BOPAL
DELHI
PUBLIC
SCHOOL
AHMEDABAD

LIBRARY MANAGEMENT SYSTEM

NUPUR FUNKWAL
BOARD RNO:
ADMN NO. 10588
CLASS XII-Z
DPS BOPAL

BOPAL DELHI PUBLIC SCHOOL AHMEDABAD

CERTIFICATE

This is to certify that the Project work was satisfactorily carried out under the guidance of Dr.Deepak Pithadia and recorded in this file as a bonafide work of your name.

School : Delhi Public School, Bopal, Ahmedabad.
Class : XII - Computer Science (083)
Board Registration no. :
Academic year : 2017-18
Admission no. : 10588

Principal Subject Teacher Examiner

School Seal

INDEX

INTRODUCTION

Library is regarded as the brain of any institute; many institutes understand the importance of the library to the growth of the institute and their esteem users (students).   Library Management System supports this general requirement of the library.

Library project system that offers many flexible and convenient features, allowing librarians and library users to maximise time and efficiency. Library System gives the information about students, staff and books. It will track on the how many books available in library and books issued to the students. It keeps record of all students of the institution and also puts up a fine on the student who exceeds the time limit. It helps to create, modify and delete records from the database with ease.

ACKNOWLEDGEMENTS

I express my heartfelt gratitude to my Computer Teacher Shri Deepak Sir for the guidance and encouragement provided during the working on the project. His in depth understanding of the subject and methods of teaching helps the students achieve their best potential. I thank all my school teachers including Class Teacher Mrs Vibhuti Singh Madam and Principal Sri Surendra Sachdeva Sir for the constant motivation. I am also thankful to my parents for providing care, love and support.

HARDWARE AND SOFTWARE REQUIREMENTS

Device	MacBook Air
Operating System	MacOS Sierra
Processor	1.6 GHz Intel Core i5
Memory	8 GB 1600 MHz DDR3
Graphics.	Intel HD Graphics 6000 1536 M
Software.	Codeblocks

CLASSES AND THEIR ASSOCIATED FUNCTION NAMES

1. Book

It is an independent class.

Functions and constructors-

- Book()
- Book(string id,string n,string auth,int q)
- void show_book_details()
- void modify_book_details()
- string return_book_id()
- void generate_report()

2. User

It is a parent class of class Student.

Functions and constructors-

- User()
- User(string reg,string n)
- void display_user_details()
- void modify_user()
- string return_registration_number()
- string return_book_record()
- int return_book_issue_status()
- void add_book_issue_status()
- void reset_book_issue_status()
- void getbook_issued(string t)
- void generate_report()

3.Student

It is a publicly inherited class of class User.

Functions and constructors-

- Student()
- Student(string reg,string n):User(reg,n)

4.Teacher
Functions and constructors-

- Teacher()
- Teacher(string reg,string n):User(reg,n)

5. Admin

- Admin()
- void write_book_to_file()
- void write_student()
- void display_book_by_id(string)
- void display_specific_student(string)
- void modify_book_details()
- void modify_user()
- void delete_student()
- void delete_book()
- void display_all_students()
- void display_all_books()
- void book_issue()
- void book_deposit()
- void dashboard()

CODE

```
//***********************************************************
//                HEADER FILE USED IN PROJECT
//***********************************************************

#include<fstream>
#include<string>
#include<iostream>
#include<algorithm>
#include<string.h>

//***********************************************************
//                CLASS USED IN PROJECT
//************************************************************

using namespace std;


class Book
{
        string book_id;
        string name;
        string author;
        int quantity;

  public:
        Book()
        {
                book_id="";
                name="";
                author="";
                quantity=0;

        }

        Book(string id,string n,string auth,int q)
        {
                book_id=id;
                name=n;
                author=auth;
                quantity=q;
        }

        void show_book_details()
        {

                cout<<"\nBook Name : \t"<<name;
                cout<<"\nAuthor's Name : \t"<<author;
                cout<<"\nBook ID : \t"<<book_id;
                cout<<"\nQuantity : \t"<<quantity;
        }

        void modify_book_details()
        {

                cout<<"\nEnter New Book Name : \t";
                cin>>name;
                cout<<"\nModify Author's Name of Book : \t";
                cin>>author;
                cout<<"\nEnter the New Book ID \t";
                cin>>book_id;
                cout<<"\nEnter the New Quantity \t";
                cin>>quantity;
        }

        string return_book_id()
        {
                return book_id;
        }

        void generate_report()
        {
                cout<<book_id<<"\t"<<name<<"\t"<<author<<"\t"<<endl;


        }

};

class User
{

protected:
        string registration_number;
        string name;
        string book_issued;
        int token;
public:
        User()
        {
                name="";
                registration_number="";
                token=0;
                book_issued="";
        }
```

```
User(string reg,string n)
{
	registration_number=reg;
	name=n;
	token=0;
	book_issued="";
	cout<<"\n\nStudent Record Successfully Created..\n\n";
}

void display_user_details()
{
	cout<<"\nRegistration Number : \t"<<registration_number;
	cout<<"\nStudent Name : \t"<<name;
	cout<<"\nNo of Book issued : \t"<<token;
	if(token==1)
		cout<<"\nBook No \t"<<book_issued;
}

void modify_user()
{
	cout<<"\nEnter New Registration Number : \t";
	cin>>registration_number;
	cout<<"\nEnter Student's New Name : \t";
	cin>>name;
	cout<<"\n Record Modified Successfully!!!\n\n";
}

string return_registration_number()
{
	return registration_number;
}

string return_book_record()
{
	return book_issued;
}

int return_book_issue_status()
{
	return token;
}

void add_book_issue_status()
{
	token=1;
}
```

```
	void reset_book_issue_status()
	{
		token=0;
		book_issued="";
	}

	void getbook_issued(string t)
	{
		book_issued=t;
	}

	void generate_report()
	{
		cout<<registration_number<<"\t"<<name<<"\t
"<<return_book_record()<<endl;

	}


};


class Student: public User
{
public:
	Student()
	{

	}

	Student(string reg,string n):User(reg,n)
	{

	}


};


class Teacher: public User
{

public:
	Teacher()
	{

	}
```

```
    Teacher(string reg,string n):User(reg,n)
    {

    }


};


class Admin
{
    string name;

public:
     Admin()
     {
         name="Admin";
     }

    void write_book_to_file();
    void write_student();
    void display_book_by_id(string);
    void display_specific_student(string);
    void modify_book_details();
    void modify_user();
    void delete_student();
    void delete_book();
    void display_all_students();
    void display_all_books();
    void book_issue();
    void book_deposit();
    void dashboard();


};


//**************************************************************
//      function to write in file
//***************************************************************

void Admin::write_book_to_file()
{
    ofstream book_fp;
    char ch;
    string id,name,auth;
    int quan;
    book_fp.open("book.txt",ios_base::app);
```

```
        do
        {
                cout<<"\nEnter Book ID\t";
                cin>>id;
                cout<<"\nEnter Name of The Book\t";
                cin>>name;
                cout<<"\nEnter The Author's Name\t";
                cin>>auth;
                cout<<"\nEnter Quantity of the Book\t";
                cin>>quan;
                Book bk(id,name,auth,quan);
                book_fp.write((char*)&bk,sizeof(bk));
                cout<<"\n\nDo you want to add more record..(y/n?)";
                cin>>ch;

        }while(ch=='y'||ch=='Y');
        book_fp.close();
}

void Admin::write_student()
{
        ofstream student_fp;
        char ch;
        string reg,name;
        student_fp.open("student.txt",ios_base::app);
        do
        {
                cout<<"\nEnter Student Registration Number\n";
                cin>>reg;
                cout<<"\nEnter Student Name\n";
                cin>>name;
                Student st(reg,name);

                student_fp.write((char*)&st,sizeof(st));
                cout<<"\n\nDo you want to add one more Student..(y/n?)";
                cin>>ch;
        }while(ch=='y'||ch=='Y');
        student_fp.close();
        cout<<"\n ...Updating Database...\n\n";
}


//************************************************************
//      function to read specific record from file
//*************************************************************


void Admin::display_book_by_id(string n)
```

```
{
        Book bk;
        ifstream book_fp;
        cout<<"\nBOOK DETAILS\n";
        int flag=0;
        book_fp.open("book.txt",ios::in);
        while(book_fp.read((char*)&bk,sizeof(Book)))
        {
                if(bk.return_book_id().compare(n)==0)
                {
                        bk.show_book_details();
                        flag=1;
                }
        }

        book_fp.close();
        if(flag==0)
                cout<<"\n\nBook does not exist";

}

void Admin::display_specific_student(string n)
{
        Student st;
        ifstream student_fp;
        cout<<"\nSTUDENT DETAILS\n";
        int flag=0;
        student_fp.open("student.txt",ios::in);
        while(student_fp.read((char*)&st,sizeof(st)))
        {
                if(st.return_registration_number().compare(n)==0)
                {
                        st.display_user_details();
                        flag=1;
                }
        }


        if(flag==0)
                cout<<"\n\nStudent does not exist\n";
    student_fp.close();

}


//***************************************************************
//      function to modify record of file
//****************************************************************
```

```
void Admin::modify_book_details()
{
        string n;
        int found=0;
        Book bk;
        ifstream book_fp;
        ofstream book_write;
        book_write.open("book_temp.txt",ios::out);
        cout<<"\n\n\t...MODIFY BOOK RECORD...";
        cout<<"\n\n\tEnter The book no. of The book";
        cin>>n;
        book_fp.open("book.txt",ios::inlios::out);
        while(book_fp.read((char*)&bk,sizeof(bk)))
        {
                if(bk.return_book_id().compare(n)==0)
                {
                        bk.show_book_details();
                        cout<<"\nEnter The New Details of book"<<endl;
                        bk.modify_book_details();

                book_write.write((char*)&bk,sizeof(bk));
                cout<<"\n\n\t Record Updated";
                found=1;
                }
                else{
                        book_write.write((char*)&bk,sizeof(bk));
                }
        }

        book_fp.close();
        book_write.close();
        remove("book.txt");
        rename("book_temp.txt","book.txt");
        if(found==0)
                cout<<"\n\n Record Not Found ";

}


void Admin::modify_user()
{
        string n;
        int found=0;
        Student st;
        ifstream student_fp;
        ofstream student_write;
```

```
	student_write.open("student_temp.txt",ios::out);

	cout<<"\n\n\tMODIFY STUDENT RECORD... ";
	cout<<"\n\n\tEnter The Registration Number of the Student";
	cin>>n;
	student_fp.open("student.txt",ios::in);
	while(student_fp.read((char*)&st,sizeof(st)))
	{
		if(st.return_registration_number().compare(n)==0)
		{
			st.display_user_details();
			// student_fp.seekg(0,ios::cur);
			cout<<"\nEnter The New Details of Student"<<endl;
			st.modify_user();
			// int pos=-1*sizeof(st);
			// student_fp.seekp(pos,ios::cur);
			student_write.write((char*)&st,sizeof(st));
			cout<<"\n\n\t Record Updated\n\n";
			found=1;
		}
		else
		{
			student_write.write((char*)&st,sizeof(st));

		}


	}

	student_fp.close();
	student_write.close();
	remove("student.txt");
	rename("student_temp.txt","student.txt");
	if(found==0)
		cout<<"\n\n Record Not Found ";

}

//************************************************************
//	function to delete record of file
//*************************************************************


void Admin::delete_student()
{
	string n;
	int flag=0;
```

```
        Student st;
        fstream student_fp,student_fp2;

        cout<<"\n\n\n\t...DELETE STUDENT...";
        display_all_students();
        cout<<"\n\nEnter The Registration Number of the Student : ";
        cin>>n;
        student_fp.open("student.txt",ios::in|ios::out);
        student_fp2.open("Temp.txt",ios::out);
        student_fp.seekg(0,ios::beg);
        while(student_fp.read((char*)&st,sizeof(st)))
        {
                if(st.return_registration_number().compare(n)!=0)
                        student_fp2.write((char*)&st,sizeof(st));
                else
                        flag=1;
        }

        student_fp2.close();
        student_fp.close();
        remove("student.txt");
        rename("Temp.txt","student.txt");
        if(flag==1)
                cout<<"\n\n\tRecord Deleted Successfully";
        else
                cout<<"\n\nRecord not found";

}


void Admin::delete_book()
{
        string n;
        Book bk;
        ifstream book_fp;
        ofstream book_write;
        cout<<"\n\n\n\tDELETE BOOK ...";
        display_all_books();
        cout<<"\n\nEnter The Book no. of the Book You Want To Delete : ";
        cin>>n;
        book_fp.open("book.txt",ios::in);
        book_write.open("Temp.txt",ios::out);
        // book_fp.seekg(0,ios::beg);
        while(book_fp.read((char*)&bk,sizeof(bk)))
        {
                if(bk.return_book_id().compare(n)!=0)
                {
                        book_write.write((char*)&bk,sizeof(bk));
        }
    }

    book_write.close();
    book_fp.close();
    remove("book.txt");
    rename("Temp.txt","book.txt");
    cout<<"\n\n\tBook Record Deleted Successfully!!!";

}



//**********************************************************
//      function to display all students list
//***********************************************************

void Admin::display_all_students()
{
        ifstream student_fp;
    student_fp.open("student.txt",ios::in);
    if(!student_fp)
    {
        cout<<"ERROR OPENING FILE ";

        return;
    }
  Student st;

    cout<<"\n\n\t\tALL STUDENT DETAILS\n\n";

cout<<"=============================================================
=======\n";
    cout<<"Reg No.  "<<"Name  "<<" Book Issued\n";

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

    while(student_fp.read((char*)&st,sizeof(st)))
    {
        st.generate_report();
    }

    student_fp.close();

}
```

```
//*******************************************************************
//      function to display Books list
//********************************************************************

void Admin::display_all_books()
{
        Book bk;
        ifstream book_fp;
        book_fp.open("book.txt",ios::in);
        if(!book_fp)
        {
                cout<<"ERROR OPENING THE FILE ";

                return;
        }

        cout<<"\n\n\t\tList of all the Books\n\n";

cout<<"====================================================================
==============\n";
        cout<<"Book ID"<<"  "<<"Book Name"<<" "<<"Author\n";

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

        while(book_fp.read((char*)&bk,sizeof(Book)))
        {
                bk.generate_report();
        }

   book_fp.close();

}



//*******************************************************************
//      function to issue book
//********************************************************************

void Admin::book_issue()
{
        string sn,bn;
        int found=0,flag=0;
        Student st;
        Book bk;
        ifstream student_fp,book_fp;
        ofstream student_write;
```

```
    student_write.open("student_temp.txt",ios_base::out);
    cout<<"\n\n...BOOK ISSUE ...";
    display_all_students();
    cout<<"\n\n\tEnter Student's Registration number\t";
    cin>>sn;
    student_fp.open("student.txt",ios::in);
    book_fp.open("book.txt",ios::in);
    while(student_fp.read((char*)&st,sizeof(st)))//read student
    {
        if(st.return_registration_number().compare(sn)==0)//check if the same
student
        {
            found=1;
            if(st.return_book_issue_status()==0)//check if book issued?
            {
                display_all_books();
           cout<<"\n\n\tEnter the Book ID \t";
                cin>>bn;
                while(book_fp.read((char*)&bk,sizeof(bk)))//read all books
                {
                    if(bk.return_book_id().compare(bn)==0)//if book found
                    {
                        bk.show_book_details();
                        flag=1;
                        st.add_book_issue_status();//issued or not?
                        st.getbook_issued(bk.return_book_id());//book
id return. book_issued keeps track of book issued.
  //                    int pos=-1*sizeof(st);
                        // student_fp.seekp(pos,ios::cur);
                        student_write.write((char*)&st,sizeof(st));
                        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 ID does not exist";
                    student_write.write((char*)&st,sizeof(st));
                }
            }
        else{
            student_write.write((char*)&st,sizeof(st));
                cout<<"Issued Book  ";
        }

        }
```

```
                else{

                        student_write.write((char*)&st,sizeof(st));
                }
        }
        student_fp.close();
        book_fp.close();
        student_write.close();
        remove("student.txt");
        rename("student_temp.txt","student.txt");

        if(found==0)
        cout<<"Student record not exist...";


}

//*************************************************************
//      function to deposit book
//**************************************************************

void Admin::book_deposit()
{
    string sn,bn;
    int found=0,flag=0,day,fine;
    Student st;
    Book bk;
    ifstream student_fp,book_fp;
    ofstream student_write;
    student_write.open("student_temp.txt",ios_base::app);

    cout<<"\n\n...BOOK DEPOSIT ...";
    display_all_students();
    cout<<"\n\n\tEnter The student's Registration Number\t";
    cin>>sn;
    student_fp.open("student.txt",ios::in);
    book_fp.open("book.txt",ios::in);
    while(student_fp.read((char*)&st,sizeof(st)))//read each student
      {
            if(st.return_registration_number().compare(sn)==0)//compare reg no
            {
                    found=1;
                    if(st.return_book_issue_status()==1)//book issued or not?
                    {
                                while(book_fp.read((char*)&bk,sizeof(bk)))//find the issued
book
                                {
```

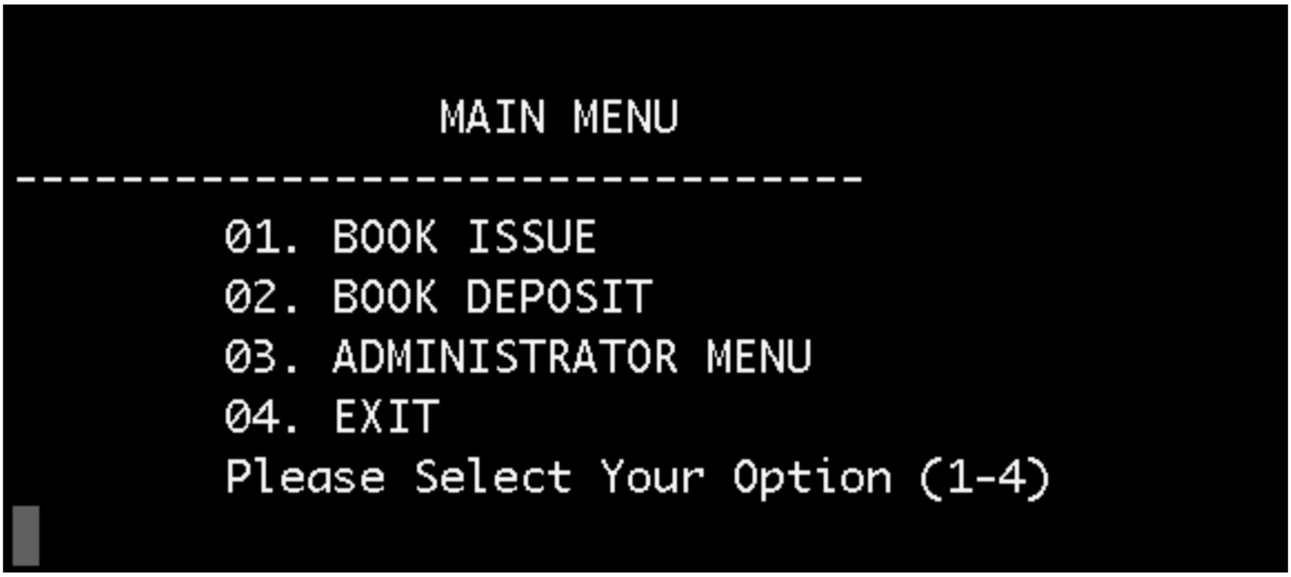
```
if(bk.return_book_id().compare(st.return_book_record())==0)//if match
                                        {
                                                bk.show_book_details();
                                                flag=1;
                                                cout<<"\n\nBook deposited in no. of days\t";
                                                cin>>day;
                                                if(day>15)
                                                {
                                                  fine=(day-15)*1;
                                                  cout<<"\n\nFine has to deposited Rs.
\t"<<fine;

                                                }
                                                st.reset_book_issue_status();//sets token = 0
                                                // int pos=-1*sizeof(st);
                                                // student_fp.seekp(pos,ios::cur);
                                                student_write.write((char*)&st,sizeof(st));
                                                cout<<"\n\n\t Book deposited successfully";
                                        }
                             }
                                if(flag==0)
                                        cout<<"Book no does not exist";
                     }
                        else
                        {
                                cout<<"No book is issued..Please check!!";
                                student_write.write((char*)&st,sizeof(st));

                        }
                }
                else
                {
                        student_write.write((char*)&st,sizeof(st));
                }

            }
    if(found==0)
        cout<<"Student record not exist...";

 student_fp.close();
 book_fp.close();
 student_write.close();

 remove("student.txt");
 rename("student_temp.txt","student.txt");
 }
```

```
//***************************************************************
//      ADMINISTRATOR MENU FUNCTION
//****************************************************************

void Admin::dashboard()
{

        int ch2;
        cout<<"\n--------------------------------";
        cout<<"\n\n\t\tADMIN DASHBOARD";
        cout<<"\n--------------------------------";
        cout<<"\n\t1.CREATE STUDENT RECORD";
        cout<<"\n\t2.DISPLAY ALL STUDENTS RECORD";
        cout<<"\n\t3.DISPLAY SPECIFIC STUDENT RECORD ";
        cout<<"\n\t4.MODIFY STUDENT RECORD";
        cout<<"\n\t5.DELETE STUDENT RECORD";
        cout<<"\n\t6.CREATE BOOK ";
        cout<<"\n\t7.DISPLAY ALL BOOKS ";
        cout<<"\n\t8.DISPLAY SPECIFIC BOOK ";
        cout<<"\n\t9.MODIFY BOOK ";
        cout<<"\n\t10.DELETE BOOK ";
        cout<<"\n\t11.BACK TO MAIN MENU";
        cout<<"\n\tPlease Enter Your Choice (1-11)\t";
        cin>>ch2;
        string num;
        switch(ch2)
        {
                case 1:
                        write_student();
                        break;
                case 2:
                        display_all_students();
                        break;
                case 3:
                        cout<<"\n\n\tPlease Enter The Registration Number ";
                        cin>>num;
                        display_specific_student(num);
                        break;
                case 4:
                        modify_user();
                        break;
                case 5:
                        delete_student();
                        break;
                        case 6:
                                write_book_to_file();
```

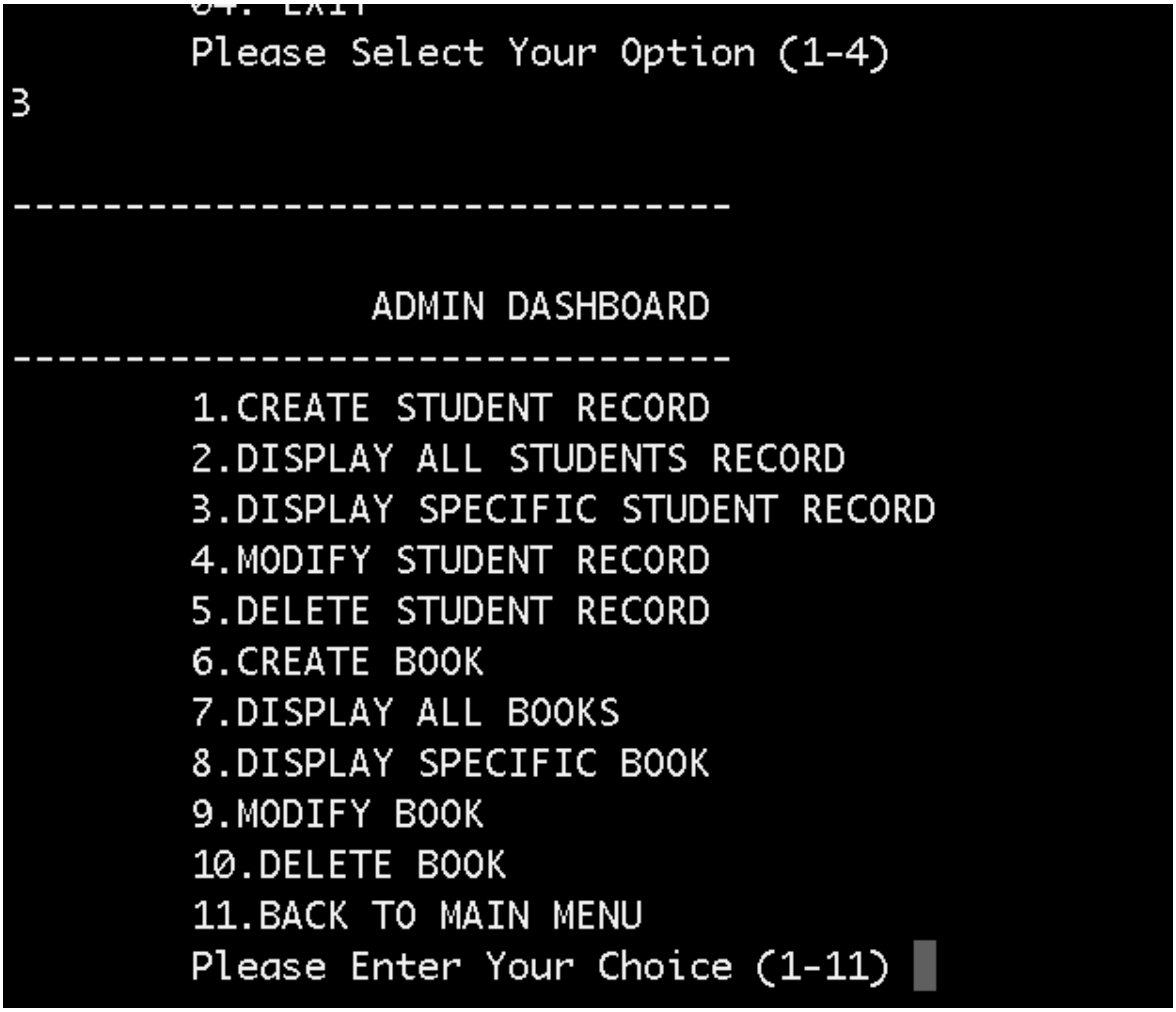
```
				break;
			case 7:
				display_all_books();
				break;
			case 8: {

				cout<<"\n\n\tPlease Enter Book ID ";
				cin>>num;
				display_book_by_id(num);
				break;
				}
		case 9:
			modify_book_details();
			break;
		case 10:
			delete_book();
			break;
		case 11:
			return;
		default:cout<<"\a";
	}
	dashboard();
}


//*************************************************************
//	THE MAIN FUNCTION OF PROGRAM
//**************************************************************


int main()
{
	Admin admin;
	char ch;
	do
	{

		cout<<"\n-------------------------------";
		cout<<"\n\n\t\tMAIN MENU";
		cout<<"\n-------------------------------";
		cout<<"\n\t01. BOOK ISSUE";
		cout<<"\n\t02. BOOK DEPOSIT";
		cout<<"\n\t03. ADMINISTRATOR MENU";
		cout<<"\n\t04. EXIT";
		cout<<"\n\tPlease Select Your Option (1-4)\n";
		ch=getchar();
		switch(ch)
		{
```

```
                        case '1':
                                admin.book_issue();
                                break;
                        case '2':
                                admin.book_deposit();
                            break;
                        case '3':
                                admin.dashboard();
                                break;
                        case '4':
                                exit(0);
                        default :cout<<"\a";
                }
    }while(ch!='4');
}

//***********************************************************
//                      END OF PROJECT
//***********************************************************
```

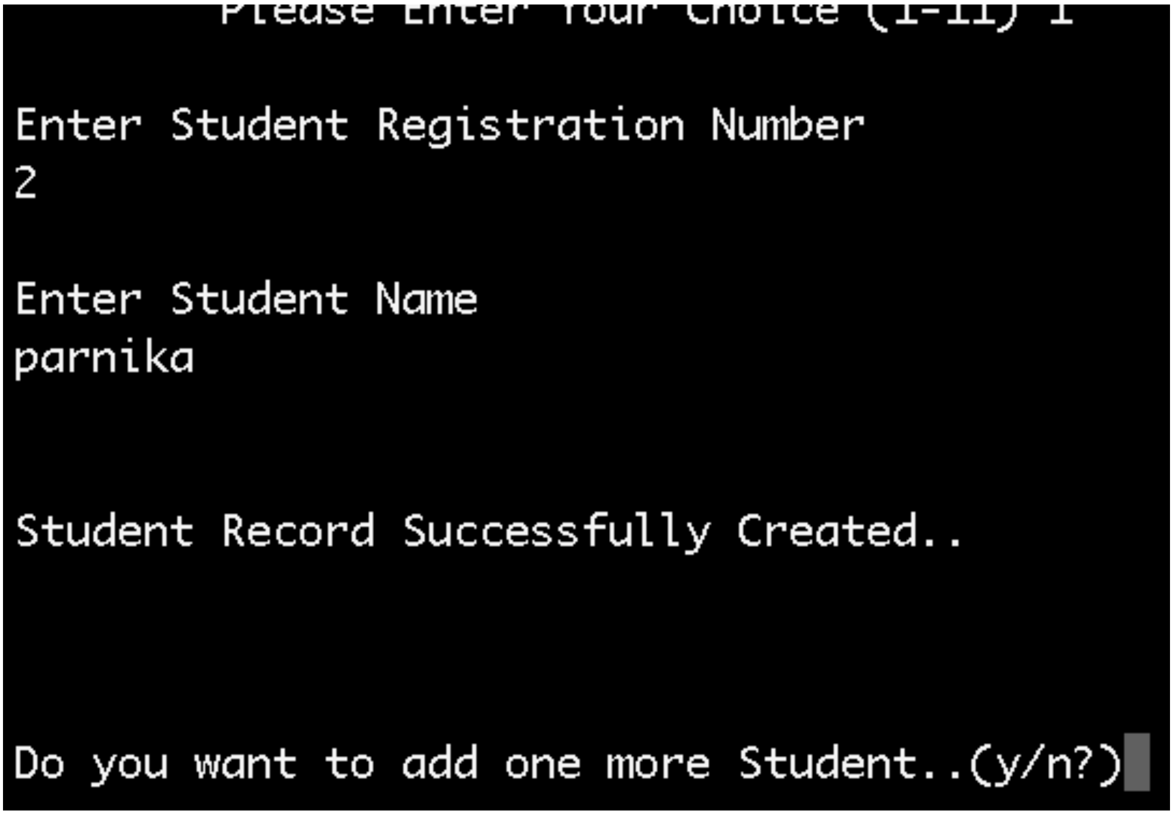
OUTPUT SCREENSHOTS



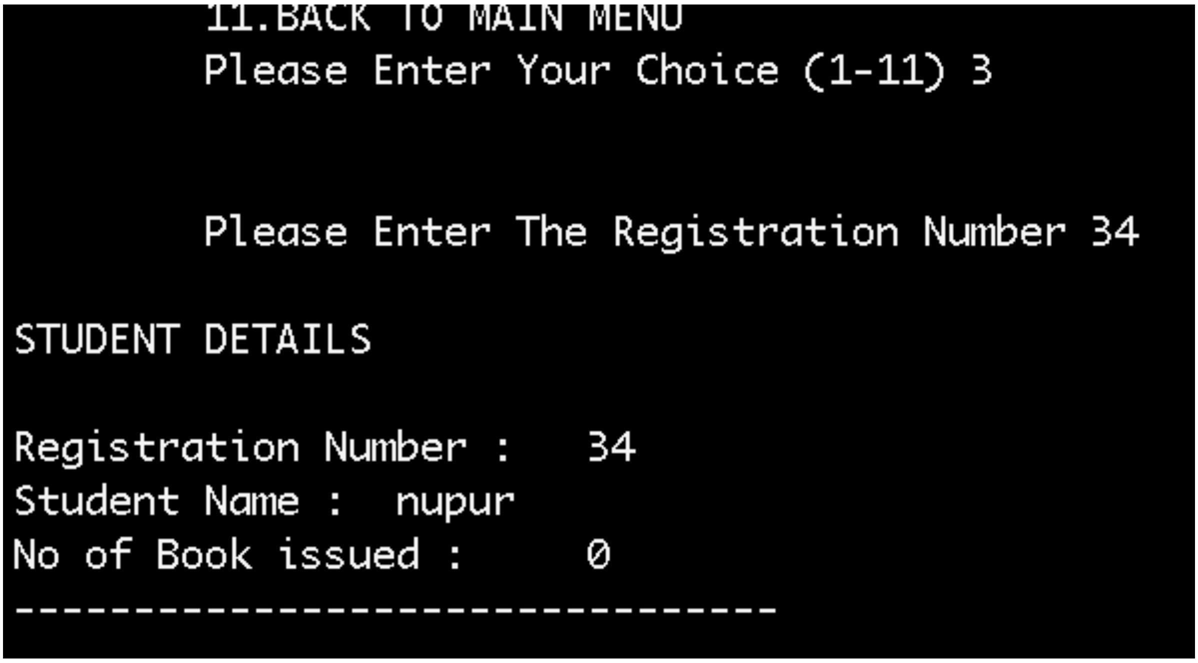
```
                    MAIN MENU
---------------------------------------
          01. BOOK ISSUE
          02. BOOK DEPOSIT
          03. ADMINISTRATOR MENU
          04. EXIT
          Please Select Your Option (1-4)
```



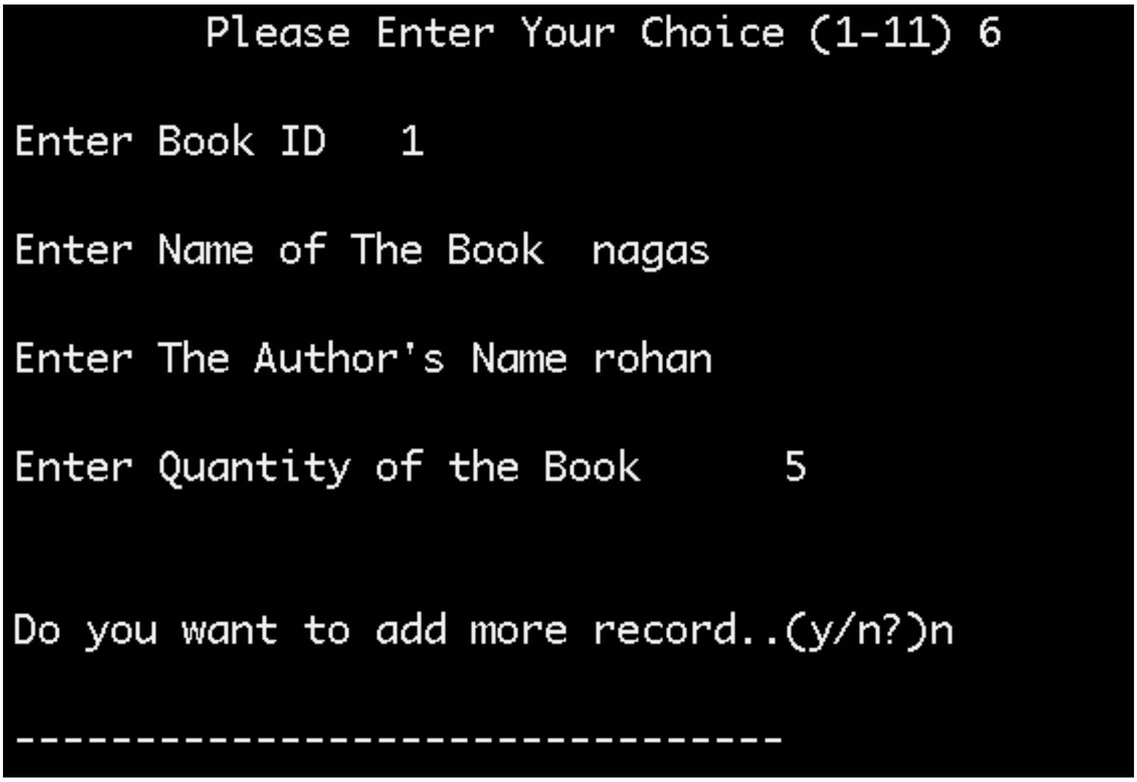
```
          04. EXIT
          Please Select Your Option (1-4)
3


------------------------------------


                   ADMIN DASHBOARD
------------------------------------
          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)
```



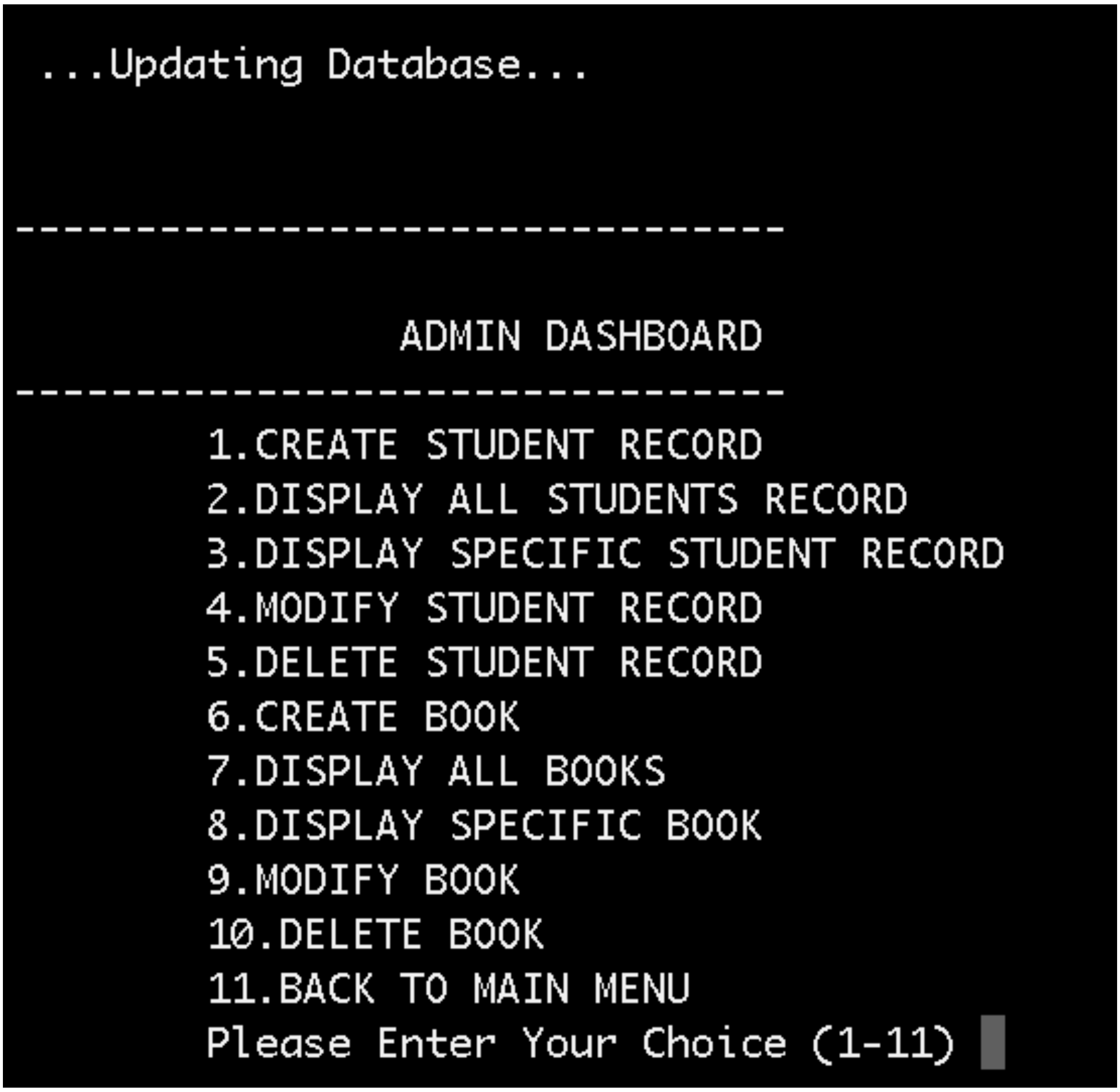
```
          Please Enter Your Choice (1-11) 1

Enter Student Registration Number
2

Enter Student Name
parnika


Student Record Successfully Created..



Do you want to add one more Student..(y/n?)
```



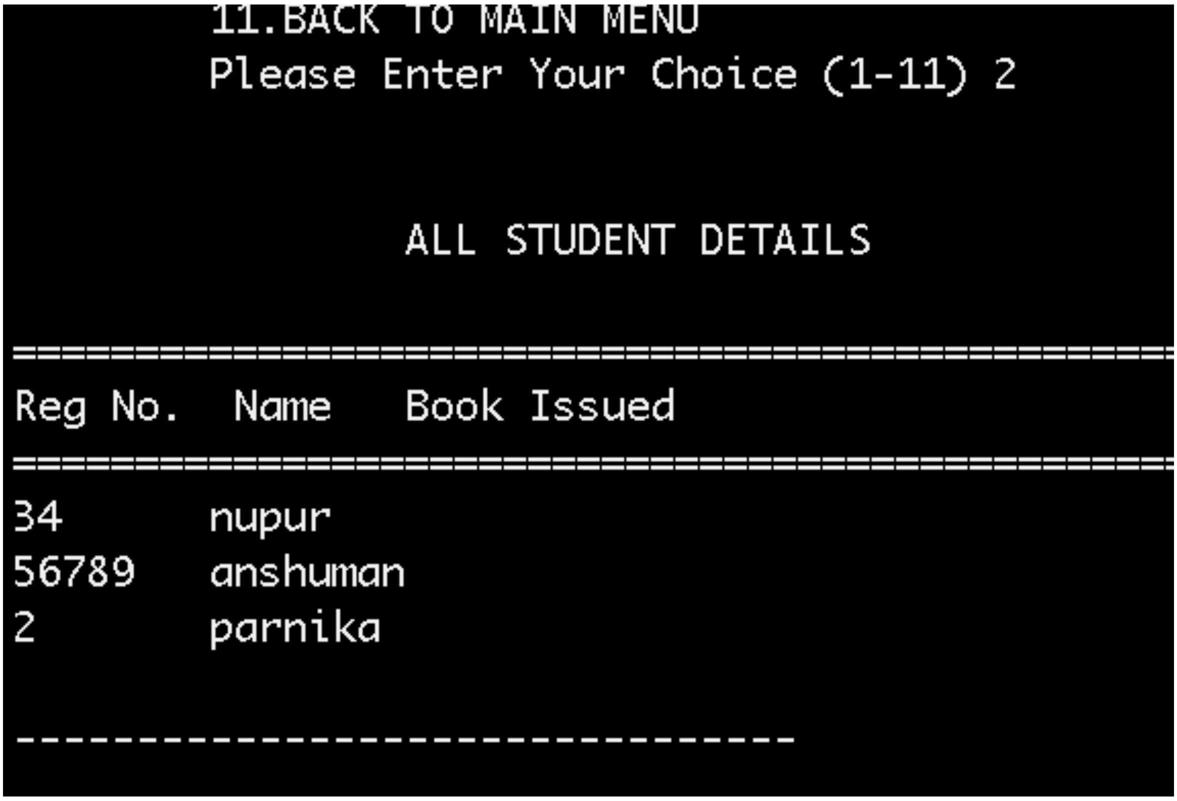
```
        11.BACK TO MAIN MENU
        Please Enter Your Choice (1-11) 3


        Please Enter The Registration Number 34

STUDENT DETAILS

Registration Number :   34
Student Name :  nupur
No of Book issued :     0
---------------------------------
```



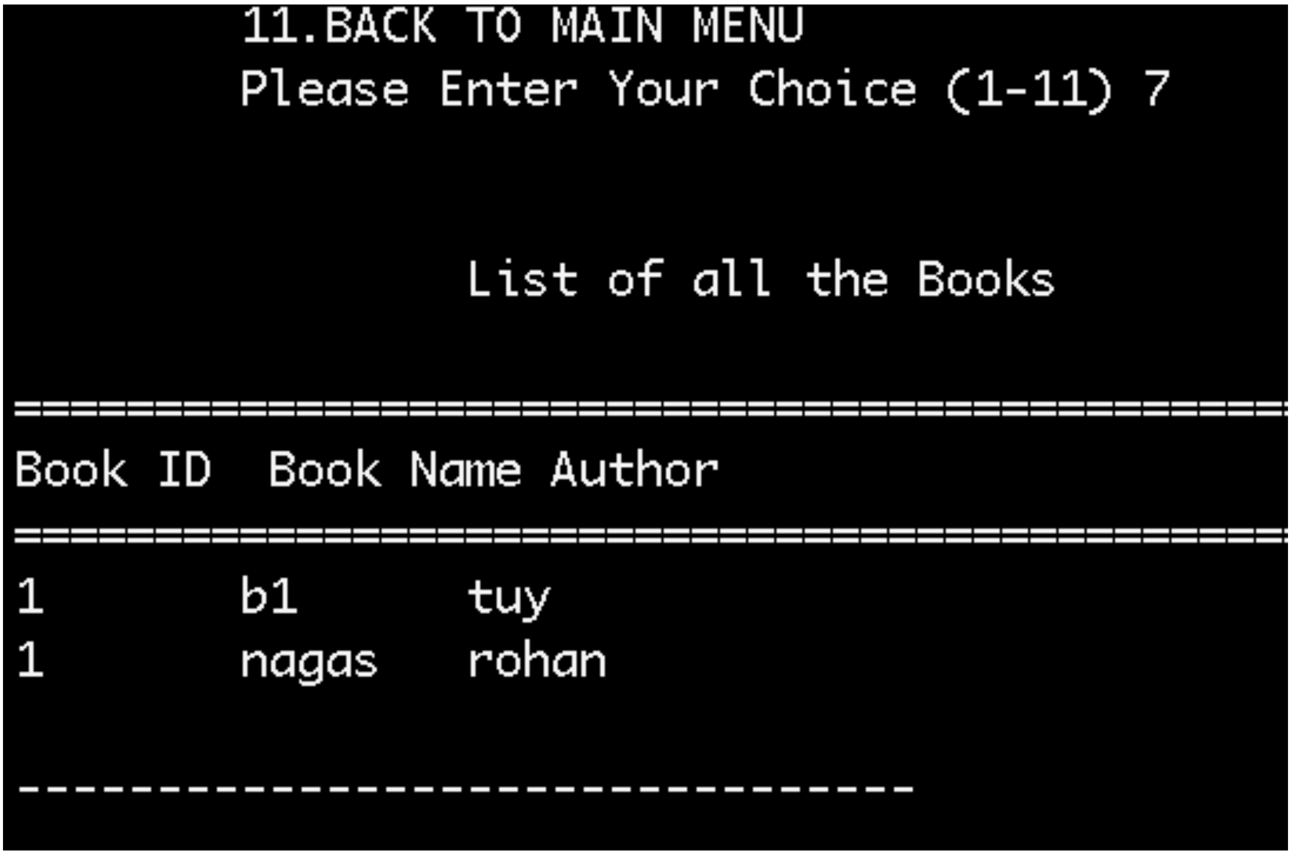
```
        Please Enter Your Choice (1-11) 6

Enter Book ID   1

Enter Name of The Book  nagas

Enter The Author's Name rohan

Enter Quantity of the Book      5


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

--------------------------------
```



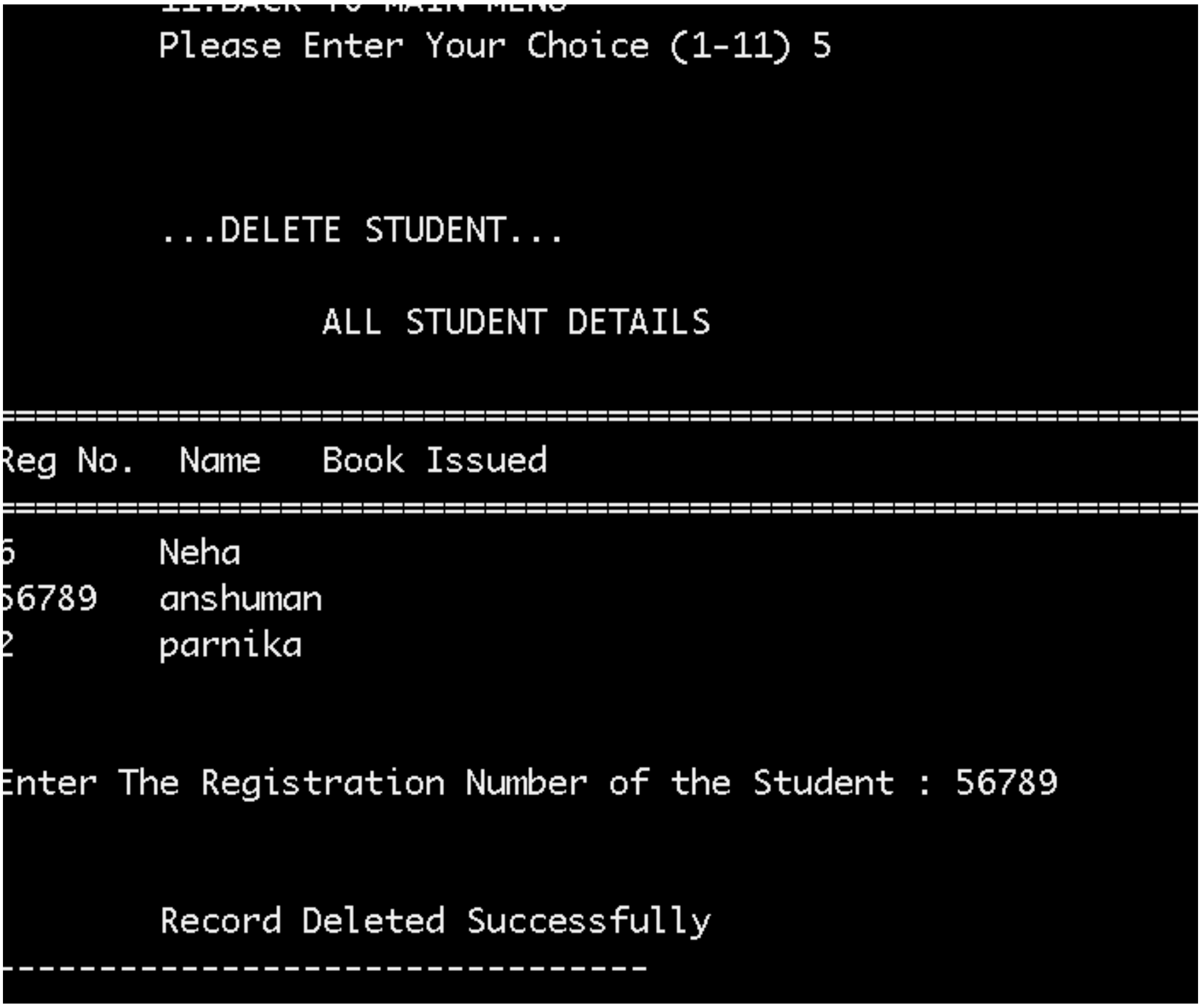
```
 ...Updating Database...



---------------------------------

                ADMIN DASHBOARD
---------------------------------
        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)
```



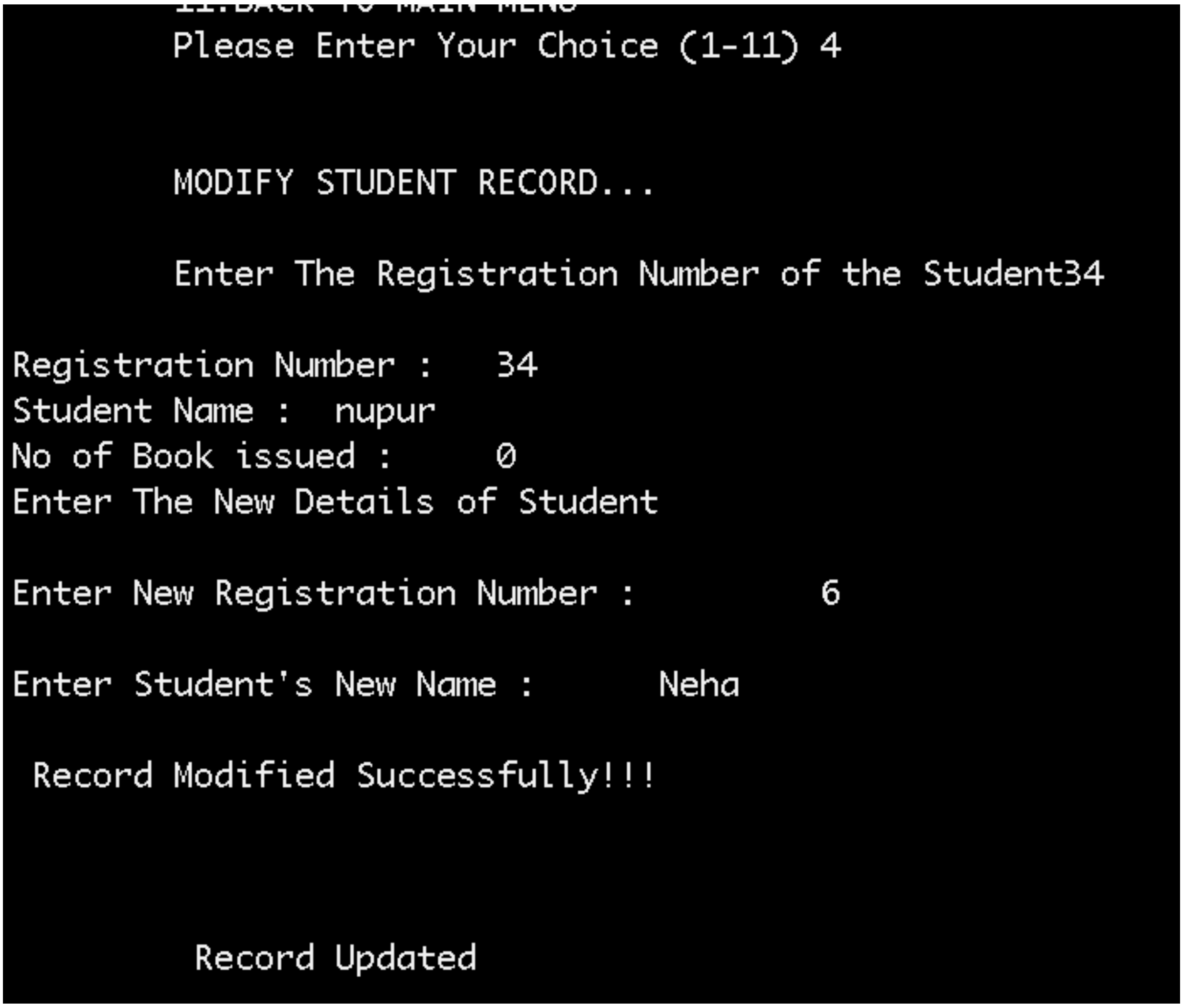
```
        11.BACK TO MAIN MENU
        Please Enter Your Choice (1-11) 2


                ALL STUDENT DETAILS

================================================
Reg No.  Name   Book Issued
================================================
34      nupur
56789   anshuman
2       parnika

--------------------------------
```



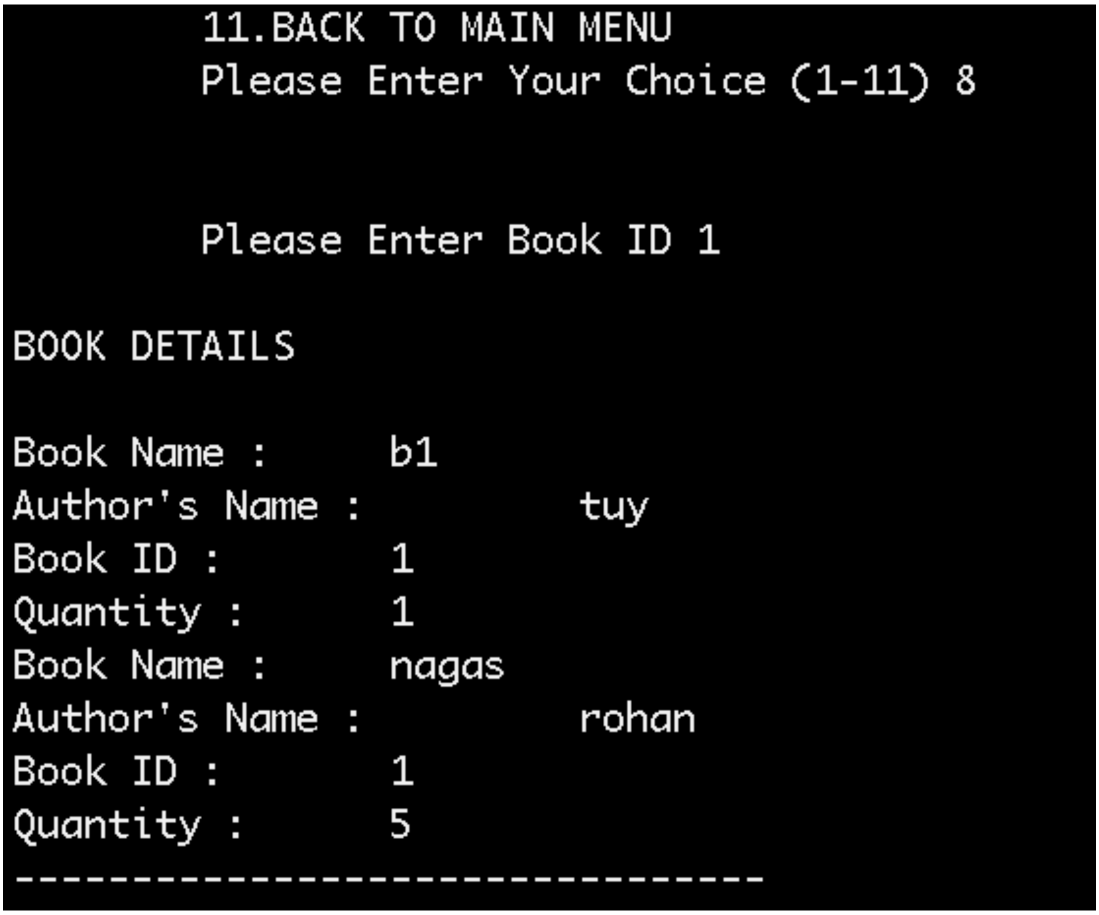
```
        11.BACK TO MAIN MENU
        Please Enter Your Choice (1-11) 7


                List of all the Books

=============================================
Book ID  Book Name Author
=============================================
1       b1      tuy
1       nagas   rohan

--------------------------------
```



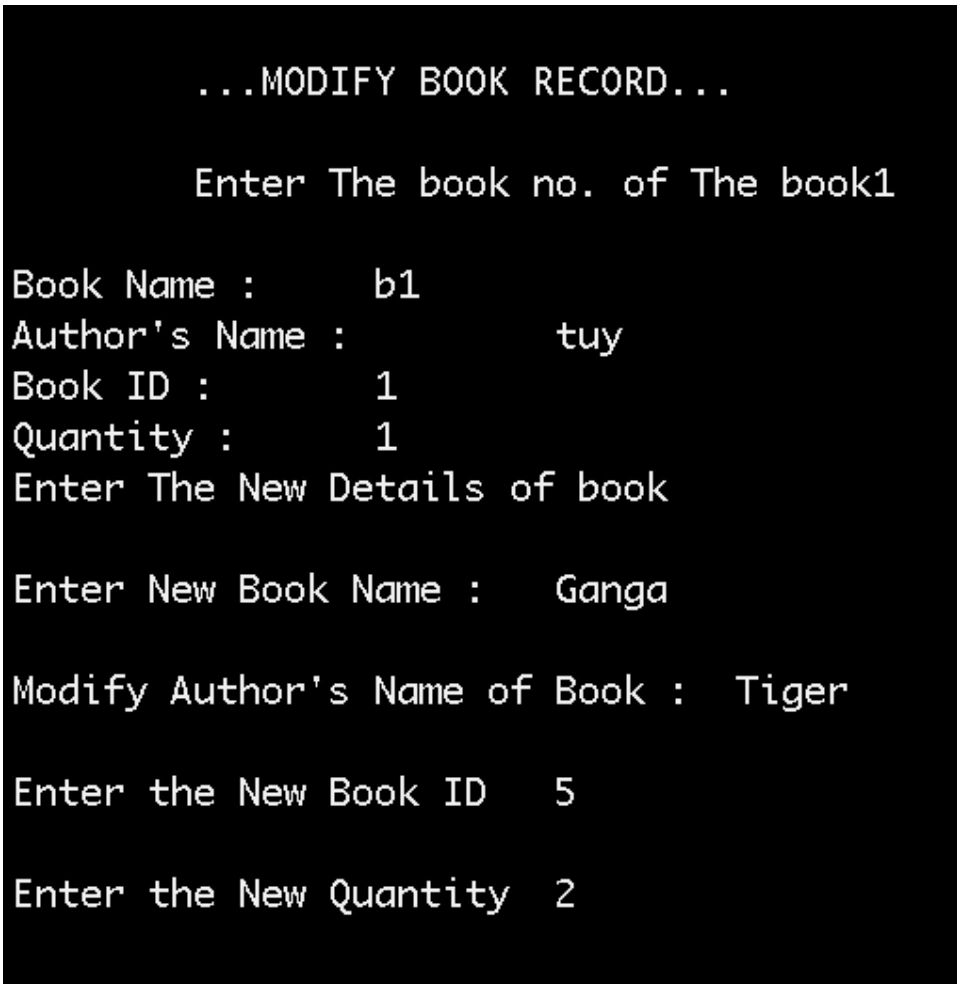
```
        11.BACK TO MAIN MENU
        Please Enter Your Choice (1-11) 5



        ...DELETE STUDENT...

                ALL STUDENT DETAILS

==============================================================
Reg No.  Name   Book Issued
==============================================================
6       Neha
56789   anshuman
2       parnika


Enter The Registration Number of the Student : 56789


        Record Deleted Successfully
---------------------------------
```



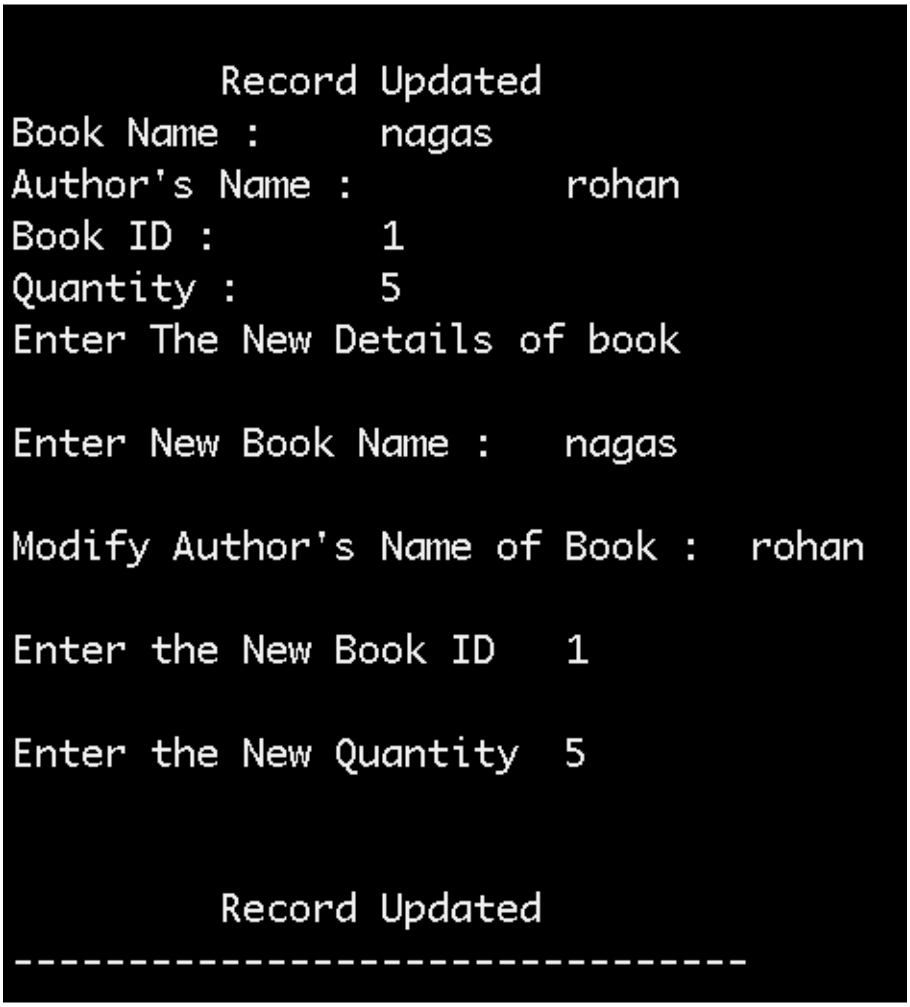
```
        11.BACK TO MAIN MENU
        Please Enter Your Choice (1-11) 4


        MODIFY STUDENT RECORD...

        Enter The Registration Number of the Student34

Registration Number :   34
Student Name :  nupur
No of Book issued :     0
Enter The New Details of Student

Enter New Registration Number :         6

Enter Student's New Name :      Neha

 Record Modified Successfully!!!



         Record Updated
```



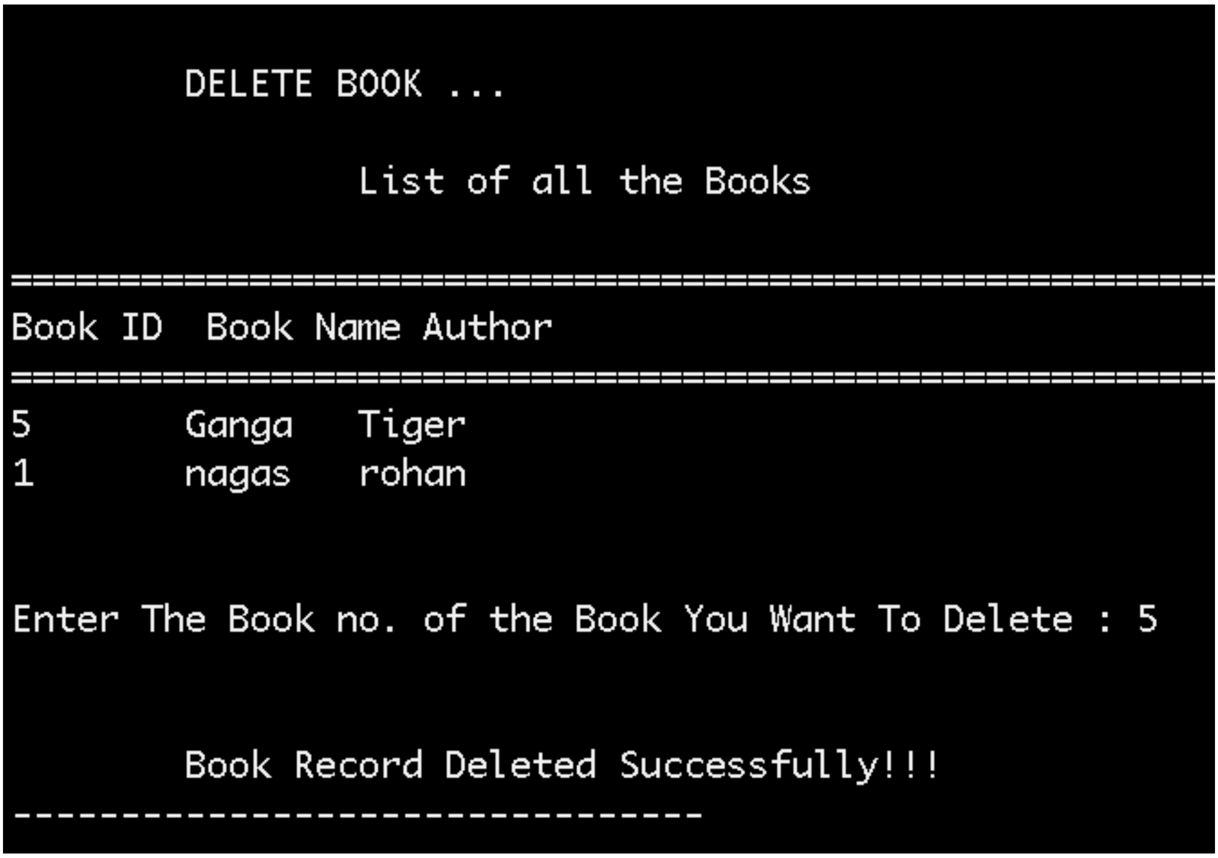
```
        11.BACK TO MAIN MENU
        Please Enter Your Choice (1-11) 8


        Please Enter Book ID 1

BOOK DETAILS

Book Name :     b1
Author's Name :         tuy
Book ID :       1
Quantity :      1
Book Name :     nagas
Author's Name :         rohan
Book ID :       1
Quantity :      5
--------------------------------
```



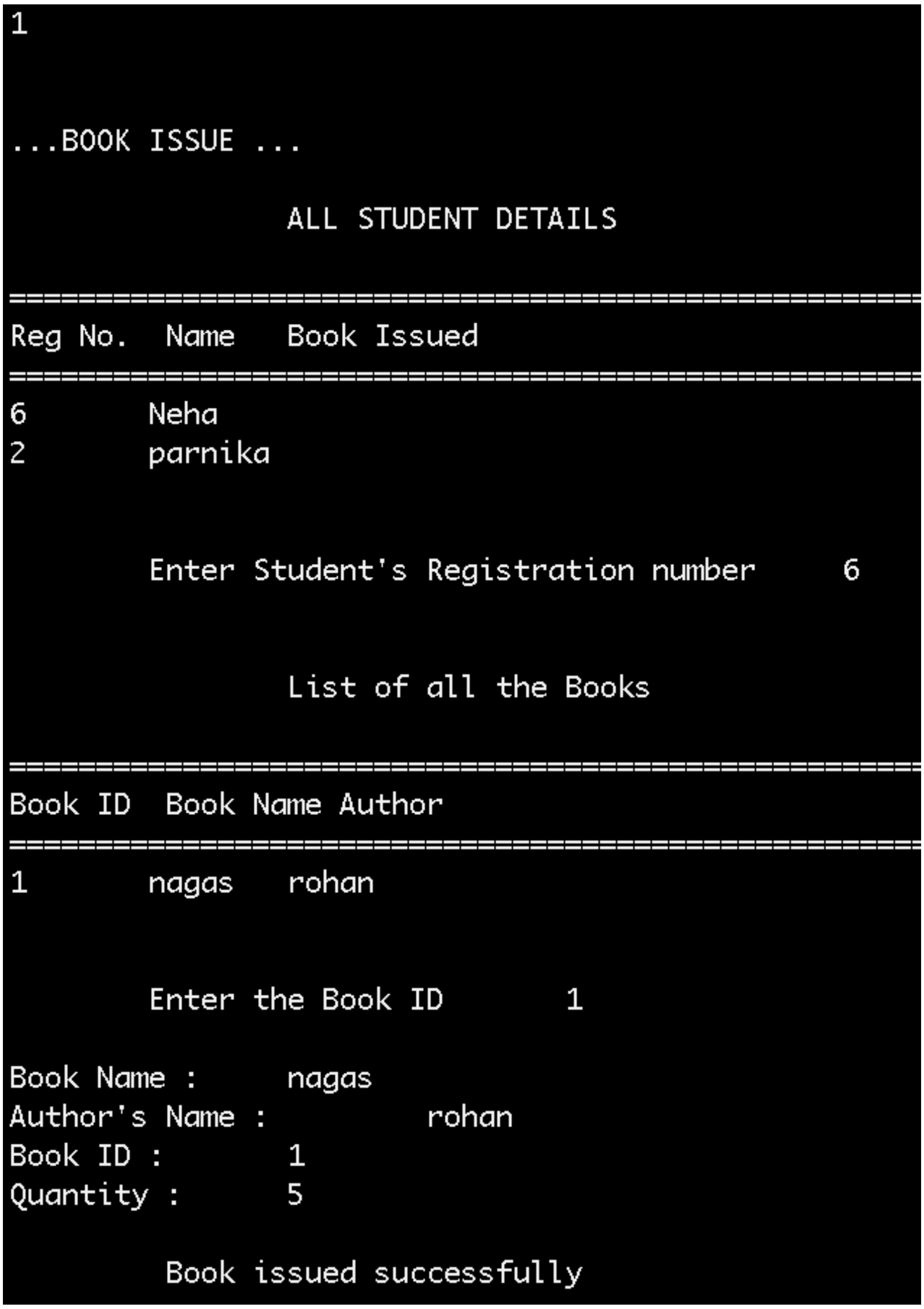
```
        ...MODIFY BOOK RECORD...

        Enter The book no. of The book1

Book Name :     b1
Author's Name :         tuy
Book ID :       1
Quantity :      1
Enter The New Details of book

Enter New Book Name :   Ganga

Modify Author's Name of Book :  Tiger

Enter the New Book ID   5

Enter the New Quantity  2
```



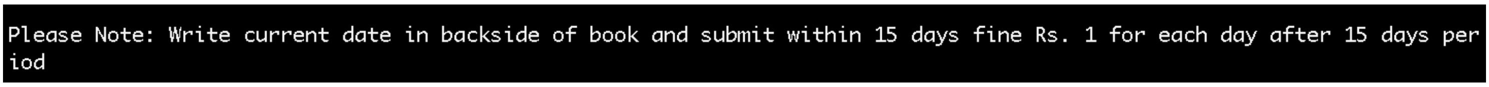
```
         Record Updated
Book Name :     nagas
Author's Name :         rohan
Book ID :       1
Quantity :      5
Enter The New Details of book

Enter New Book Name :   nagas

Modify Author's Name of Book :  rohan

Enter the New Book ID   1

Enter the New Quantity  5


         Record Updated
--------------------------------
```



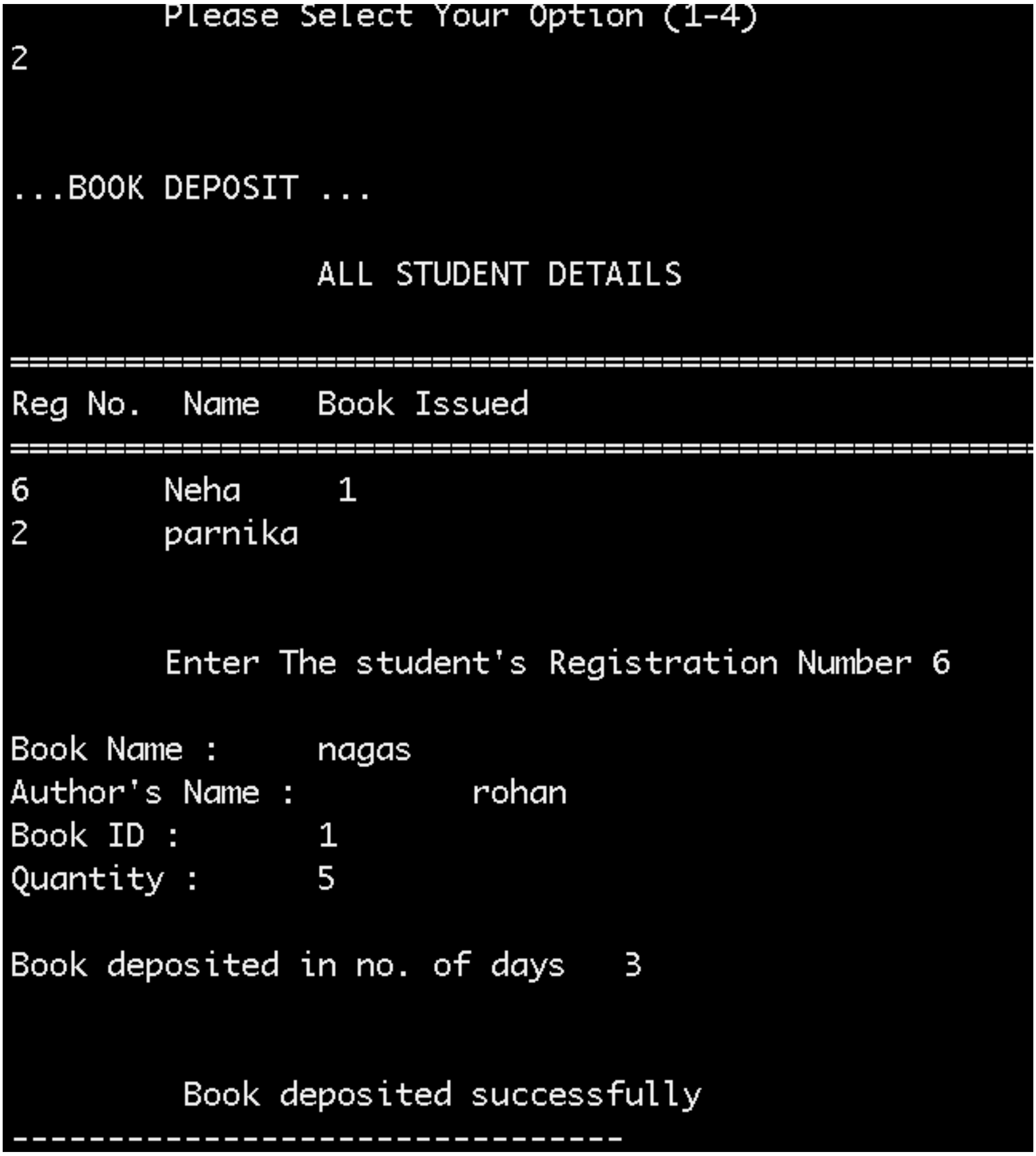
```
        DELETE BOOK ...

                List of all the Books


========================================================
Book ID  Book Name Author
========================================================
5       Ganga   Tiger
1       nagas   rohan


Enter The Book no. of the Book You Want To Delete : 5


        Book Record Deleted Successfully!!!
--------------------------------
```



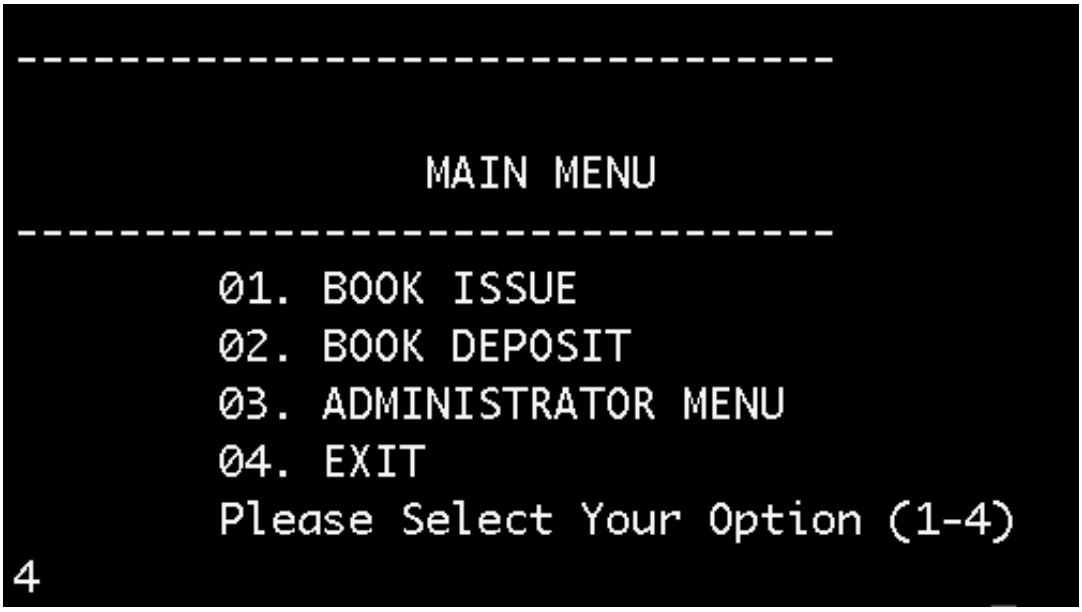
```
1


...BOOK ISSUE ...

                ALL STUDENT DETAILS

=====================================================
Reg No.  Name   Book Issued
=====================================================
6       Neha
2       parnika


        Enter Student's Registration number     6


                List of all the Books

=====================================================
Book ID  Book Name Author
=====================================================
1       nagas   rohan


        Enter the Book ID       1

Book Name :     nagas
Author's Name :         rohan
Book ID :       1
Quantity :      5

         Book issued successfully
```



```
Please Note: Write current date in backside of book and submit within 15 days fine Rs. 1 for each day after 15 days per
iod
```



```
        Please Select Your Option (1-4)
2


...BOOK DEPOSIT ...

                ALL STUDENT DETAILS

=====================================================
Reg No.  Name   Book Issued
=====================================================
6       Neha     1
2       parnika


        Enter The student's Registration Number 6

Book Name :      nagas
Author's Name :          rohan
Book ID :        1
Quantity :       5

Book deposited in no. of days   3


         Book deposited successfully
--------------------------------
```



```
--------------------------------


                MAIN MENU
--------------------------------
        01. BOOK ISSUE
        02. BOOK DEPOSIT
        03. ADMINISTRATOR MENU
        04. EXIT
        Please Select Your Option (1-4)
4
```

CONCLUSION

This Library Management System has been computed successfully and was also tested successfully. It is a user friendly and has required options, which can be utilised by the user to perform the desired operations.It is also portable for further enhancement.

BIBLOGRAPHY

- Computer Science with C++ by Sumita Arora Class XII
- www.geeksforgeeks.com
- www.google.com
- www.cplusplus.com