SDM COLLEGE OF ENGINEERING AND TECHNOLOGY, Dharwad-580002

(An autonomous Institution affiliated to
Visvesvaraya Technological University, Belagavi – 590018)

Department of Electronics and Communication Engineering

A Report on Project entitled

Implementation of Library Management System Using C++

Proposed by

Mr. Venkaraddi Raddi	USN:2SD20EC114
Mr. Nitishgouda Patil	USN:2SD20EC063

Students of 5th Semester ECE Department

Division: A

Under the guidance of

Prof. S. S. Navalgund

Department of ECE, SDMCET, Dharwad-02

Academic Year 2022-2023

INTRODUCTION:

As the name suggest the Library Management System is a software which handles the entire data of library. It makes the work of librarian very easy instead of writing data in a notebook. In past the librarians were using notebooks to write the data of books along with students name who borrowed that book. So it was very difficult to keep track on each and every book.
If librarian want's to search for a particular book then that task was very time consuming. So to make this task easy the programming languages were developed and C++ language is one of them.

PROBLEM STATEMENT:

In this Library Management System project, we have to design system that takes inputs from user and performs respective operations which results desirable output .Store details of books and students in database.

METHODOLOGY:

Step 1: At first we need to include suitable header files. These header files contains the set of predefined standard library functions.

Step 2: A class created for Student with data member std_usn, std_name, gender, phone_num and num_book_issued with member function Read_stud_data to read data from user. Disp_stud_data to display. These information are stored in file "Students_Details.dat". In which we can store, remove, update, search and display all data with file handling concept.

Step 3:A another class created for Books with data member book_id, book_name, book_authr, rack_no, stock; with the member function Read_book_data to read book data from user. Disp_book_data to display book details. These information are stored in file "Books_Details.dat". In which we can store, remove, update, search and display all data with file handling concept.

Step 4: main menu function written to display options to choose. Next another function to execute operation entered option.

Step 5: At last main function written where executions starts. It'll run until we enter 0.

Step 6: Run the code, with the input to produce the desired output.

SOURCE CODE:

```
#include <iostream>
#include <fstream>
#include <conio.h>
#include <string.h>
#include <windows.h>
using namespace std;
// ***********************student class****************************************
class Student
{
protected:
    char std_usn[12];
    char std_name[30];
    char gender;
    unsigned long long int phone_num;
    unsigned int num_book_issued;

public:
    Student();
    void Read_stud_data();
    void Disp_stud_data();
    void Disp_stud_in_row();

    void Store_Student_Details();
    int Remove_Student_Details(char *usn, char *name);
    void Veiw_All_Student_Details();
    int Seacher_Student(char *usn, int x);
    int Update_Student_Details(char *usn, char *name, int x);
};

Student::Student()
{
    strcpy(std_usn, "");
    strcpy(std_name, "STUDENT_NAME");
    gender = '-';
    phone_num = 0;
    num_book_issued = 0;
}

void Student::Read_stud_data()
{
    system("cls");
    cout << "\n\n\t\tENTER STUDENT DETAILS:";
    cout << "\n\t\tUSN      :";
    getchar();
    cin.getline(std_usn, 11);
    cout << "\t\tNAME     :";
    cin.getline(std_name, 29);
    cout << "\t\tGENDER   :";
    cin >> gender;
    cout << "\t\tPHONE No.:";
    cin >> phone_num;
}

void Student::Disp_stud_data()
{
    system("cls");
    cout << "\n\n\t\tSTUDENT DETAILS:";
    cout << "\n\tSTUDENT USN      :" << std_usn;
    cout << "\n\tSTUDENT NAME     :" << std_name;
    cout << "\n\tSTUDENT GENDER   :" << gender;
    cout << "\n\tSTUDENT PHONE No.:" << phone_num;
}

void Student::Disp_stud_in_row()
{
    cout << "\n\tSTUDENT ID:" << std_usn << "\tNAME :" << std_name << "\tNUMBER OF
BOOK ISSUED :" << num_book_issued;
}

void Student::Store_Student_Details()
{
    if (!(strcmp(std_usn, "")))
    {
        cout << "\n\n\tstudent is data not stored";
    }
    else
    {
        ofstream fout;
        fout.open("Students_Details.dat", ios::app | ios::binary);
        fout.write((char *)this, sizeof(*this));
        fout.close();
    }
}

int Student::Remove_Student_Details(char *usn, char *name)
{
    int x = 0, y = 0;
    ifstream fin;
    ofstream fout;
    fin.open("Students_Details.dat", ios::in | ios::binary);
    if (!fin)
        cout << "\n file is not found ";
    else
    {
        fout.open("TempStudents_Details.dat", ios::out | ios::binary);
        fin.read((char *)this, sizeof(*this));
        while (!fin.eof())
        {
            x++;
            if ((strcmp(usn, std_usn)) && (strcmp(name, std_name)))
            {
                fout.write((char *)this, sizeof(*this));
                y++;
            }
            fin.read((char *)this, sizeof(*this));
        }
        fin.close();
        fout.close();
        remove("Students_Details.dat");
        rename("TempStudents_Details.dat", "Students_Details.dat");
    }
    return (x - y);
}

void Student::Veiw_All_Student_Details()
{
    ifstream fin;
    fin.open("Students_Details.dat", ios::in | ios::binary);
    if (!fin)
    {
        cout << "\nfile is not opened ";
    }
    else
    {
        fin.read((char *)this, sizeof(*this));
        while (!fin.eof())
        {
            Disp_stud_in_row();
            fin.read((char *)this, sizeof(*this));
        }
        fin.close();
    }
}

int Student::Seacher_Student(char *usn, int x)
{
    int Cunt = 0;
    ifstream fin;
    fin.open("Students_Details.dat", ios::in | ios::binary);
    if (!fin)
        cout << "file is not found ";
    else
    {
        fin.read((char *)this, sizeof(*this));
        while (!fin.eof())
        {
            if (!strcmp(usn, std_usn))
            {
                if (x == 1)
                {
                    Disp_stud_in_row();
                    Cunt++;
                }
                else if (x == 2)
                {
                    Disp_stud_data();
                    Cunt++;
                }
            }
            fin.read((char *)this, sizeof(*this));
        }
        if (Cunt == 0)
            cout << "\n RECORD NOT FOUND IN THIS USN " << *usn << endl;
        fin.close();
    }
    return 0;
}

int Student::Update_Student_Details(char *usn, char *name, int x)
{
    int a = 0, b = 0;
    fstream file;
    file.open("Books_Details.dat", ios::in | ios::out | ios::ate | ios::binary);
    file.seekg(0);
    file.read((char *)this, sizeof(*this));
    while (!file.eof())
    {
        if (x == 1)
        {
            a++;
            if ((!strcmp(usn, std_usn)) && (!strcmp(name, std_name)))
            {
                b++;

                Read_stud_data();
                file.seekp(-sizeof(*this), ios::cur);
                file.write((char *)this, sizeof(*this));
            }
        }
        file.read((char *)this, sizeof(*this));
    }
    file.close();
    return (b);
}

// *****************book class****************************************

class Books
{
protected:
    unsigned int book_id;
    char book_name[30];
    char book_authr[30];
    unsigned int rack_no;
    unsigned int stock;

public:
    Books();
    void Read_book_data();
    void Disp_book_data();
    void Disp_book_in_row();
    void Disp_book_name();

    void Store_Book_Details();
    void Veiw_All_Book_Details();
    void Seacher_Book(char *ttl, unsigned int ID, int x);
    int Remove_Book(char *ttl, unsigned int ID);
    int Update_Book_Details(char *ttl, unsigned int ID, int x);
};

Books::Books()
{
    book_id = 0;
    strcpy(book_name, "NAME_BOOK");
    strcpy(book_authr, "NAME_AUTHOR");
    rack_no = 0;
    stock = 0;
}

void Books::Read_book_data()
{
    system("cls");
    cout << "\n\n\t\tENTER BOOK DETAILS:";
    cout << "\n\t\tID          :";
    cin >> book_id;
    cout << "\t\tNAME        :";
    getchar();
    cin.getline(book_name, 29);
    cout << "\t\tNAME_AUTHOR:";
    // getchar();
    cin.getline(book_authr, 29);
    cout << "\t\tRACK NUMBER :";
    cin >> rack_no;
    cout << "\t\tSTOCK       :";
    cin >> stock;
}

void Books::Disp_book_data()
{
    system("cls");
    cout << "\n\n\t\tBOOK DETAILS:";
    cout << "\n\tBOOK ID          :" << book_id;
    cout << "\n\tBOOK NAME        :" << book_name;
    cout << "\n\tBOOK NAME AUTHOR:" << book_authr;
    cout << "\n\tBOOK RACK NUMBER :" << rack_no;
    cout << "\n\tBOOK STOCK       :" << stock;
}
void Books::Disp_book_in_row()
{

    cout << "\n\tBOOK ID :" << book_id << "\tNAME  :" << book_name << "\tAUTHOR " <<
book_authr;
}
void Books::Disp_book_name()
{
    cout << book_name;
}

void Books::Store_Book_Details()
{
    if (book_id == 0)
    {
        cout << "\n\n\tbook is data not stored";
    }
    else
    {
        ofstream fout;
        fout.open("Books_Details.dat", ios::app | ios::binary);
        fout.write((char *)this, sizeof(*this));
        fout.close();
    }
}
void Books::Veiw_All_Book_Details()
{
    ifstream fin;
    fin.open("Books_Details.dat", ios::in | ios::binary);
    if (!fin)
    {
        cout << "\nfile is not opened ";
    }
    else
    {
        fin.read((char *)this, sizeof(*this));
        while (!fin.eof())
        {
            Disp_book_in_row();
            fin.read((char *)this, sizeof(*this));
        }
        fin.close();
    }
}

void Books::Seacher_Book(char *ttl, unsigned int ID, int x)
{
    int C = 0;
    ifstream fin;
    fin.open("Books_Details.dat", ios::in | ios::binary);
    if (!fin)
        cout << "file is not found ";
    else
    {
        fin.read((char *)this, sizeof(*this));
        while (!fin.eof())
        {
            if ((!strcmp(ttl, book_name)) || (ID == book_id))
            {
                if (x == 1)
                {
                    Disp_book_data();
                    C++;
                }
                else if (x == 2)
                {
                    Disp_book_name();
                }
            }
            fin.read((char *)this, sizeof(*this));
        }
        if ((x == 1) && (C == 0))
            cout << "\n Record not found in the name " << *ttl << endl;
        fin.close();
    }
}
int Books::Remove_Book(char *ttl, unsigned int ID)
```

```
{
    ifstream fin;
    ofstream fout;
    int x = 0, y = 0;
    fin.open("Books_Details.dat", ios::in | ios::binary);
    if (!fin)
        cout << "\n file is not found ";
    else
    {
        fout.open("TempBooks_Details.dat", ios::out | ios::binary);
        fin.read((char *)this, sizeof(*this));
        while (!fin.eof())
        {
            x++;
            if ((strcmp(book_name, ttl)) && (ID != book_id))
            {
                fout.write((char *)this, sizeof(*this));
                y++;
            }
            fin.read((char *)this, sizeof(*this));
        }
        fin.close();
        fout.close();
        remove("Books_Details.dat");
        rename("TempBooks_Details.dat", "Books_Details.dat");
    }
    return (x - y);
}
int Books::Update_Book_Details(char *ttl, unsigned int ID, int x)
{
    int a = 0, b = 0;
    fstream file;
    file.open("Books_Details.dat", ios::in | ios::out | ios::ate | ios::binary);
    file.seekg(0);
    file.read((char *)this, sizeof(*this));
    while (!file.eof())
    {
        if (x == 1)
        {
            a++;
            if (!(strcmp(ttl, book_name)) && (ID == book_id))
            {
                b++;
                Read_book_data();
                file.seekp(-sizeof(*this), ios::cur);
                file.write((char *)this, sizeof(*this));
            }
        }
```

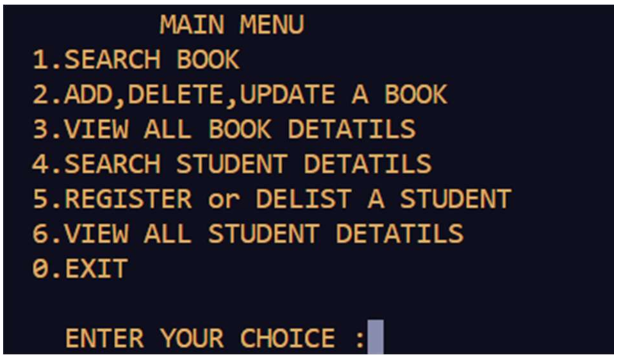
```
            file.read((char *)this, sizeof(*this));
        }

        file.close();
        return (b);
}

//**************************main manu*******************************************
void mainmenu()
{
    system("cls");
    system("Color 06");

    cout << "\n\t\tMAIN MANU\t\t\t";
    cout << "\n\t1.SEARCH BOOK";
    cout << "\n\t2.ADD,DELETE,UPDATE A BOOK";
    cout << "\n\t3.VIEW ALL BOOK DETATILS";
    cout << "\n\t4.SEARCH STUDENT DETATILS";
    cout << "\n\t5.REGISTER or DELIST A STUDENT"; // delist
    cout << "\n\t6.VIEW ALL STUDENT DETATILS";
    cout << "\n\t0.EXIT";
}

bool choice_op(int ip_ch)
{
    bool op_ch; // used to return back
    switch (ip_ch)
    {
    case (1):
    {
        system("Color 09");

        int chs = 0;
        unsigned int ID = 0;
        Books b;
        char Bk_Name[30];
        strcpy(Bk_Name, "");
        system("cls");
        cout << "\n\n\t\tSEARCH A BOOK BY:";
        cout << "\n\t\t1.BOOK NAME";
        cout << "\n\t\t2.BOOK ID";
        cout << "\n\n\t\tENTER YOUR CHOICE :";
        cin >> chs;
        if (chs == 1)
        {
            cout << "\n\n\t\tENTER NAME OF BOOK ";
            getchar();
            cin.getline(Bk_Name, 29);
```

```
                b.Seacher_Book(Bk_Name, ID, 1);
            }
            else if (chs == 2)
            {
                cout << "\n\n\t\tENTER ID OF BOOK ";
                cin >> ID;
                b.Seacher_Book(Bk_Name, ID, 1);
            }
            else
                cout << "\n\t\tERROR:WRONG INPUT";

            chs = 5;
            while (!(chs == 0))
            {
                cout << "\n\n\t\tENTER 0.EXIT : ";
                cin >> chs;
                if (chs == 0)
                    return true;
                else
                    cout << "\n\n\t\tERROR:WRONG INPUT ";
            }
            break;
        }
        case (2):
        {
            int chs = 0;
            Books b;
            system("cls");
            system("Color 04");
            cout << "\n\n\t\tADD,DELETE,UPDATE A BOOK:";
            cout << "\n\n\t\tPRESS";
            cout << "\n\t\t1.ADD BOOK";
            cout << "\n\t\t2.DELETE BOOK ";
            cout << "\n\t\t3.UPDATE BOOK ";
            cout << "\n\n\t\tENTER YOUR CHOICE :";
            cin >> chs;
            switch (chs)
            {
            case (1):
            {
                chs = 0;
                b.Read_book_data();
                b.Disp_book_data();
                cout << "\n\n\t\tENTER 1 TO SAVE DETAILS OR 0 SKIP : ";
                cin >> chs;
                if (chs == 1)
                {
                    b.Store_Book_Details();
                    cout << "\n\tBOOK DETAILS ARE SAVED SUCCESSFULLY";
                }
                else
                    cout << "\n\n\t\tERROR:WRONG INPUT ";
            }
            break;
            case (2):
            {
                unsigned int ID;
                char Bk_Name[30];
                system("cls");
                cout << "\n\n\t\tDELETE A BOOK :";
                cout << "\n\n\tENTER NAME OF BOOK ";
                getchar();
                cin.getline(Bk_Name, 29);
                cout << "\n\tENTER ID OF BOOK ";
                cin >> ID;
                cout << "\n\n\t\tPRESS 1 CONFIRM OR 0 SKIP : ";
                cin >> chs;
                if (chs == 1)
                {
                    if (b.Remove_Book(Bk_Name, ID))
                        cout << "\n\tBOOK RECORD IS DELETED SUCCESSFULLY";
                    else
                        cout << "\n\tERROR:WRONG INPUT NO BOOK RECORD IS FOUND";
                }
                else
                    cout << "\n\n\t\tERROR:WRONG INPUT ";
            }
            break;
            case (3):
            {
                unsigned int ID;
                char Bk_Name[30];
                system("cls");

                cout << "\n\n\t\tUPDATE A BOOK :";
                cout << "\n\n\tENTER NAME OF BOOK ";
                getchar();
                cin.getline(Bk_Name, 29);
                cout << "\n\n\tENTER ID OF BOOK ";
                cin >> ID;
                cout << "\n\n\t\tPRESS 1 CONFIRM OR 0 SKIP : ";
                cin >> chs;
                if (chs == 1)
                {
                    if (b.Update_Book_Details(Bk_Name, ID, 1))
                        cout << "\n\tBOOK RECORD IS UPDATEDED SUCCESSFULLY";
                    else
                        cout << "\n\tERROR:WRONG INPUT NO BOOK RECORD IS FOUND";
                }
                else
                    cout << "\n\n\t\tERROR:WRONG INPUT ";
            }
            break;
            default:
                cout << "\n\t\tERROR:WRONG INPUT";
                break;
            }

            chs = 5;
            while (!(chs == 0))
            {
                cout << "\n\n\t\tENTER 0.EXIT : ";
                cin >> chs;
                if (chs == 0)
                    return true;
                else
                    cout << "\n\n\t\tERROR:WRONG INPUT ";
            }
            break;
        }
        case (3):
        {
            system("cls");
            system("Color 05");
            Books b;
            int chs = 5;
            cout << "\n\t\tALL BOOKS DETAILS";
            cout << "\n\tBOOK ID \t\tBOOK NAME  \t\tAUTHOR";
            b.Veiw_All_Book_Details();
            while (!(chs == 0))
            {
                cout << "\n\n\t\tENTER 0.EXIT : ";
                cin >> chs;
                if (chs == 0)
                    return true;
                else
                    cout << "\n\n\t\tERROR:WRONG INPUT ";
            }
            break;
        }
        case (4):
        {
            system("cls");
            system("Color 06");
```

```
            char usn[12], name[30];
            Student s;

            cout << "\n\n\t\tENTER STUDENT DETAILS TO SEARCH:";
            cout << "\n\t\tUSN      :";
            getchar();
            cin.getline(usn, 11);
            s.Seacher_Student(usn, 2);
            int chs = 5;
            while (!(chs == 0))
            {
                cout << "\n\n\t\tENTER 0.EXIT : ";
                cin >> chs;
                if (chs == 0)
                    return true;
                else
                    cout << "\n\n\t\tERROR:WRONG INPUT ";
            }
            break;
        }
        case (5):
        {
            Student s;
            int chs = 0;
            system("cls");
            system("Color 07");
            cout << "\n\n\t\t  SEARCH A BOOK BY:";
            cout << "\n\t\t1.REGISTER STUDENT";
            cout << "\n\t\t2.DEREGISTER STUDENT";
            cout << "\n\n\t\t  ENTER YOUR CHOICE :";
            cin >> chs;
            if (chs == 1)
            {
                s.Read_stud_data();
                s.Disp_stud_data();
                cout << "\n\n\t\t  PRESS 1 CONFIRM OR 0 SKIP : ";
                cin >> chs;
                if (chs == 1)
                {
                    s.Store_Student_Details();
                    cout << "\n\n\tSTUDENT DETAILS SAVED SUCCESSFULLY";
                }
            }
            else if (chs == 2)
            {
                char usn[12], name[30];
                cout << "\n\n\t\tENTER STUDENT DETAILS TO DERESGISTER:";
                cout << "\n\t\tUSN      :";
                getchar();
                cin.getline(usn, 11);
                cout << "\t\tNAME     :";
                cin.getline(name, 29);
                cout << "\n\n\t\tPRESS 1 CONFIRM OR 0 SKIP : ";
                cin >> chs;
                if (chs == 1)
                {
                    if (s.Remove_Student_Details(usn, name))
                        cout << "\n\n\tSTUDENT RECORD IS DELETED SUCCESSFULLY";
                    else
                        cout << "\n\n\tERROR:WRONG INPUT,STUDENT RECORD IS NOT FOUND";
                }
            }
            else
                cout << "\n\t\tERROR:WRONG INPUT";
            chs = 5;
            while (!(chs == 0))
            {
                cout << "\n\n\t\tENTER 0.EXIT : ";
                cin >> chs;
                if (chs == 0)
                    return true;
                else
                    cout << "\n\n\t\tERROR:WRONG INPUT ";
            }
            break;
        }
        case (6):
        {
            system("cls");
            system("Color 08");
            Student s;
            int chs = 5;
            cout << "\n\t\tALL STUDENT DETAILS";
            cout << "\n\t\tSTUDENT ID \t\tNAME\t\t  NUMBER OF BOOK ISSUED";
            s.Veiw_All_Student_Details();
            while (!(chs == 0))
            {
                cout << "\n\n\t\tENTER 0.EXIT : ";
                cin >> chs;
                if (chs == 0)
                    return true;
                else
                    cout << "\n\n\t\tERROR:WRONG INPUT ";
            }
            break;
        }
```

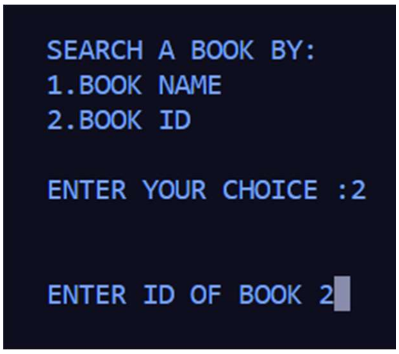
```
    case (0):
    {
        system("cls");
        system("Color 09");
        cout << "\n\n\n\n\t\tTHANK YOU\n\n\n\n";
        return false;
        break;
    }
    default:
    {
        cout << "\n\n\tERROR:ENTER CORRET CHOISE";
        Sleep(2000); // to make delay
        return true;
        break;
    }
    }
    return false;
}

int main()
{

    int ch;
    bool x = true;
    while (x == true)
    {
        mainmenu();
        cout << "\n\n\t  ENTER YOUR CHOICE :";
        cin >> ch;
        x = choice_op(ch);
        if (x == false)
        {
            exit;
        }
    }
}
```

RESULTS:

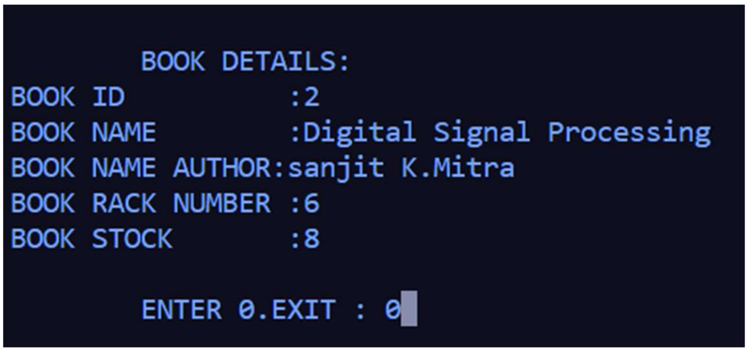


```
          MAIN MENU
1.SEARCH BOOK
2.ADD,DELETE,UPDATE A BOOK
3.VIEW ALL BOOK DETATILS
4.SEARCH STUDENT DETATILS
5.REGISTER or DELIST A STUDENT
6.VIEW ALL STUDENT DETATILS
0.EXIT

  ENTER YOUR CHOICE :
```

1. Search book

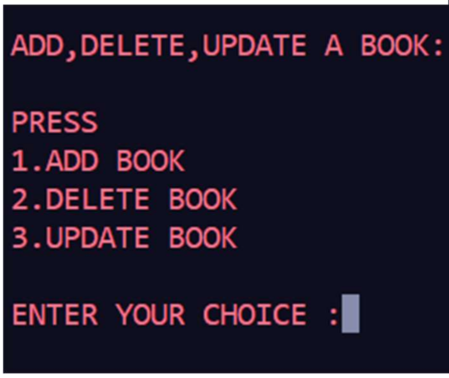


```
SEARCH A BOOK BY:
1.BOOK NAME
2.BOOK ID

ENTER YOUR CHOICE :2


ENTER ID OF BOOK 2
```

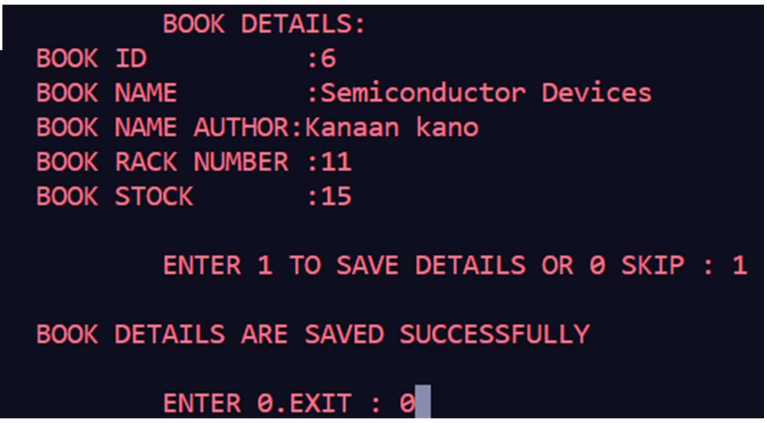


```
        BOOK DETAILS:
BOOK ID          :2
BOOK NAME        :Digital Signal Processing
BOOK NAME AUTHOR:sanjit K.Mitra
BOOK RACK NUMBER :6
BOOK STOCK       :8

        ENTER 0.EXIT : 0
```

2.1. Adding Book

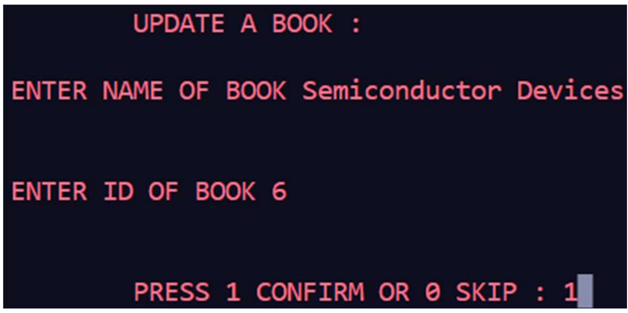


```
ADD,DELETE,UPDATE A BOOK:

PRESS
1.ADD BOOK
2.DELETE BOOK
3.UPDATE BOOK

ENTER YOUR CHOICE :
```

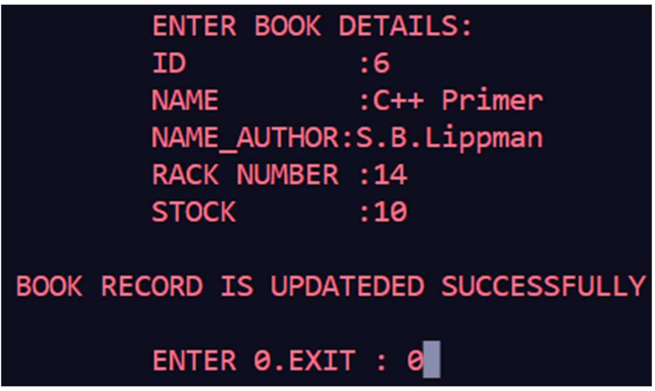


```
        BOOK DETAILS:
BOOK ID          :6
BOOK NAME        :Semiconductor Devices
BOOK NAME AUTHOR:Kanaan kano
BOOK RACK NUMBER :11
BOOK STOCK       :15

        ENTER 1 TO SAVE DETAILS OR 0 SKIP : 1

BOOK DETAILS ARE SAVED SUCCESSFULLY

        ENTER 0.EXIT : 0
```

2.2. Update Book

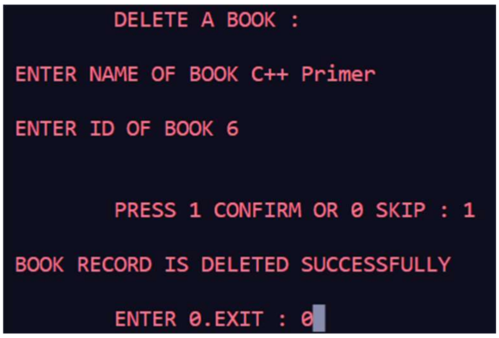


```
        UPDATE A BOOK :

ENTER NAME OF BOOK Semiconductor Devices


ENTER ID OF BOOK 6


        PRESS 1 CONFIRM OR 0 SKIP : 1
```



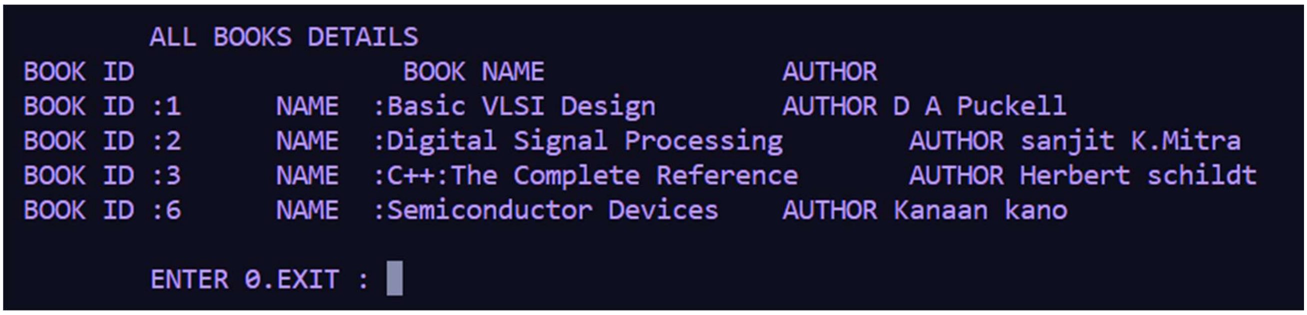
```
        ENTER BOOK DETAILS:
        ID           :6
        NAME         :C++ Primer
        NAME_AUTHOR:S.B.Lippman
        RACK NUMBER :14
        STOCK        :10

BOOK RECORD IS UPDATEDED SUCCESSFULLY

        ENTER 0.EXIT : 0
```

2.3. Delete Book



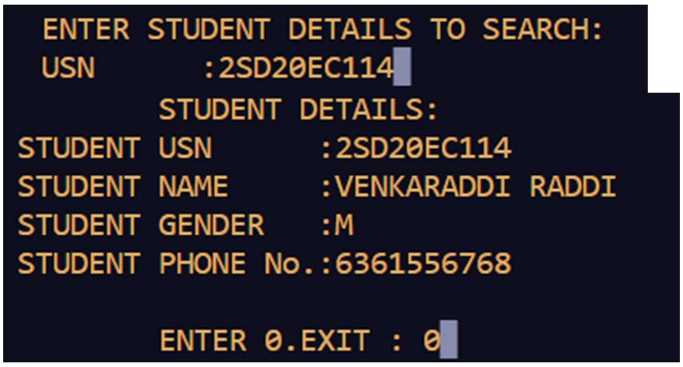
```
        DELETE A BOOK :

ENTER NAME OF BOOK C++ Primer

ENTER ID OF BOOK 6


        PRESS 1 CONFIRM OR 0 SKIP : 1

BOOK RECORD IS DELETED SUCCESSFULLY

        ENTER 0.EXIT : 0
```

3. View All Book Detail’s



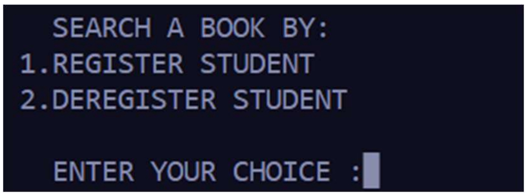
```
        ALL BOOKS DETAILS
BOOK ID                  BOOK NAME                AUTHOR
BOOK ID :1      NAME  :Basic VLSI Design          AUTHOR D A Puckell
BOOK ID :2      NAME  :Digital Signal Processing          AUTHOR sanjit K.Mitra
BOOK ID :3      NAME  :C++:The Complete Reference         AUTHOR Herbert schildt
BOOK ID :6      NAME  :Semiconductor Devices      AUTHOR Kanaan kano

        ENTER 0.EXIT :
```

4. Search Student Detail’s

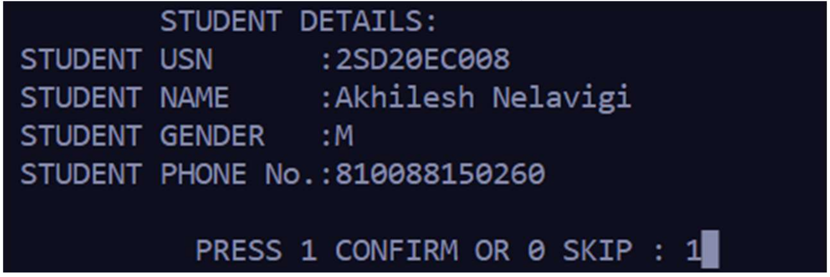


```
  ENTER STUDENT DETAILS TO SEARCH:
  USN       :2SD20EC114
         STUDENT DETAILS:
STUDENT USN        :2SD20EC114
STUDENT NAME       :VENKARADDI RADDI
STUDENT GENDER     :M
STUDENT PHONE No.:6361556768

         ENTER 0.EXIT : 0
```

5.1. Register Student

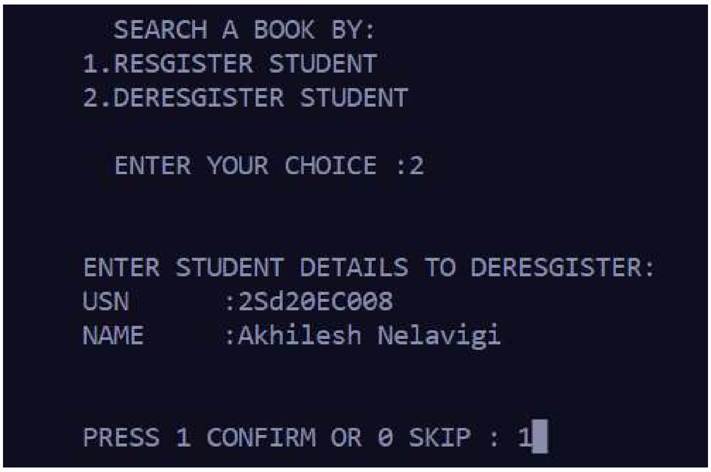


```
  SEARCH A BOOK BY:
1.REGISTER STUDENT
2.DEREGISTER STUDENT

  ENTER YOUR CHOICE :
```



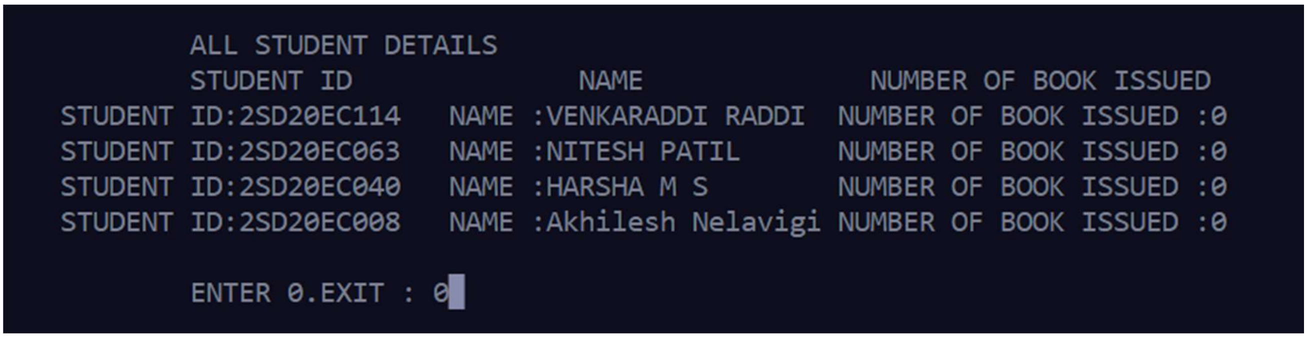
```
         STUDENT DETAILS:
STUDENT USN        :2SD20EC008
STUDENT NAME       :Akhilesh Nelavigi
STUDENT GENDER     :M
STUDENT PHONE No.:810088150260

            PRESS 1 CONFIRM OR 0 SKIP : 1
```

5.2. Deregister Student



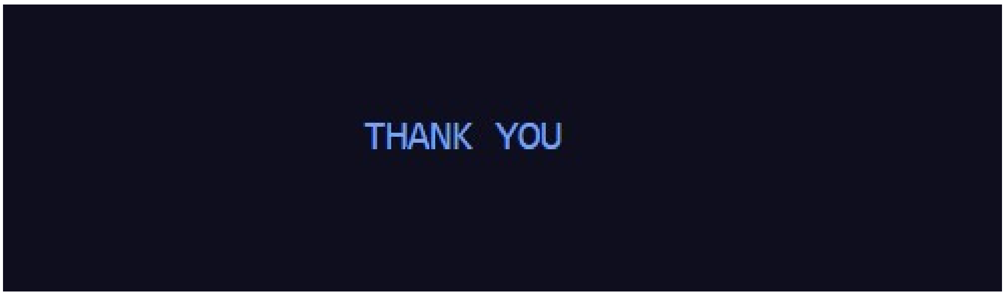
```
  SEARCH A BOOK BY:
1.RESGISTER STUDENT
2.DERESGISTER STUDENT

  ENTER YOUR CHOICE :2


ENTER STUDENT DETAILS TO DERESGISTER:
USN      :2Sd20EC008
NAME     :Akhilesh Nelavigi


PRESS 1 CONFIRM OR 0 SKIP : 1
```

6. View All Student Detail’s



```
        ALL STUDENT DETAILS
        STUDENT ID               NAME               NUMBER OF BOOK ISSUED
STUDENT ID:2SD20EC114   NAME :VENKARADDI RADDI   NUMBER OF BOOK ISSUED :0
STUDENT ID:2SD20EC063   NAME :NITESH PATIL       NUMBER OF BOOK ISSUED :0
STUDENT ID:2SD20EC040   NAME :HARSHA M S         NUMBER OF BOOK ISSUED :0
STUDENT ID:2SD20EC008   NAME :Akhilesh Nelavigi NUMBER OF BOOK ISSUED :0

        ENTER 0.EXIT : 0
```

0. Exit



```
                      THANK YOU
```