RANJAN 1SI20CS008

Department of Computer Science & Engineering
(Program Accredited by NBA)
Siddaganga Institute of Technology
B.H Road, Tumakuru-572103, Karnataka, India.
Web: www.sit.ac.in

Siddaganga Institute of Technology, Tumakuru

(An Autonomous Institution affiliated to Visvesvaraya Technological University, Belagavi,
Approved by AICTE, New Delhi, Accredited by NAAC and ISO 9001:2015 certified)

Department of Computer Science and Engineering

(Program Accredited by NBA)

CERTIFICATE

This is to certify that open ended project entitled "LIBRARY MANAGEMENT SYSTEM" is a bona fide work carried out by **ADEEBA MIR, ADITYA RANJAN, AKASH RANJAN, AKRITI RANJAN** of IV semester **Bachelor of Engineering in Computer Science and Engineering** of the **SIDDAGANGA INSTITUTE OF TECHNOLOGY** (An Autonomous Institution, affiliated to VTU, Belagavi, Approved by AICTE, New Delhi, Accredited by NAAC and ISO 9001:2015 certified) during the academic year 2021-2022.

Name of Faculty **Signature**

Dr. MANJUNATH KG

Abstract

Library Management Systems are one of the most popular and recognized tools to assist in the management of library books and dues. Library Management System is one of the most widely needed tool in the schools and colleges, and it is helpful for the institutes to update the student book issue details easily and efficiently. This application will ease the process of this job than earlier pen-paper based system.

Now a days this kind of application is very essential for any small or medium sized organization. This Library Management System enables the user to maintain the library system in schools and colleges helps in administrator menu, book issue, book deposit, dues and maintaining a student detail. It is a software with menu driven programs with user-friendly interface which efficiently takes care of the library activities of the organization. An accountant regardless of the number of staffs and students can maintain books details digitally. The software provides benefits like quick find out of information and faster access of information. It also removes the monotonous access to the information.

Main aim of this software is to provide easy and efficient functionalities for library activities and also to provide full functional reports to management of institute with necessary details.

Table of Contents

Introduction

This Library Management System project is for monitoring and controlling the transactions in a library. It mainly focuses on basic operations in a library like adding new members, new books, and updating new information, searching books and members and provides facility to borrow and return books.

We can enter the record of new books and retrieve the details of books available in the library. We can issue the books to the students and maintain their records. Late fine is charged for students who return the issued books after the due date. Only one book is issued to students at a time. New Book is not issued to students who have not returned the last book.

Objective

The objective of the project is to computerise the functions of the Library. Computerisation helps in better record keeping and management of the Library. This simplifies the work of the Librarian as he/she is able to keep a better check on which book is issued by which student.

Newly arrived books can easily be added, and data about any book can easily be modified with the help of this project. Also, records can be transferred from one system to the other without any loss of information.

System Implementation

Hardware:

PC with Intel i5 quad core processor having 16.00 GB RAM , 500 GB SSD, SVGA and other required devices.

Software:

Microsoft Windows® 10 as Operating System.

Visual Studio Code for coding.

Purpose

Main aim of this software is to provide easy and efficient functionalities for library activities and also to provide full functional reports to management of library with necessary details.

Nowadays large-scale organizations (school in case) are committed to bring the best way of management in the various forms. It is a tool to manage the inner operation of the organization related to student fees.

Benefits

- Interact with the software with menu-driven programs with user friendly interface.
- Manage student information efficiently.
- To provide easy and faster access to information.
- Prepare the detailed books record of all the students in an institute.

Problem Statement

Nowadays schools and universities are committed to bring the best way of management in the various forms. In every part of the institution like library, accounts etc. there are need of software which must provide efficient and faster completion of work. Among the various methods of maintaining records of students most used once are maintaining ledgers or excel files which are commonly used in the organizations. These methods are time consuming and inefficient. Hence an alternative is needed. The alternative must need to have few objectives according to the requirements of the organization.

Objectives

- Develop a system that will improve the library management process in the timekeeping and maintaining record of students.
- Develop a system that will monitor student data that is efficient to use.
- Secure the record of books and to have more manageable files, managed by administrator of the department.
- Summarize all the details of issued books.
- Develop a system to provide easy and faster access to information.

Header Files

1. #include<fstream.h>
2. #include<conio.h>
3. #include<stdio.h>
4. #include<process.h>
5. #include<string.h>
6. #include<iomanip.h>

Classes

1. CLASS BOOK

Private Members

Data members

bno	:	char type (6 characters)
bname	:	char type (50 characters)
aname	:	char type (20 characters)

Public Members

Functions

create_book()	:	to read bno, bname & aname and create book.
show_book()	:	to display details of the book when it is issued.
modify_book()	:	to modify the name & author of the
bookretbno()	:	to return bno

2. CLASS STUDENT

Private Members

Data members

Admno	:	char type (6 characters)
name	:	char type (20 characters)
stbno	:	char type (6 characters)
token	:	integer type

Public Members

Functions

create_student()	:	to create student record.
show_student()	:	to display a specific student record.
modify_student()	:	to modify student's name.
retadmno()	:	to return admission no. of student.
retstbno()	:	to return the issued book no.
addtoken()	:	to store the value of token as 1 book is issued.
resettoken()	:	to reset the value of token to 0 is book is returned.

Functions

functions to write in file
write_book()
write_student()

functions to read specific record from file
display_spb()
display_sps()

function to modify record of file
modify_book()
modify_student()

function to delete record from file
delete_student()
delete_book()

function to display all record lists
display_alls()
display_allb()

Other functions

book_issue()	:	function to issue book
book_deposit()	:	function to deposit book
intro()	:	Introduction function
admin_menu()	:	Administrator menu function

Source Code

```
#include <bits\stdc++.h>
#include <fstream>
using namespace std;

// CLASS BOOK
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 ";
      cin >> bname;
      cout << "\n\nEnter the author's name ";
      cin >> aname;
      cout << "\n\n\nBOOK CREATED..";
   }
   void show_book()
   {
      cout << "\nBook no. : ";
      cout << bno;
      cout << "\nBook name : ";
      puts(bname);
      cout << "Author name : ";
      puts(aname);
   }
   void modify_book()
   {
      cout << "\nBook no. : " << bno;
      cout << "\nModify book name : ";
      cin >> bname;
      cout << "\nModify author's name of book:";
      cin >> aname;
   }
   char *retbno()
   {
      return bno;
   }
   void report()
   {
      cout << bno << setw(30) << bname << setw(30) << aname << endl;
   }
```

```
}; // CLASS BOOK ENDS

// CLASS STUDENT
class student
{
   char admno[6];
   char name[20];
   char stbno[6];
   int token;
   public:
   void create_student()
   {
      cout << "\nNEW STUDENT ENTRY...\n";
      cout << "\nEnter the admission no. ";
      cin >> admno;
      cout << "\n\nEnter the name of the student";
      cin >> name;
      token = 0;
      stbno[0] = '\0';
      cout << "\n\nSTUDENT RECORD CREATED..";
   }
   void show_student()
   {
      cout << "\nAdmission no. : ";
      cout << 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 : ";
      cin >> 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);
       cout << token << endl;
    }

}; // CLASS STUDENT ENDS

fstream fp, fp1;
book bk;
student st;

void write_book()
{
    char ch;
    fp.open("book.dat", ios::out | ios::app);
    do
    {
       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();
}
void write_student()
{
    char ch;
    fp.open("student.dat", 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();
}
```

```
void display_spb(char n[])
{
   cout << "\nBOOK DETAILS\n";
   int flag = 0;
   fp.open("book.dat", ios::in);
   while (fp.read((char *)&bk, sizeof(book)))
   {
      if (strcmp(bk.retbno(), n) == 0)
      {
         bk.show_book();
         flag = 1;
      }
   }
   fp.close();
   if (flag == 0)
   {
      cout << "\n\nBook does not exist";
   }
   cin.get();
}
void display_sps(char n[])
{
   cout << "\nSTUDENT DETAILS\n";
   int flag = 0;
   fp.open("student.dat", ios::in);
   while (fp.read((char *)&st, sizeof(student)))
   {
      if ((strcmp(st.retadmno(), n) == 0))
      {
         st.show_student();
         flag = 1;
      }
   }
   fp.close();
   if (flag == 0)
   {
      cout << "\n\nStudent does not exist";
   }
   cin.get();
}
void modify_book()
{
   char n[6];
   int found = 0;
   cout << "\n\n\tMODIFY BOOK REOCORD.... ";
   cout << "\n\n\tEnter The book no. of The book";
   cin >> n;
   fp.open("book.dat", ios::in | ios::out);
   while (fp.read((char *)&bk, sizeof(book)) && found == 0)
   {
```

```
        if (strcmp(bk.retbno(), n) == 0)
        {
            bk.show_book();
            cout << "\nEnter The New Details of book";
            cout << endl;
            bk.modify_book();
            int pos = -1 * sizeof(bk);
            fp.seekp(pos, ios::cur);
            fp.write((char *)&bk, sizeof(book));
            cout << "\n\n\t Record Updated";
            found = 1;
        }
    }
    fp.close();
    if (found == 0)
    {
        cout << "\n\n Record Not Found ";
    }
    cin.get();
}
void modify_student()
{
    char n[6];
    int found = 0;
    cout << "\n\n\tMODIFY STUDENT RECORD... ";
    cout << "\n\n\tEnter the admission no. of the student";
    cin >> n;
    fp.open("student.dat", ios::in | ios::out);
    while (fp.read((char *)&st, sizeof(student)) && found == 0)
    {
        if (strcmp(st.retadmno(), n) == 0)
        {
            st.show_student();
            cout << "\nEnter the new details of student";
            cout << 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 ";
    }
    cin.get();
}
```

```
void delete_student()
{
   char n[6];
   int flag = 0;
   cout << "\n\n\n\tDELETE STUDENT...";
   cout << "\n\nEnter the admission no. of the student you want To Delete : ";
   cin >> n;
   fp.open("student.dat", ios::in | ios::out);
   fstream fp2;
   fp2.open("Temp.dat", ios::out);
   fp.seekg(0, ios::beg);
   while (fp.read((char *)&st, sizeof(student)))
   {
      if (strcmp(st.retadmno(), n) != 0)
      {
         fp2.write((char *)&st, sizeof(student));
      }
      else
      {
         flag = 1;
      }
   }
   fp2.close();
   fp.close();
   remove("student.dat");
   rename("Temp.dat", "student.dat");
   if (flag == 1)
   {
      cout << "\n\n\tRECORD DELETED ..";
   }
   else
   {
      cout << "\n\nRECORD NOT FOUND";
   }
   cin.get();
}
void delete_book()
{
   char n[6];
   //      clrscr();
   cout << "\n\n\n\tDELETE BOOK ...";
   cout << "\n\nEnter the book no. of the book you want to delete: ";
   cin >> n;
   fp.open("book.dat", ios::in | ios::out);
   fstream fp2;
   fp2.open("Temp.dat", ios::out);
   fp.seekg(0, ios::beg);
   while (fp.read((char *)&bk, sizeof(book)))
   {
```

```
        if (strcmp(bk.retbno(), n) != 0)
        {
            fp2.write((char *)&bk, sizeof(book));
        }
    }
    fp2.close();
    fp.close();
    remove("book.dat");
    rename("Temp.dat", "book.dat");
    cout << "\n\n\tRECORD DELETED ..";
    cin.get();
}

void display_alls()
{
    fp.open("student.dat", ios::in);
    if (!fp)
    {
        cout << "ERROR!!! FILE COULD NOT BE OPEN ";
        cin.get();
        return;
    }
    cout << "\n\n\t\tSTUDENT LIST\n\n";
    cout << "-----------------------------------------------\n";
    cout << "\tAdmissionNo.";
    cout << setw(10) << "Name" << setw(20) << "Book Issued\n";
    cout << "-----------------------------------------------\n";
    while (fp.read((char *)&st, sizeof(student)))
    {
        st.report();
    }
    fp.close();
    cin.get();
}

void display_allb()
{
    fp.open("book.dat", ios::in);
    if (!fp)
    {
        cout << "ERROR!!! FILE COULD NOT BE OPEN ";
        cin.get();
        return;
    }
    cout << "\n\n\t\tBOOK LIST\n\n";
    cout << "-----------------------------------------------\n";
    cout << "BookNumber" << setw(20) << "BookName" << setw(25);
    cout << "Author\n";
    cout << "-----------------------------------------------\n";
```

```
    while (fp.read((char *)&bk, sizeof(book)))
    {
        bk.report();
    }
    fp.close();
    cin.get();
}
void book_issue()
{
    char sn[6], bn[6];
    int found = 0, flag = 0;
    cout << "\n\nBOOK ISSUE ...";
    cout << "\n\n\tEnter the student's admission no.";
    cin >> sn;
    fp.open("student.dat", ios::in | ios::out);
    fp1.open("book.dat", ios::in | ios::out);
    while (fp.read((char *)&st, sizeof(student)) && found == 0)
    {
        if (strcmp(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 (strcmp(bk.retbno(), bn) == 0)
                    {
                        bk.show_book();
                        flag = 1;
                        st.addtoken();
                        st.getstbno(bk.retbno());
                        int pos = -1 * sizeof(st);
                        fp.seekp(pos, ios::cur);
                        fp.write((char *)&st, sizeof(student));
                        cout << "\n\n\t Book issued successfully";
                        cout << "\n\nPlease note: write current";
                        cout << "date in backside of book and";
                        cout << "submit within 15 days\n.";
                        cout << "Fine of Rs.1 for each day will be";
                        cout << "taken after 15 days period";
                    }
                }
                if (flag == 0)
                {
                    cout << "Book no. does not exist";
                }
            }
            else
            {
```

```
            cout << "You have not returned the last book.";
            cout << " Book can be issued only after returning the";
            cout << "previous one.";
          }
       }
       if (found == 0)
       {
           cout << "Student record not exist...";
       }
       cin.get();
       fp.close();
       fp1.close();
    }
}
void book_deposit()
{
    char sn[6], bn[6];
    int found = 0, flag = 0, day, fine;
    cout << "\n\nBOOK DEPOSIT ...";
    cout << "\n\n\tEnter the student's admission no.";
    cin >> sn;
    fp.open("student.dat", ios::in | ios::out);
    fp1.open("book.dat", ios::in | ios::out);
    while (fp.read((char *)&st, sizeof(student)) && found == 0)
    {
       if (strcmp(st.retadmno(), sn) == 0)
       {
          found = 1;
          if (st.rettoken() == 1)
          {
             while (fp1.read((char *)&bk, sizeof(book)) && flag == 0)
             {
                if (strcmp(bk.retbno(), st.retstbno()) == 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";
                      cout << 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";
                }
             }
```

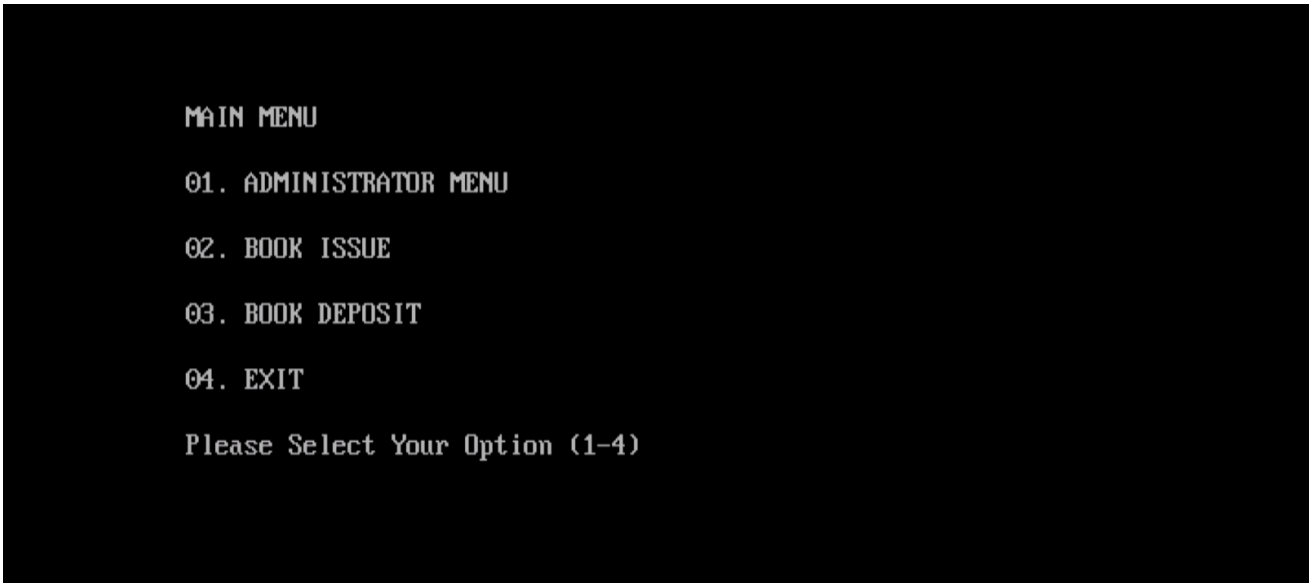
```
            if (flag == 0)
            {
               cout << "Book no does not exist";
            }
            else
            {
               cout << "No book is issued..please check!!";
            }
         }
      }
   }
   if (found == 0)
      cout << "Student record not exist...";
   cin.get();
  fp.close();
  fp1.close();
}
void intro()
{
   cout << "LIBRARY MANAGEMENT SYSTEM";
}
void admin_menu()
{
   //     clrscr();
   int ch2;
   char num[6];
   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:
      write_student();
      break;
   case 2:
      display_alls();
      break;
   case 3:
      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:
        write_book();
        break;
    case 7:
        display_allb();
        break;
    case 8:
        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();
}
int main()
{
    char ch;
     intro();
    do
    {
        cout << "\n\n\n\t MAIN MENU";
        cout << "\n\n\t01. ADMINISTRATOR MENU";
        cout << "\n\n\t02. BOOK ISSUE";
        cout << "\n\n\t03. BOOK DEPOSIT";
        cout << "\n\n\t04. EXIT";
        cout << "\n\n\tPlease Select Your Option (1-4): ";
        cin >> ch;
        switch (ch)
        {
        case '1':
            admin_menu();
            break;
        case '2':
```

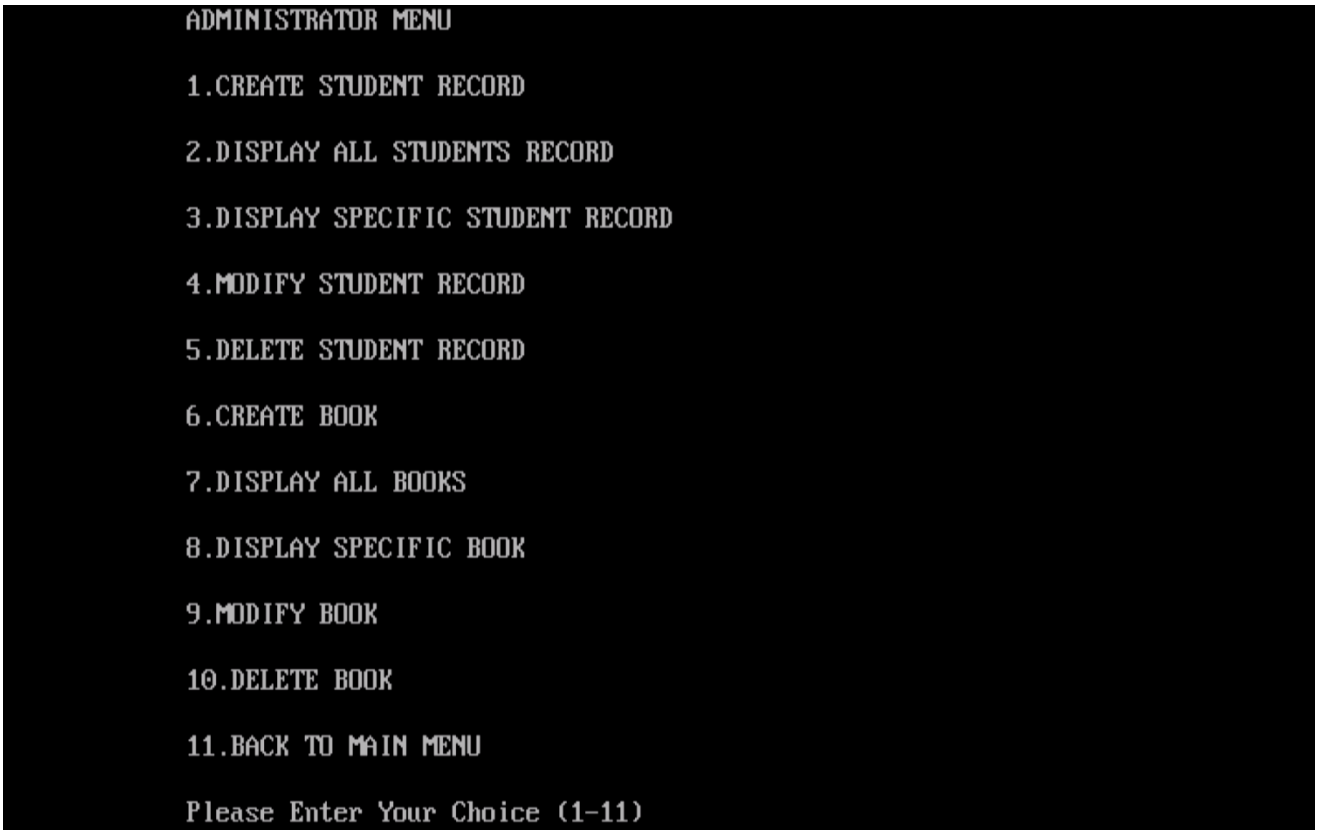
```
            book_issue();
            break;
        case '3':
            book_deposit();
            break;
        case '4':
            return 0;
        default:
            cout << "\a";
        }
    } while (ch != '4');
}
```

Working

1. Main Menu



2. Administrator Menu



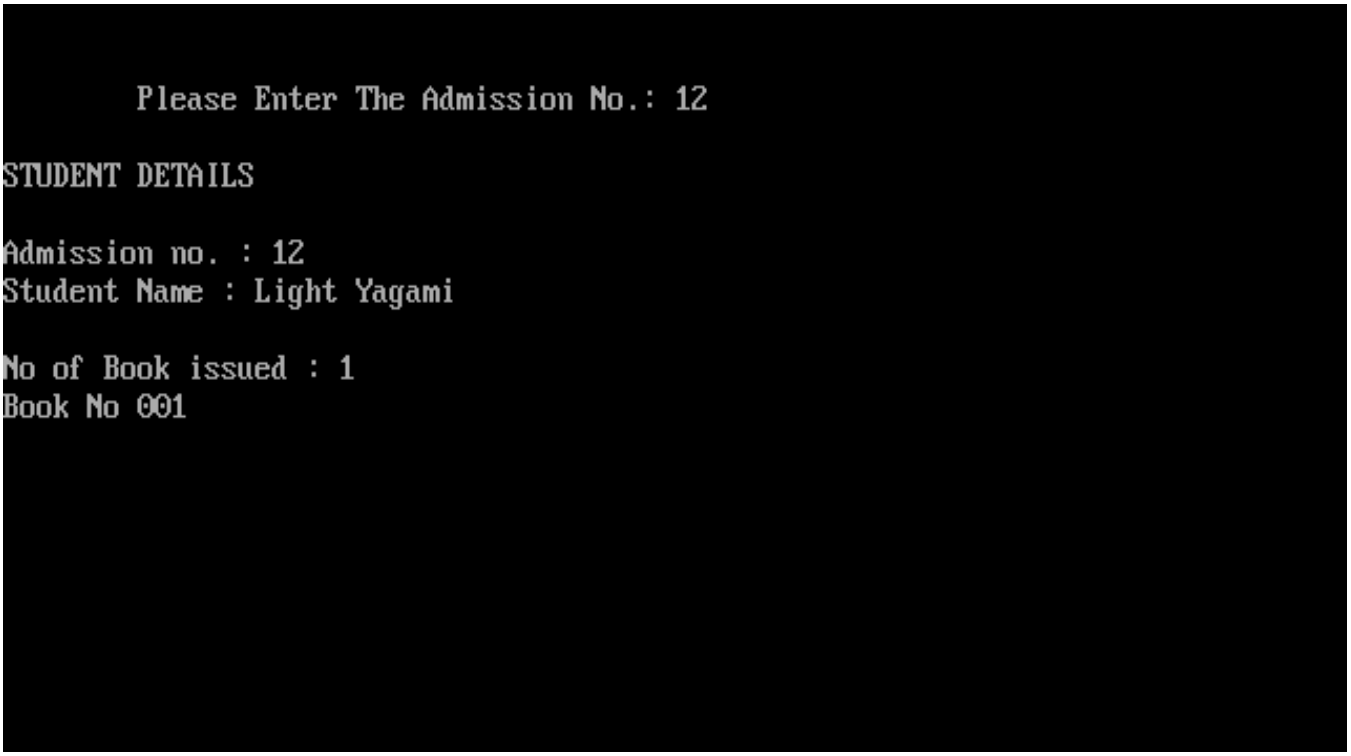
3. Creating a student record

```
NEW STUDENT ENTRY...

Enter The admission no.: 10

Enter The Name of The student: Amit Naglu

Student Record Created..

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

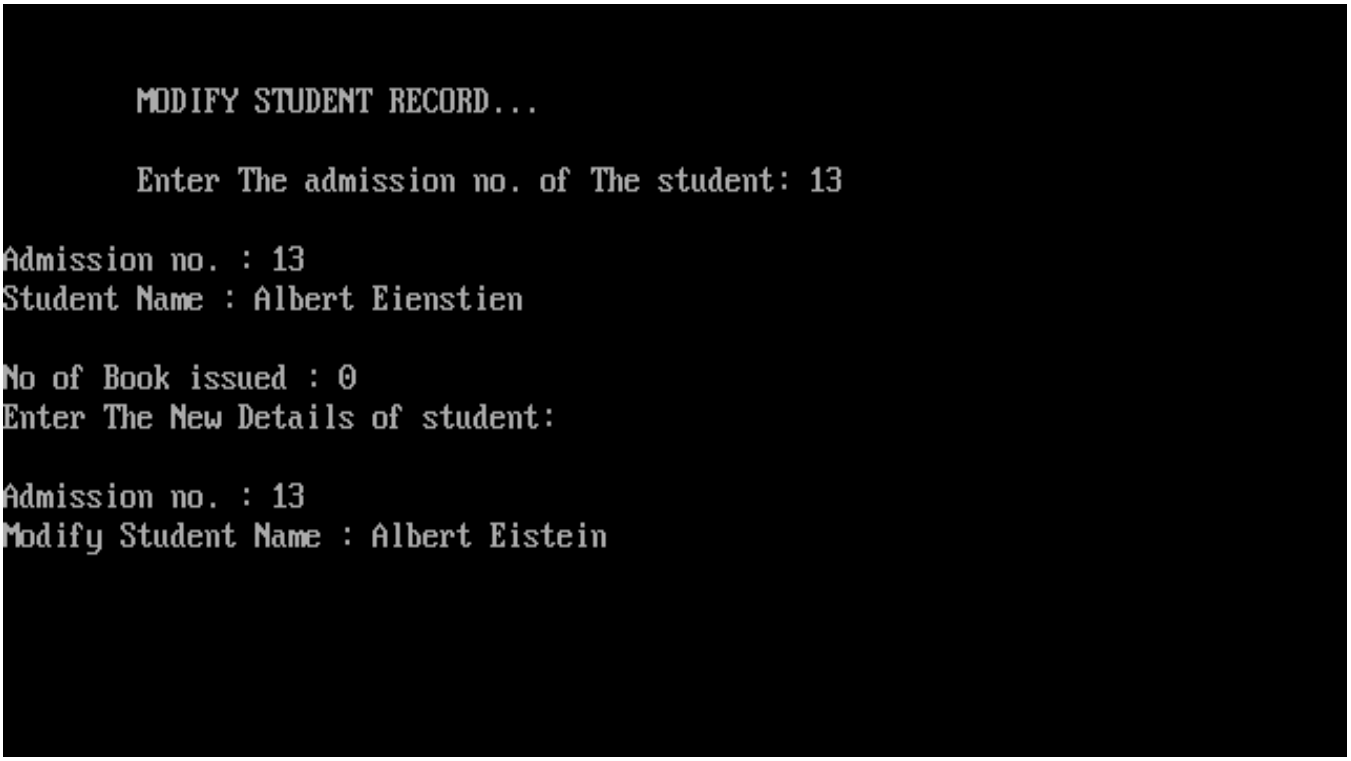
4. Display all the student's record

```
                STUDENT LIST

################################################################################
        AdmissionNo.      Name        Book Issued
################################################################################
        24              Priyansu          0
        22          Sudhanshu Dubey       0
        01           Kisuke Urahara       0
        02             Aizen Sosuke       0
        04        Sir Issac Newton        0
        05              Md. Wasif         0
        06            Aarti Kumari        1
        07       Gurleen Kaur Gill        0
        09         Aman Shrivastav        0
        10              Amit Naglu        0
        11              L Lawlight        0
        12            Light Yagami        0
        13          Albert Eistein        0
        14                Prajwal         0
        20         Zaraki Kenpachi        0
_
```

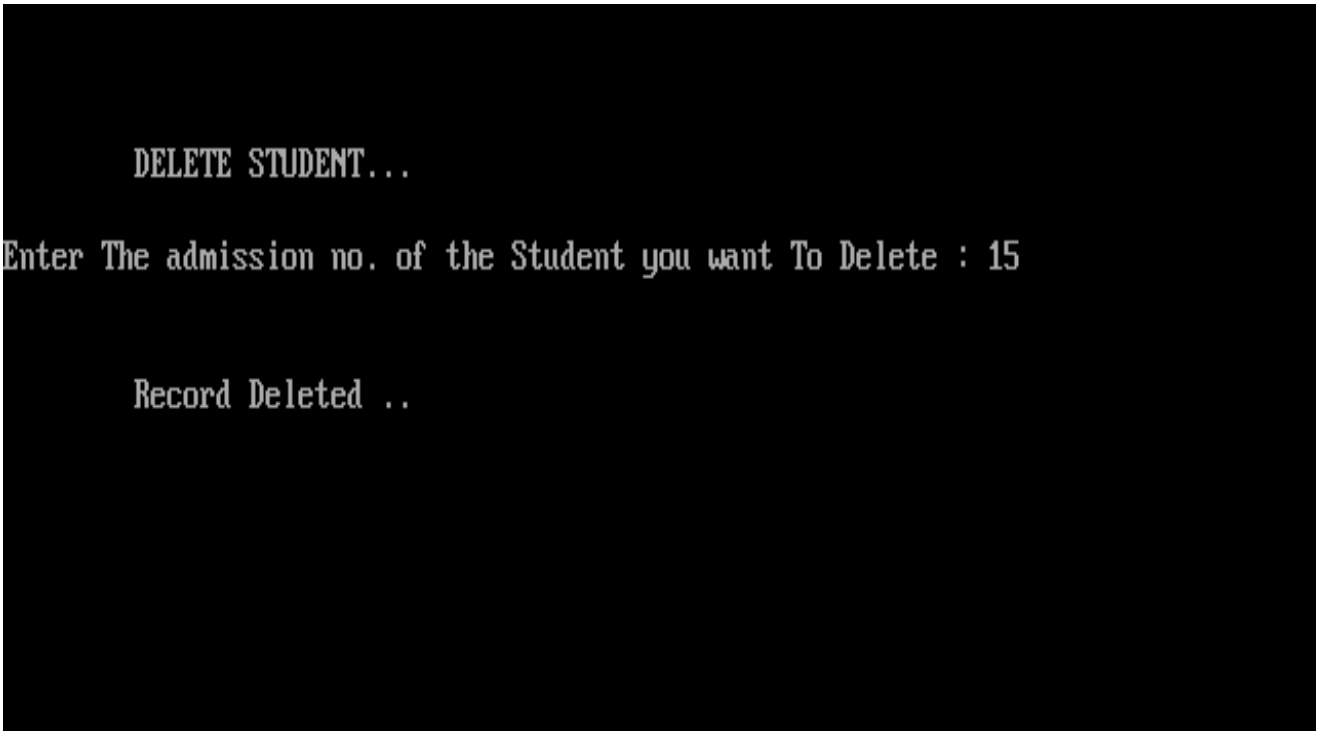
5. Displaying details of a specific student



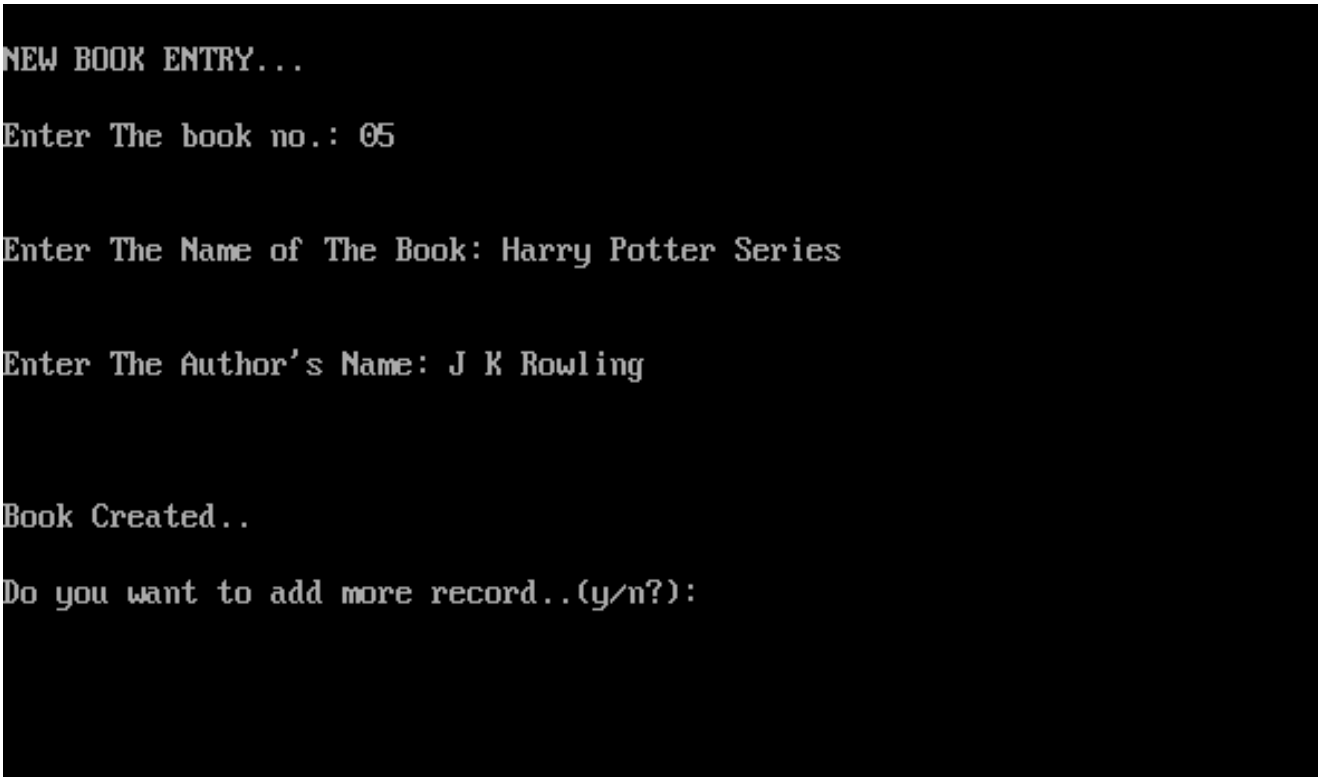
6. Modifying a student record



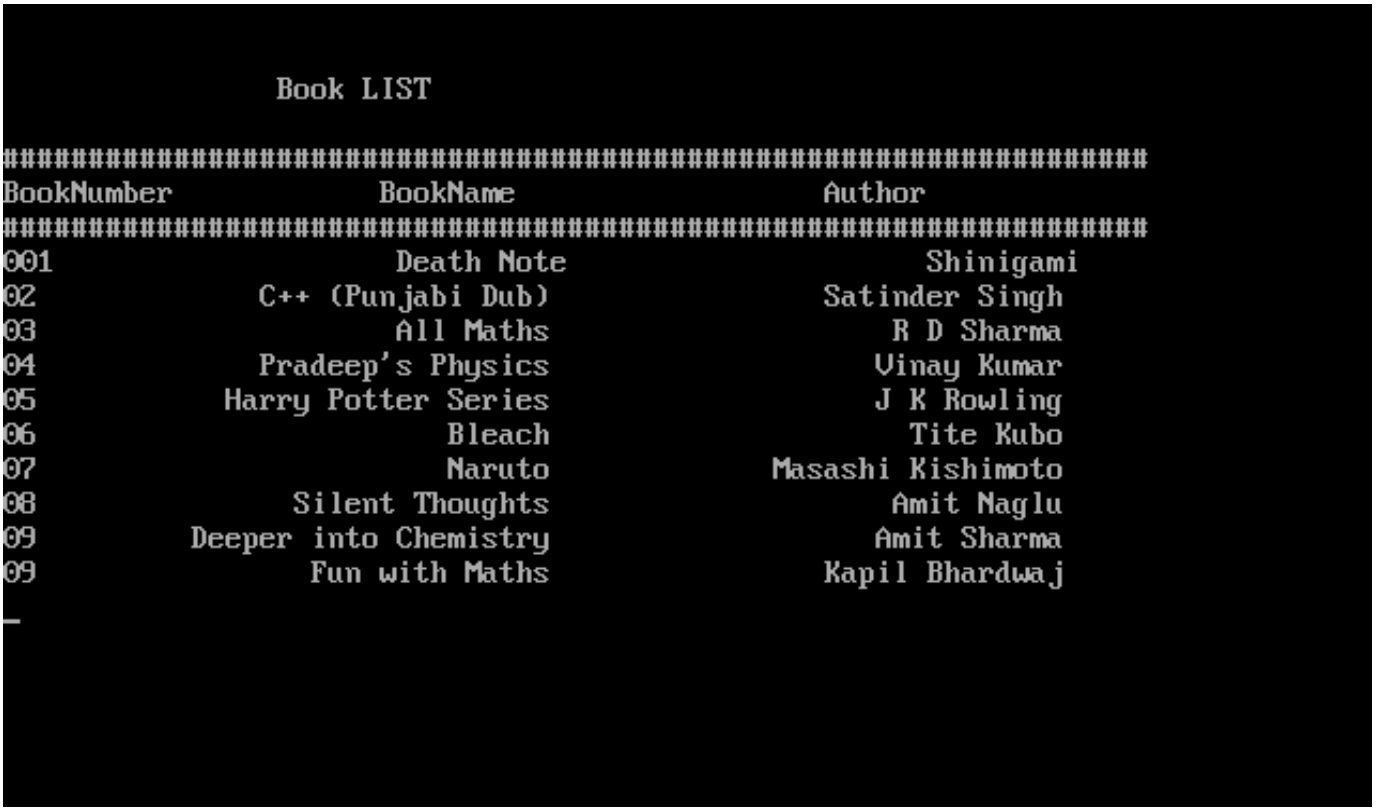
7. Deleting a student record



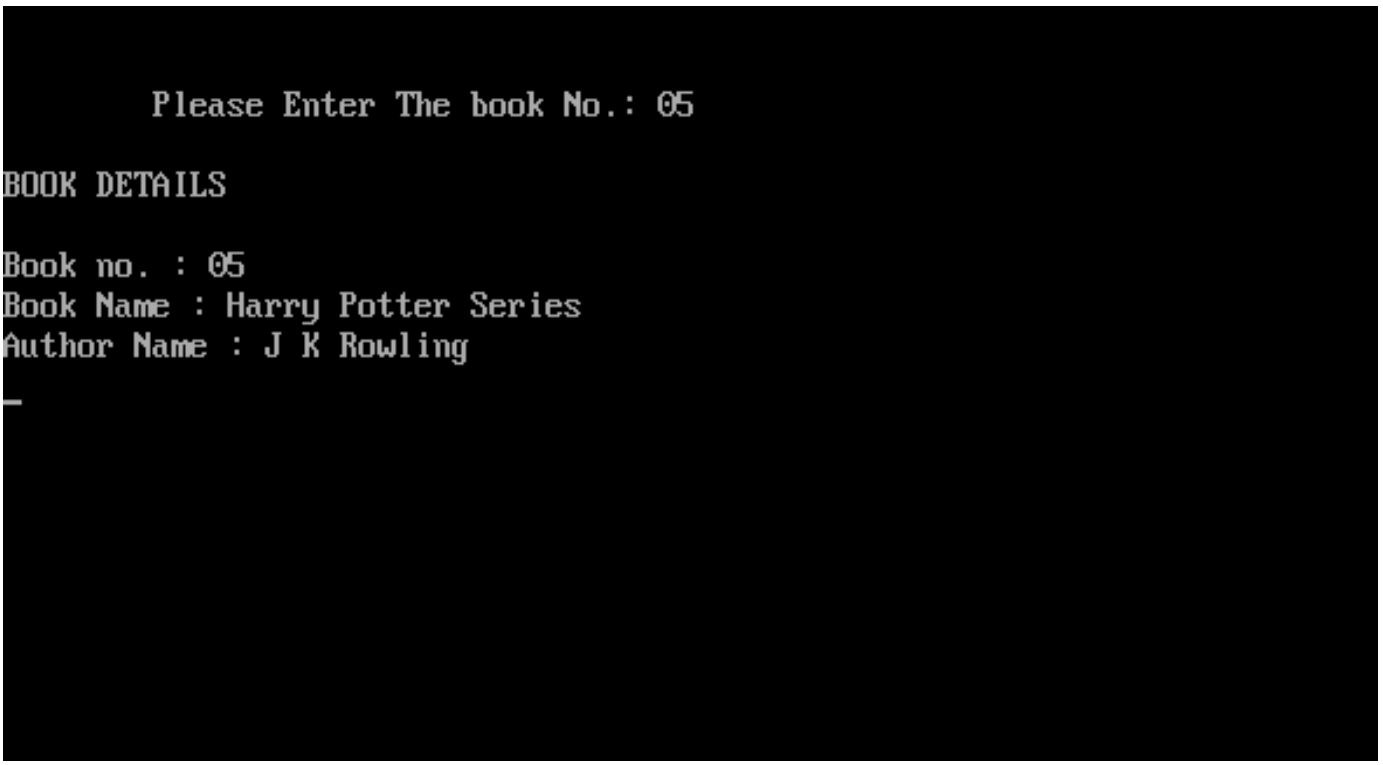
8. Entering a new book



9. Displaying all the books



10. Displaying details of a specific book



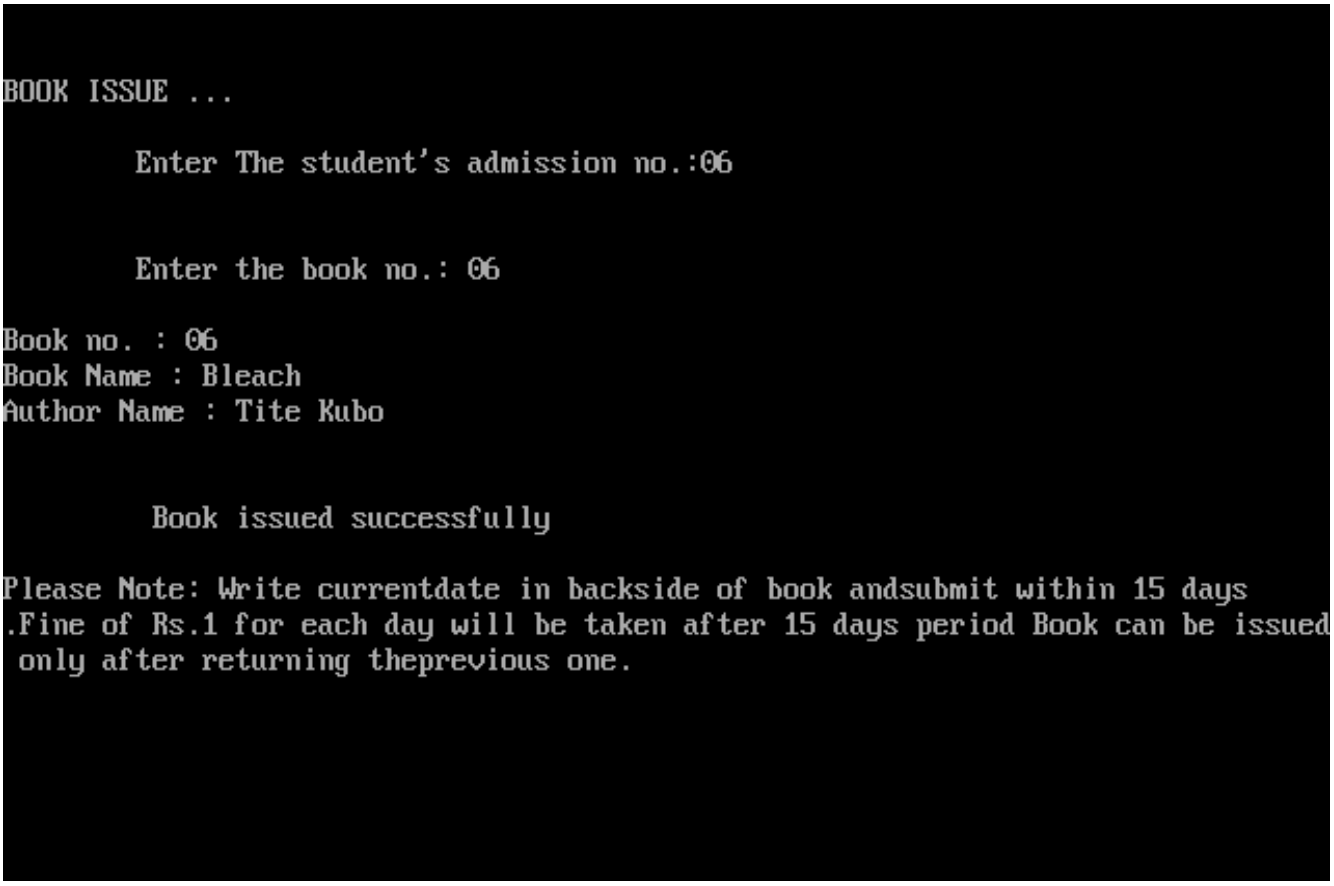
11. Modifying details of a book

```
        MODIFY BOOK REOCORD....

        Enter The book no. of The book: 03

Book no. : 03
Book Name : All Maths
Author Name : R D Sharma

Enter The New Details of book :

Book no. : 03
Modify Book Name : All about Maths

Modify Author's Name of Book:R. D. Sharma


         Record Updated.
```

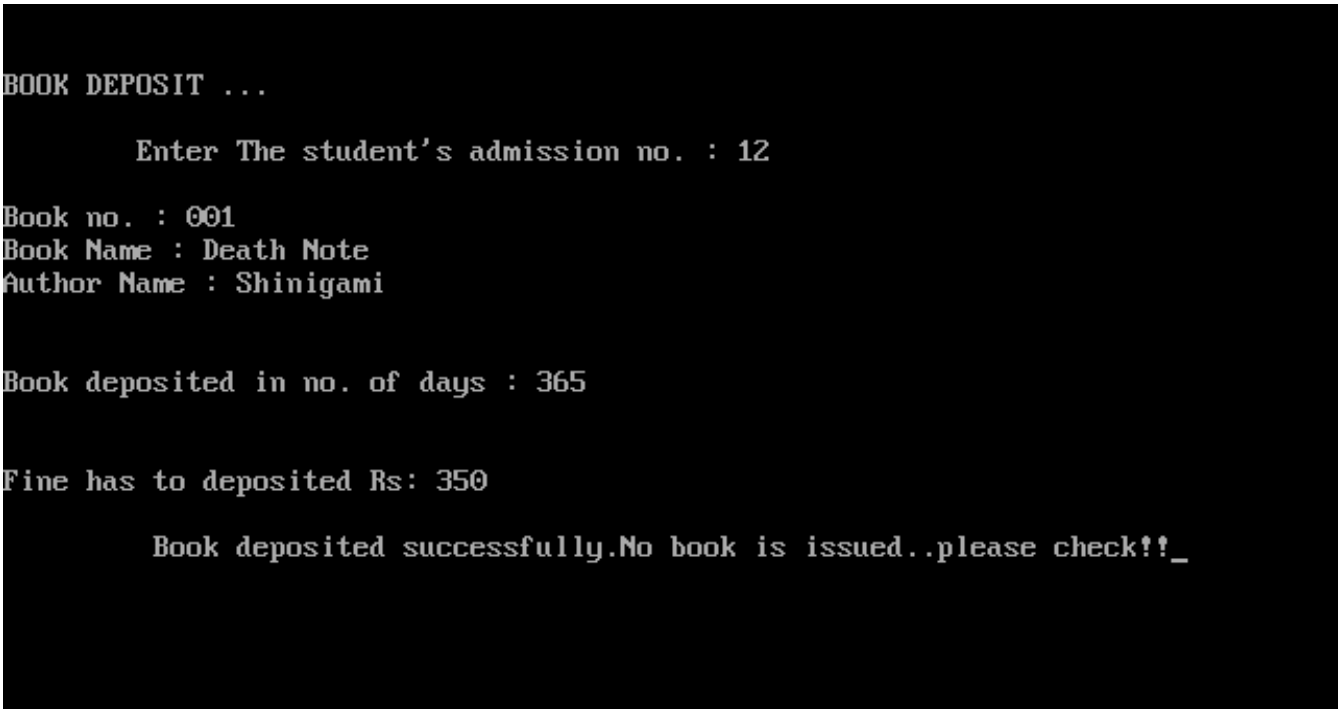
12. Deleting a book

```
        DELETE BOOK ...

Enter The Book no. of the Book You Want To Delete: 13

        Record Deleted ..
```

13. Issuing a book



14. Depositing a book



Conclusion

The Library Management System is a novel and interesting application. Unlike those commonly used methods of maintaining hardcopy of details and maintaining those records. User can interact with the software with menu-driven programs with user friendly interface. These properties allow our model to be accepted widely in school universities and public libraries. It makes the work easy and efficient. The proposed software will cover many aspects of time keeping and library process. The system has a file management where it covers the records of students and a good amount of data can be stored in the files. The system also covers the tracking of book in the library. A record can be added or deleted easily. Such advantages of this software have been verified empirically in various tasks including dues detection and fine imposition. The system is reliable and has a quickresponse time. The application can be used everywhere and it will reduce doing the same work over and over. The software has an effective and efficient way of monitoring their students to give a higher quality of service.

Reference

[E. Balagurusamy] "Object Oriented Programming With C++".

[geeksforgeeks]

https://www.geeksforgeeks.org/c-classes-and-objects/

https://www.geeksforgeeks.org/c-plus-plus/

https://www.geeksforgeeks.org/object-oriented-programming-in-cpp/