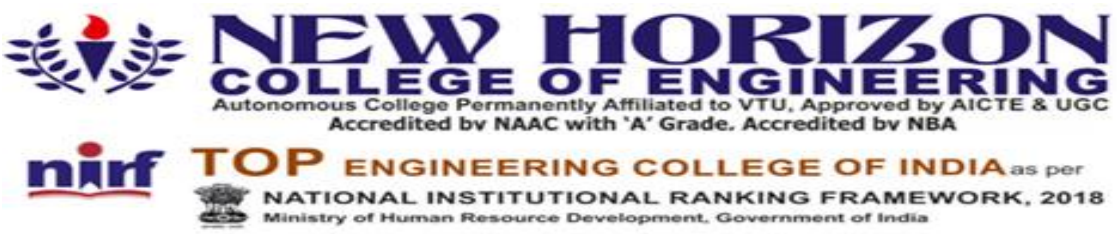

A

MINI PROJECT REPORT

ON

"LIBRARY MANAGEMENT SYSTEM USING C"

Submitted in the partial fulfillment of the requirements in the 3rd semester of

BACHELOR OF ENGINEERING

IN

INFORMATION SCIENCE AND ENGINEERING

BY

SOUJANYA S

(1NH17IS105)

Under the guidance of

Prof. Karthiyayini

Sr. Assistant Professor

Dept. of ISE, NHCE

DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING

NEW HORIZON COLLEGE OF ENGINEERING

(Autonomous College Permanently Affiliated to VTU, Approved by AICTE,
Accredited by NAAC with 'A' Grade & NBA)
Ring Road, Bellandur Post, Near Marathahalli,
Bengaluru-560103, INDIA

CERTIFICATE

Certified that the mini project entitled "Library Management System using C" carried out by Soujanya S(1NH17IS105), a bonafied student of New Horizon College of Engineering, Bengaluru, in partial fulfillment of the requirements in the III semester of Bachelor of Engineering in Information Science and Engineering of the Visvesvaraya Technological University, Belgaum during the year 2018-2019. The project report has been approved as it satisfies the academic requirement in respect of mini project work.

Project Guide **Head of the Department**

ABSTRACT

The project titled `Library Management System` is an application developed to monitor and control the activities performed in a library such as updating the books available in the library which includes adding new editions of the books and deleting the older ones, viewing and searching the book details, issuing the books and many more. Through the above proposed library management system it is possible to overcome the problems faced by the existing manual library management system. Managing a library manually can be a tedious task. There are higher probabilities of data loss in the case of manual library management system and performing all the tasks in a library without software involves a lot of manual labor. Therefore to make the process of managing a library easier and effective the above proposed library management system can be adapted.

ACKNOWLEDGEMENT

Any achievement, be it scholastic or otherwise does not depend solely on the individual efforts but on the guidance, encouragement and cooperation of the intellectuals, elders and friends. A number of personalities, in their own capacities have helped me in carrying out on this mini project. I would like to take this opportunity to thank them all.

I thank the management, **Dr. Mohan Manghnani**, chairman, New Horizon Educational Institutions for providing the necessary infrastructure and creating conducive environment for effective learning.

I also record here the constant encouragement, support and facilities extended to us by **Dr. Manjunatha**, Principal, New Horizon College of Engineering, Bengaluru.

I extend sincere gratitude for the constant encouragement, support and facilities provided to us by **Dr. R J Anandhi**, Professor and Head of the Department, Department of Information Science and Engineering, New Horizon College of Engineering, Bengaluru.

I sincerely acknowledge the encouragement, timely help and guidance to me by **Prof. Karthiyayini**, Sr. Assistant Professor, Department of Information Science and Engineering, New Horizon College of Engineering, Bengaluru, to complete the mini project within the stipulated time successfully.

Finally, a note of thanks to the teaching and non-teaching staff of Information Science and Engineering Department for their cooperation extended to us and our friends, who helped me directly or indirectly in the successful completion of this mini project.

TABLE OF CONTENTS

Chapter 01

INTRODUCTION

The mini project `Library Management System` is an application developed for managing a library. The project aims at developing an application using C Language that enables a organization or a university to maintain its library efficiently. The Library Management System uses basic C functions and concepts to generate menus, display messages, print texts on the screen and many more. The above proposed system implements the concept of structures using C to define the library items along with various other concepts of C such as files and its operations, looping and branching constructs and strings and string manipulation functions.

1.1 Motivation of project

Libraries play a very vital role in running an institution as it aids the students to acquire knowledge on any topic of their wish and provide them an overall development. Even in the digital era that we are living in, many libraries are managed manually. The existing manual library management system has its own set of disadvantages which includes data loss, requirement of manual labor, consumes more time and many more.

To overcome the disadvantages discussed above the application `Library Management System` is developed. The `Library Management System` allows the user to perform various different tasks such as entering the records of the new books and retrieving their details. Through the above proposed system, the user can list all the books available in the library, issue them, maintain the records of the issued books and many more tasks which makes the task of managing the library simpler and effective. The librarian can log into the system using the password, modify the database, issue books and store the details of the issued books using which the librarian can identify the books returned after the due date.

Users can search for books and can renewal the books online. In the above proposed system, we assume that every student will have an identity card which can be used to take books from library, return them and pay fines if necessary. They can recommend for new books by just sending messages to the librarian from anywhere in the college.

Whenever a book is issued to a student, the details about the issued book which can include the unique book id, book name, its author and the date the book was issued, the due date along with the information about the student including the student's name, USN and the year of study is saved to a file. When the student returns the book the details about the issued date is displayed and the student needs to pay fine if necessary.

1.2 Problem Statement

To design and implement different functions in a library management system to overcome the disadvantages of the existing manual library management system.

Chapter 02

SYSTEM REUIREMENT SPECIFICATION

Purpose: The main purpose of the library management system is that it reduces the manual work involved in managing a library. Through the above proposed system the library can be maintained efficiently.

The purposes of developing of a library management system are:

1. To implement different constructs of C language like lops, branching constructs, switch cases and many more.
2. To be able to solve different problem statements in C.
3. To be able to work in a group and hence develop team management skills.
4. To understand different concept of C.

The objectives of the library management system are:

1. To eliminate the errors involved in the existing manual library management system.
2. To simplify different processes in the library and make it efficient.
3. To reduce the paper work involved in the manual library management system.
4. To ensure that n data is lost by saving all the details in a file.

Scope:

1. Ensures effective management of library.
2. The above proposed system is password protected therefore only the person responsible can have access to the application.
3. Any book details can be viewed and searched easily.
4. The details about the books issued are recorded along with the date issued and due date which makes the process of issuing and returning books simpler.

3.1 Hardware System Configuration

Processor	-Intel Core i5
Speed	-1.8 GHz
RAM	-256 MB (min)
Hard disk	-10 GB

3.2 Software System Specifications

Operating System	-Windows 8
Programming Language	-C language
Compiler	-Code:: blocks

Chapter 03

METHODOLOGY

3.1 Algorithm

MAINMENU FUNCTION

The above proposed library management enables the user to perform various functions such as adding new books to the library, deleting the older ones, issuing the books, searching and viewing book details and returning the books to the library. As soon as the user opens the application, it demands for a password and only if the password is correct the user can continue.

After entering the correct password, there will be a main menu displayed on the screen which displays the different actions that can be performed in the application. Based on the user's choice the particular function will be called and the task can be performed. The algorithm for the above function is as follows-

STEPS:

1) Starts with the welcome screen which asks the user to enter the password.
2) Password is checked
   - If correct: mainmenu() function is called

     I) the mainmenu is displayed as below:

        1. ADD BOOKS
        2. DELETE BOOKS
        3. SEARCH BOOKS
        4. ISSUE BOOKS
        5. VIEW BOOKS
        6. RETURN BOOKS
        7. CLOSE THE APPLICATION

     II) Get the choice from the user

     III) If the user enters:

        Choice 1 then call add_books () function

        Choice 2 then call delete_books () function

        Choice 3 then call search_books () function

        Choice 4 then call issue_books () function

        Choice 5 then call view_books () function

Choice 6 then call return_books () function

Choice 7 then exit 0;

- If incorrect: displays a message “INVALID PASSWORD” and exits.

3) Stop

ADD BOOKS FUNCTION

In a library we will need to add new editions of the books regularly so as to keep the library updated. When the user enters the choice 1 the add_books () function is called. This function asks the user for the book details and the given book details will be saved in a file named aboutbook.txt. The algorithm for the above function is as follows:

STEPS

1) system (“cls”)

2) Declare the file pointer `f1`

3) Open the file named “aboutbook.txt” in append mode

4) while (ch==1)

- If the condition is true, then ask the user for the book details.
- Write the details into the file opened
- Ask the user if he wants to continue
  I. If the input is 1: repeat the step 4
  II. If the input is 0: call the mainmenu() function

5) Close the file

6) Stop

DELETE BOOKS FUNCTION

Updating a library even includes deletion of the older editions of the books. When the user enters choice 2 the delete books will be called. This function will ask the user to enter the book id to be deleted and will delete the book if it is present in the file.

STEPS

1) Declare the file pointers f1 and ft.
2) While(ch==1)
   - Ask the user to enter the book id
   - Open the file named aboutbook.txt
   - Read till end of the file and check if the book id given and the bookid present in the file are same.

I. If yes: display the message “book is found” and the display the details of the books.
II. ask the user if he is sure of deleting the book
    i. if yes: open the temporary file temp.txt.
    ii. copy all the book details from the aboutbook.txt to temp.txt except the details of the bookid to be deleted.
    iii. Remove aboutbook.txt and rename temp.txt to aboutbook.txt.
    iv. Close the file.
III. If no: display an error message and call the mainmenu function.

SEARCH BOOKS

This function is called if the user wants to a search a particular book detail. He can search using either the book id or the book name.

STEPS

1) Declare a file pointer f1.
2) While(e==1)
    - Ask the user if he wants search by book id or the book name.
    - If he wants to search by book id then:
        i. Ask the user to enter the book id.
        ii. Open the file aboutbook.txt.
        iii. Read till and of the file.
        iv. If the book id given and the book id in the file are same display the message “book found” and display the details.
    - If he wants to search using the book name then:
        i. Ask the user to enter the book id.
        ii. Open the file aboutbook.txt.
        iii. Read till and of the file.
        v. If the book id given and the book id in the file are same display the message “book found” and display the details.
3) Close the files.

ISSUE BOOKS FUNCTION

This function is used to issue books to the students and the user can view issued books and search them using this function.

STEPS

1) Declare the file pointers fb and f1.
2) Display the issue books menu to the user.
    i) ISSUE BOOKS
    ii) VIEW ISSUE BOOKS
    iii) SEARCH ISSUED BOOKS
    iv) MAINMENU
3) Ask the user to enter the choice and call the required functions.
4) If the user enters 1, issuebooks function is called .
    i) Ask the user to enter the book id.
    ii) Compare this id with the book id in the file.
    iii) If the book is present display the book information and ask for student details along with the issued date and the due date.
    iv) Write the above given details into the file issuedbook.txt.
    v) If the bookid is not present, then display an error message.
    vi) Close the files and call issue books functions once again.
5) If the user enters 2,
    i) Declare the file pointer fb and open the file in read mode.
    ii) Read till end of the file and display all the book details.
    iii) Close the file and call the issuedbooks function again.
6) If the user enters 3,
    i) Declare the file pointer fb.
    ii) Ask the user for book id.
    iii) Open the file issuedbook.txt and read till end of the file.
    iv) If the book id given and the book id in the file matches then display the book details.
    v) Else display an error message and close the file.
7) If the user enters 4, call the mainmenu function.

RETURN BOOKS

If the student wants to return the book back to library this function is called.

1) Declare the file pointers fb and ft.

2) While(ch==1)
    - Ask the user to enter the book id to be returned.
    - Open the file named issuedbook.txt
    - Read till end of the file and check if the book id given and the bookid present in the file are same.
        i. If yes: display the message “book is found” and the display the details of the books.
        ii. ask the user if he is sure of returning the book
            a) If yes: open the temporary file temp.txt.
            b) Copy all the book details from the issuedbook.txt to delete.txt except the details of the bookid to be deleted.
            c) Remove issuedbook.txt and rename delete.txt to issuedbook.txt.
            d) Close the file.
            e) If no: display an error message and call the mainmenu function.

VIEW BOOKS FUNCTIONS

This function is called when the user wants to view all the books in a library.

STEPS

1) Declare the file pointer f1 and open the file aboutbook.txt in read mode.
2) Read till end of the file.
3) Display all the book details.
4) Close the file.
5) Call the mainmenu function.

3.3 Code and implementation

```
#include<stdio.h>
#include<strings.h>
#include<stdlib.h>
#include<time.h>

void mainmenu();
void add_books();
void delete_books();
void search_books();
void issue_books();
void view_books();
void return_books();
struct date
 {
 int dd,mm,yy;
};
struct books
{
  int rack,quantity,year;
  char branch[50],bookid[20],name[50],author[50],stname[100],stusn[10];
  struct date issued;
  struct date due;
  };
struct books a;
int main()
{
char password[10];
char s[20]={"mini"};
printf("\n**********PASSWORD PROTECTED**********\n");
printf("\nENTER PASSWORD: ");
scanf("%s",password);
if(strcmp(s,password)==0)
{
```

```
printf("\nWELCOME!\nLOGIN SUCCESSFUL\n");
   system("cls");
   mainmenu();
   return;
}
else
   {
   printf("\nINVALID PASSWORD\n");
   return;
}
}
void mainmenu()
{
   int ch;
printf("\n1.ADD BOOKS\n2.DELETE BOOKS\n3.SEARCH BOOKS\n4.ISSUE BOOKS\n5.VIEW BOOKS\n6.RETURN BOOKS\n7.CLOSE THE APPLICATION\n");
printf("Enter choice\n");
scanf("%d",&ch);
switch(ch)
{
case 1:
   add_books();
   break;
case 2:
   delete_books();
   break;
case 3:
   search_books();
   break;
case 4:
   issue_books();
    break;
case 5:
   view_books();
```

```
  break;
case 6:
    return_books();
    break;
case 7:
    exit(0);
    break;
default:
    printf("\nWRONG CHOICE\nENTER THE NUMBER BETWEEN 1 TO 7");
}
}
void add_books()
{
    system("cls");
int ch=1;
FILE *f1;
 f1=fopen("aboutbook.txt","a");
fseek(f1,0,SEEK_CUR);
while(ch==1)
    {
    printf("\nENTER THE INFORMATION OF THE BOOK\n");
    printf("\nBRANCH:");
scanf("%s",a.branch);
printf("\nBOOK ID:");
scanf("%s",a.bookid);
printf("\nBOOK NAME:");
scanf("%s",a.name);
printf("\nAUTHOR:");
scanf("%s",a.author);
printf("\nQUANTITY:");
scanf("%d",&a.quantity);
printf("\nRACK NUMBER:");
scanf("%d",&a.rack);
fwrite(&a,sizeof(a),1,f1);
```

```
  printf("\nTHE BOOK DETAILS ARE SUCCESSFULLY SAVED!\n");
printf("\nDO YOU WANT TO ADD MORE BOOKS?\tPRESS 1 IF YES OR 0 IF NO.\n");
scanf("%d",&ch);
system("cls");
    }
    fclose(f1);
    mainmenu();
}
void delete_books()
{
system("cls");
char delid[50];
int ch=1,q=0,s=0;
FILE *f1,*ft;
while(ch==1)
{
printf("\nENTER THE BOOK ID TO BE DELETED\n");
scanf("%s",delid);
f1=fopen("aboutbook.txt","r");
rewind(f1);
 while(fread(&a,sizeof(a),1,f1)==1)
        {
if(strcmp(a.bookid,delid)==0)
{
    printf("\nBOOK FOUND\n");
    printf("\nBOOK NAME:%s\n",a.name);
    printf("\nAUTHOR:%s\n",a.author);
    printf("\nRACK NUMBER:%d\n",a.rack);
q=1;
 break;
 }
}
if(q==1)
```

```
{
printf("\nARE YOU SURE OF DELETING THE BOOK\tIF YES PRESS 1\n");
 scanf("%d",&s);
 if(s==1)
 {
 ft=fopen("temp.txt","w");
 rewind(f1);
  while(fread(&a,sizeof(a),1,f1)==1)
    {
     if(strcmp(a.bookid,delid)!=0)
     {
           fseek(ft,0,SEEK_CUR);
           fwrite(&a,sizeof(a),1,ft);
     }
    }
    fclose(f1);
    fclose(ft);
       remove("aboutbook.txt");
    rename("temp.txt","aboutbook.txt");
 }
 printf("\nTHE BOOK IS SUCCESSFULLY DELETED\n");
 printf("\nDO YOU WANT TO DELETE MORE BOOKS\tPRESS 1 IF YES\n");
 scanf("%d",&ch);
}
else
{
    printf("\nNO BOOK WITH THE REUIRED BOOKID FOUND\n");
printf("\nENTER ANY KEY TO RETURN TO MAINMENU\n");
}
}
system("cls");
mainmenu();
}
void search_books()
```

```
{
system("cls");
int ch=0,e=1,f=0;
char b[50];
FILE *f1;
while(e==1)
{
printf("\n**********SEARCH BOOKS**********\n");
printf("\nSEARCH BY \n1.BOOK ID \n2.BOOK NAME\n");
scanf("%d",&ch);
switch(ch)
{
 case 1:
    {
        ch=1;
        char b[50];
        system("cls");
        printf("\nENTER THE BOOK ID\n");
        scanf("%s",b);
        f1=fopen("aboutbook.txt","r");
  while(fread(&a,sizeof(a),1,f1)==1)
  {
     if(strcmp(a.bookid,b)==0)
     {
         f=1;
     break;
     }
    }
    if(f==1)
    {
        printf("\nTHE BOOK IS AVAILABLE\n");
        printf("\nTHE BOOK DETAILS ARE:\nBRANCH:%s\nBOOK
NAME:%s\nAUTHOR:%s\nRACK:%d\n",a.branch,a.name,a.author,a.rack);
    break;
```

```
    }
    else
       printf("\nTHE BOOK NOT FOUND\n");
    }
    break;
 case 2:
   {
   system("cls");
  ch=2;
   char n[50];
   printf("\nENTER THE BOOK NAME\n");
  scanf("%s",n);
    f1=fopen("aboutbook.txt","r");
 while(fread(&a,sizeof(a),1,f1)==1)
 {
  if(strcmp(a.name,n)==0)
  {
      f=1;
  break;
  }
}
   if(f==1)
   {
        printf("\nTHE BOOK IS FOUND\n");
   printf("\nTHE BOOK DETAILS ARE:\nBRANCH:%s\nBOOK NAME:%s\nBOOK
ID:%s\nAUTHOR:%s\nRACK:%d\n",a.branch,a.name,a.bookid,a.author, a.rack);
  break;
   }
    else
  printf("\nTHE BOOK IS NOT FOUND\n");
   }
break;
}
printf("\nDO YOU WANT TO CONTINUE SEARCHING?\tPRESS 1 IF YES\n");
```

```
scanf("%d",&e);
}
system("cls");
mainmenu();
}
void issue_books()
{
system("cls");
int s,o,x=0,p=1,d=0,m=0,y=0,f=0,n,g,h,i;
char b[50];
time_t now;
time(&now);
struct tm *local=localtime(&now);
d= local->tm_mday;
m= local->tm_mon+1;
y= local->tm_year+1900;
FILE *f1,*fb;
printf("\n**********ISSUE BOOKS**********\n");
printf("\n1.ISSUE BOOKS\n2.VIEW ISSUED BOOKS\n3.SEARCH ISSUED BOOKS\n4.MAINMENU\n");
printf("\nENTER A CHOICE\n");
scanf("%d",&o);
switch(o)
{
case 1:
   {
   while(p==1)
   {
      system("cls");
       printf("\nENTER THE BOOK ID TO BE ISSUED\n");
       scanf("%s",b);
       f1=fopen("aboutbook.txt","r");
       fb=fopen("issuedbook.txt","a");
       fseek(fb,0,SEEK_CUR);
```

```
 while(fread(&a,sizeof(a),1,f1)==1)
 {
    if(strcmp(b,a.bookid)==0)
    {
       f=1;
    break;
    }
     }
 if(f==1)
 {
    s=a.quantity;
    printf("\n %d BOOKS ARE AVAILABLE\n",a.quantity);
       printf("\nBOOK NAME:%s\nAUTHOR:%s\n",a.name,a.author);
       printf("\nENTER THE STUDENT DETAILS\n");
       printf("\nSTUDENT NAME:\n");
       scanf("%s",a.stname);
       printf("\nUSN\n");
       scanf("%s",a.stusn);
      printf("\nYEAR\n");
      scanf("%d",&a.year);
printf("\nTHE ISSUED DATE IS %d/%d/%d\n",d,m,y);
n=m+1;
a.due.dd=d;
a.due.mm=n;
a.due.yy=y;
printf("\nTHE 8BOOK SHOULD BE RETURNED ON OR BEFORE
%d/%d/%d\n",a.due.dd,a.due.mm,a.due.yy);
fwrite(&a,sizeof(a),1,fb);
s=s-1;
  g=d;
  h=m;
  i=y;
  }
    else
```

```
        printf("\nTHE BOOK NOT FOUND\n");
printf("\nANY MORE BOOKS TO BE ISSUED?\tPRESS 1 IF YES\n");
scanf("%d",&p);
    }
    fclose(fb);
fclose(f1);
    mainmenu();
}
break;
case 2:
{
    int ch;
 system("cls");
 FILE *fb;
 printf("\n*****VIEW BOOKS*****\n");
 fb=fopen("issuedbook.txt","r");
 while(fread(&a,sizeof(a),1,fb)==1)
 {
printf("\nSTUDENT NAME:%s\nSTUDENT USN:%s\nBOOK
ID:%s\nNAME:%s\nISSUED DATE:%d/%d/%d\nDUE DATE:%d/%d/%d\n"
,a.stname,a.stusn,a.bookid,a.name,g,h,i,a.due.dd,a.due.mm,a.due.yy);
 }
 fclose(fb);
 printf("\nENTER 1 TO RETURN TO MAINMENU\n");
 scanf("%d",&ch);
 if(ch==1)
 {
     system("cls");
 mainmenu();
}
}
break;
case 3:
    {
```

```
    system("cls");
int e=1,f=0;
char b[50];
FILE *fb;
while(e==1)
{
printf("\n*****SEARCH ISSUED BOOKS*****\n");
       printf("\nENTER THE BOOK ID\n");
       scanf("%s",b);
       fb=fopen("issuedbook.txt","r");
 while(fread(&a,sizeof(a),1,fb)==1)
 {
    if(strcmp(b,a.bookid)==0)
    {
       f=1;
    }}

 if(f==1)
 {
    printf("\nTHE BOOK IS AVAILABLE\n");
       printf("\nSTUDENT NAME:%s\nSTUDENT USN:%s\nBOOK ID:%s\nBOOK
NAME:%s\nISSUED                                    DATE:%d/%d/%d\nRETURN
DATE:%d/%d/%d\n",a.stname,a.stusn,a.bookid,a.name,d,m,y,a.due.dd,a.due.mm,a.due.y
y);
 }
 else
 {
    printf("\nBOOK NOT FOUND\n");
 }
      printf("\nDO YOU WANT TO SEARCH MORE BOOKS\n");
      scanf("%s",&e);
}
issue_books();
}
```

```
case 4:
    system("cls");
    mainmenu();
  break;
system("cls");
mainmenu();
}
}
void return_books()
{
system("cls");
char delid[50];
int ch=1,q=0,s=0,d=0,m=0,y=0;
FILE *fb,*ft;
time_t now;
time(&now);
struct tm *local=localtime(&now);
d = local->tm_mday;
m = local->tm_mon+1;
y = local->tm_year+1900;
while(ch==1)
{
   printf("\nENTER THE BOOK ID TO BE RETURNED\n");
scanf("%s",delid);
 fb=fopen("issuedbook.txt","r");
rewind(fb);
while(fread(&a,sizeof(a),1,fb)==1)
        {
if(strcmp(a.bookid,delid)==0)
{
    printf("\nBOOK FOUND\n");
    printf("\nBOOK NAME:%s\n",a.name);
    printf("\nAUTHOR:%s\n",a.author);
    printf("\nTODAY'S DATE:%d/%d/%d\n",d,m,y);
```

```
    printf("\nDUE DATE:%d/%d/%d\n",a.due.dd,a.due.mm,a.due.yy);
q=1;
 break;
 }
}
if(q==1)
{
printf("\nARE YOU SURE OF RETURNING THE BOOK\tIF YES PRESS 1\n");
 scanf("%d",&s);
 if(s==1)
 {
 ft=fopen("delete.txt","w");
 rewind(fb);
  while(fread(&a,sizeof(a),1,fb)==1)
    {
     if(strcmp(a.bookid,delid)!=0)
     {
           fseek(ft,0,SEEK_CUR);
           fwrite(&a,sizeof(a),1,ft);
     }
    }
    fclose(fb);
    fclose(ft);
    remove("issuedbook.txt");
    rename("delete.txt","issuedbook.txt");
 }
 printf("\nTHE BOOK IS SUCCESSFULLY RETURNED\n");
 printf("\nDO YOU WANT TO DELETE MORE BOOKS\tPRESS 1 IF YES\n");
 scanf("%d",&ch);
}
else
{
    printf("\nNO BOOK WITH THE REUIRED BOOKID FOUND\n");
    printf("\nENTER ANY KEY TO GO TO THE MAINMENU\n");
```

```
ch=0;
}}
system("cls");
mainmenu();
}
void view_books()
{
   int ch;
system("cls");
 FILE *f1;
printf("\n**********VIEW BOOKS**********\n");
 f1=fopen("aboutbook.txt","r");
 while( fread(&a,sizeof(a),1,f1)==1)
 {
 printf("\nBRANCH: %s\nBOOK ID: %s\nBOOK NAME: %s\nAUTHOR: %s\nQUANTITY: %d\nRACK: %d\n",a.branch,a.bookid,a.name,a.author,a.quantity,a.rack);
}
 fclose(f1);
 printf("\nENTER 1 TO RETURN TO MAINMENU\n");
 scanf("%d",&ch);
 if(ch==1)
 {
system("cls");
 mainmenu();
}
}
```

Chapter 04

RESULTS AND DISCUSSION

4.1 Output Snapshots

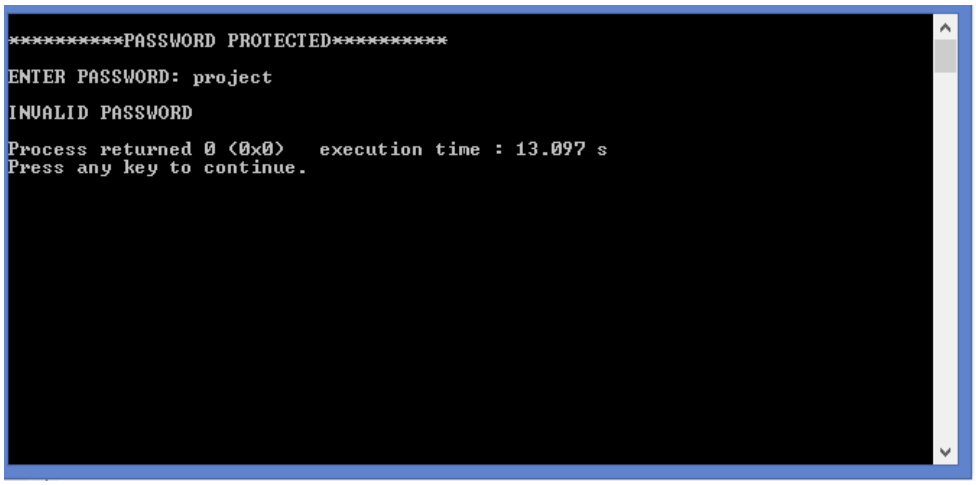

Fig 4.1 output when invalid password is entered

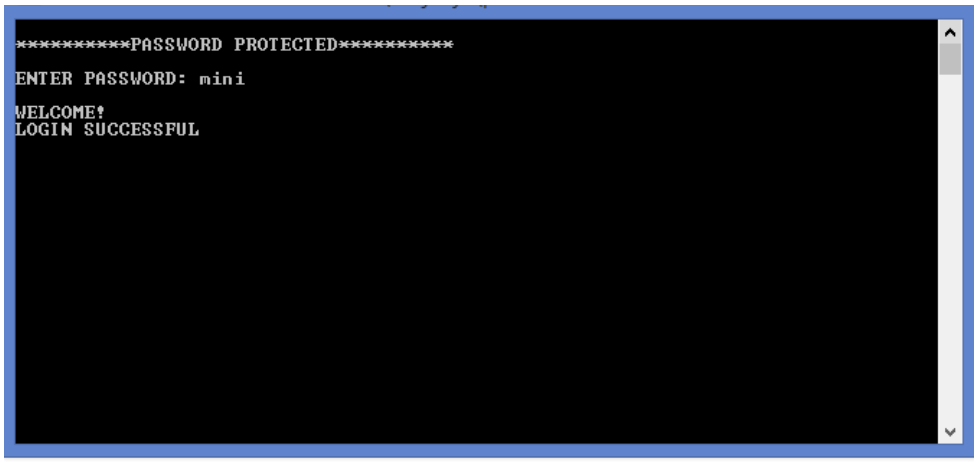

Fig 4.2 output when the correct password is entered

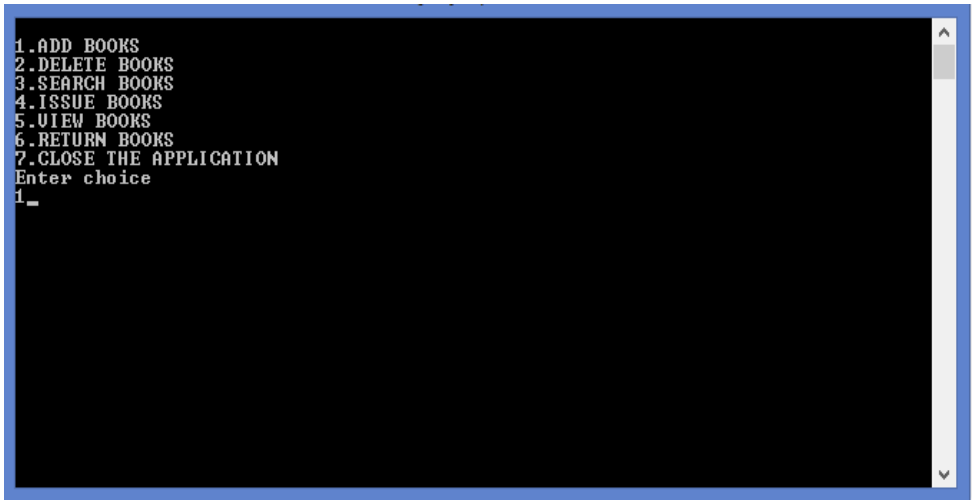

Fig 4.3 the mainmenu displayed and the users choice is 1

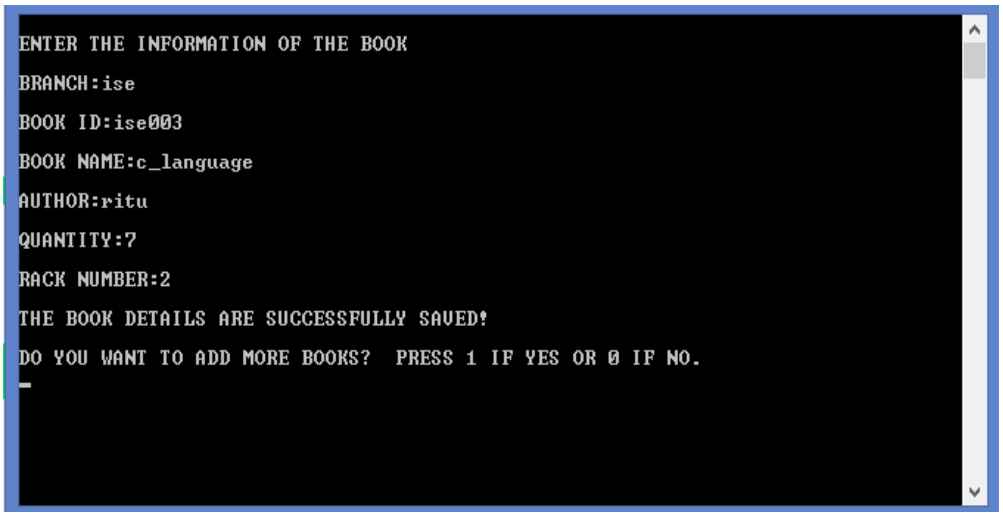

Fig 4.4 the book details entered by the user

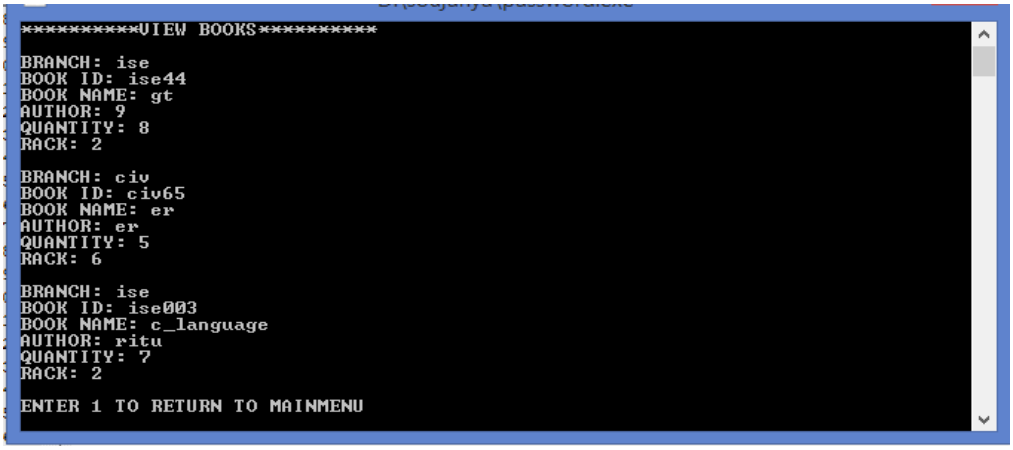

Fig 4.5 the book details is added into the file

Fig 4.6 the user gives option 2 and the delete books is called

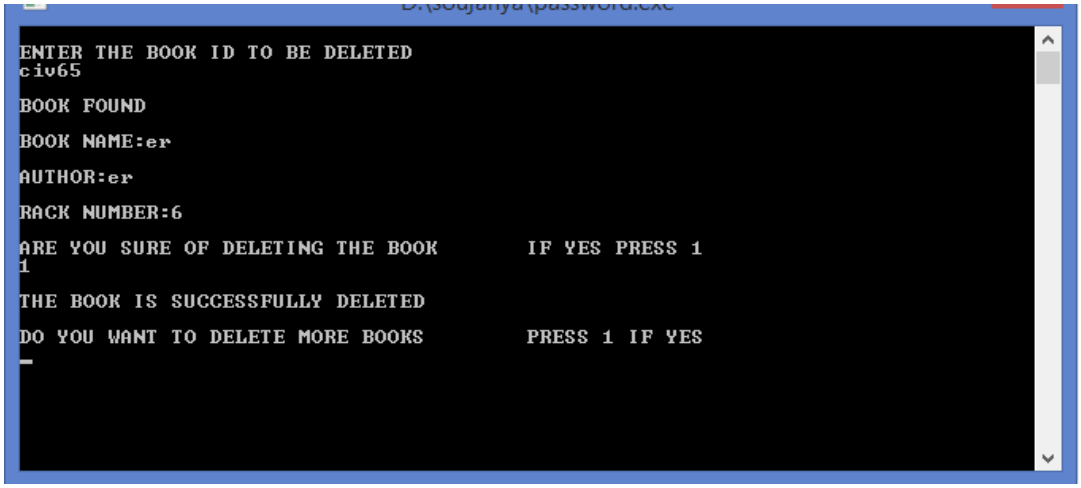

Fig 4.7 the bookid to be deleted is entered

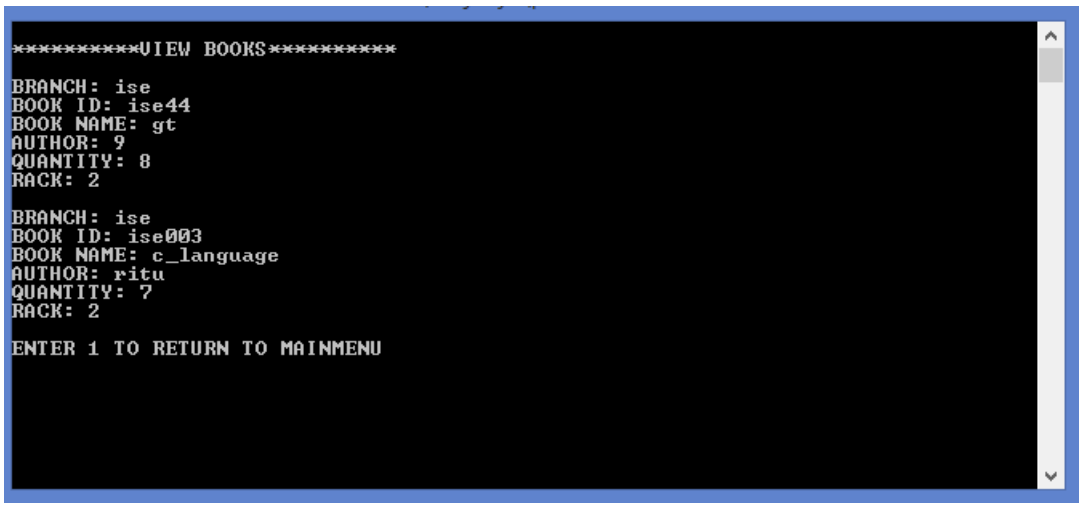

Fig 4.8 the book details of the given bookid is deleted.

Fig 4.9 the users enters choice 3

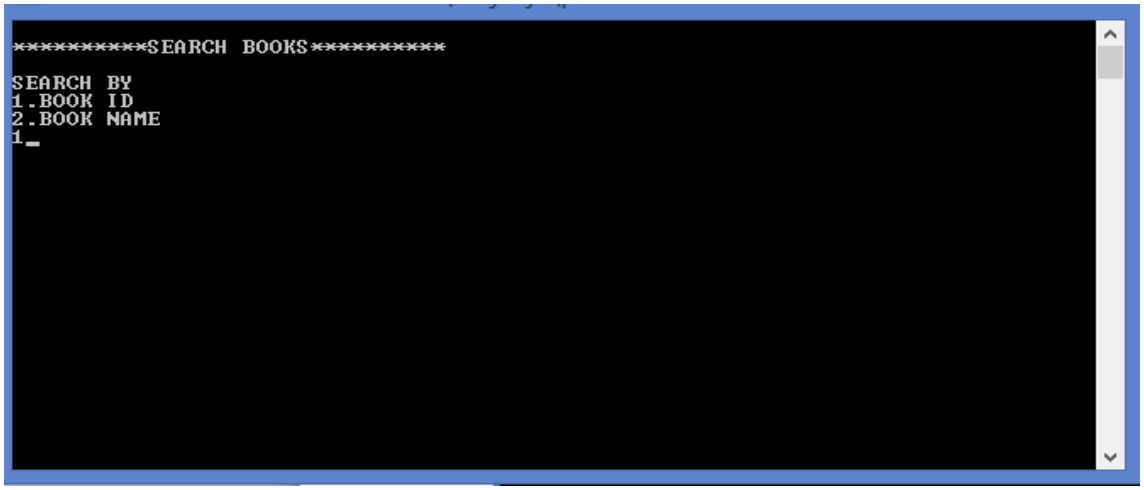

Fig 4.10 the users wants to search by the book id.

Fig 4.11 the bookid to be searched is asked and the details are displayed.

Fig 4.12 the book name to be searched is asked and the book details are displayed.

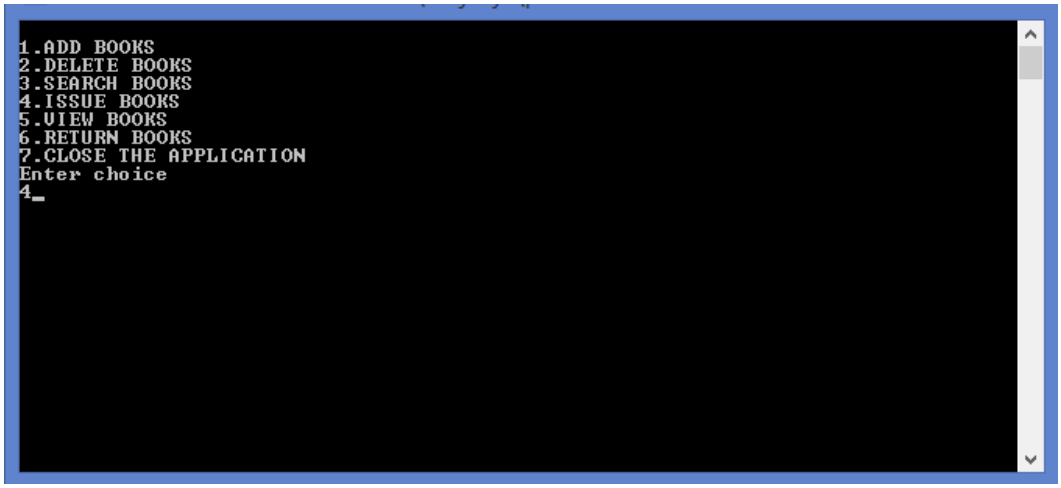

Fig 4.13 the user enters the choice 4

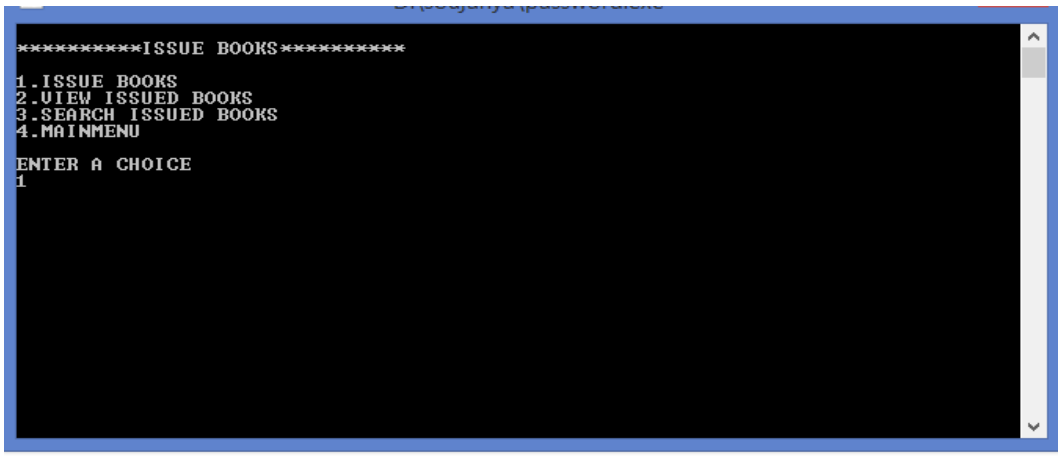

Fig 4.14 the users enters 1

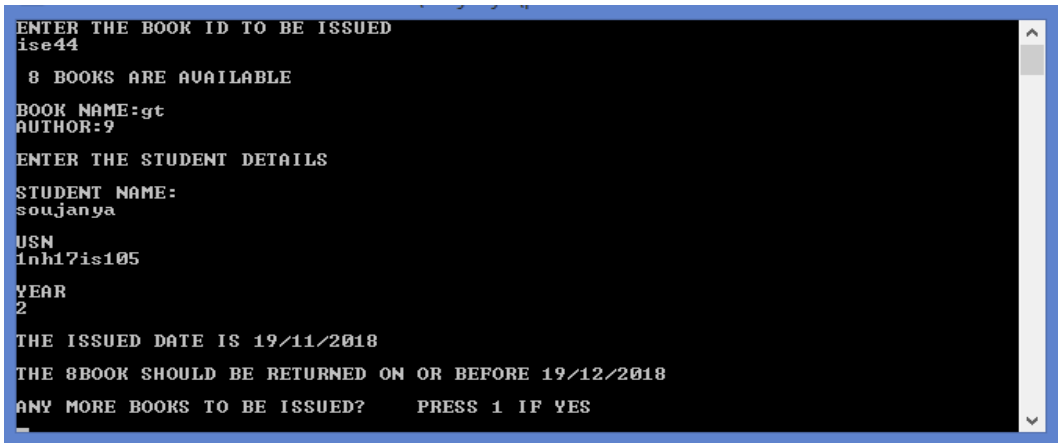

Fig 4.15 the student details along with the book details to be issued are displayed

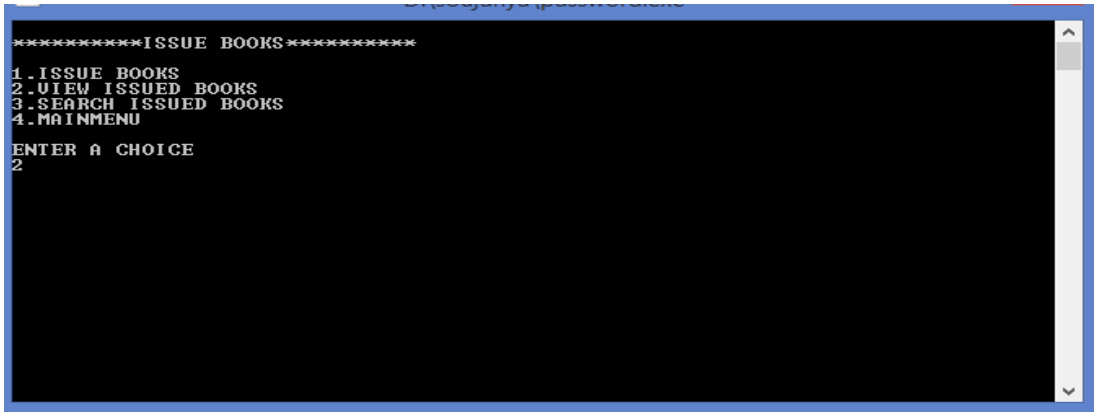

Fig 4.16 the issued book details are displayed.

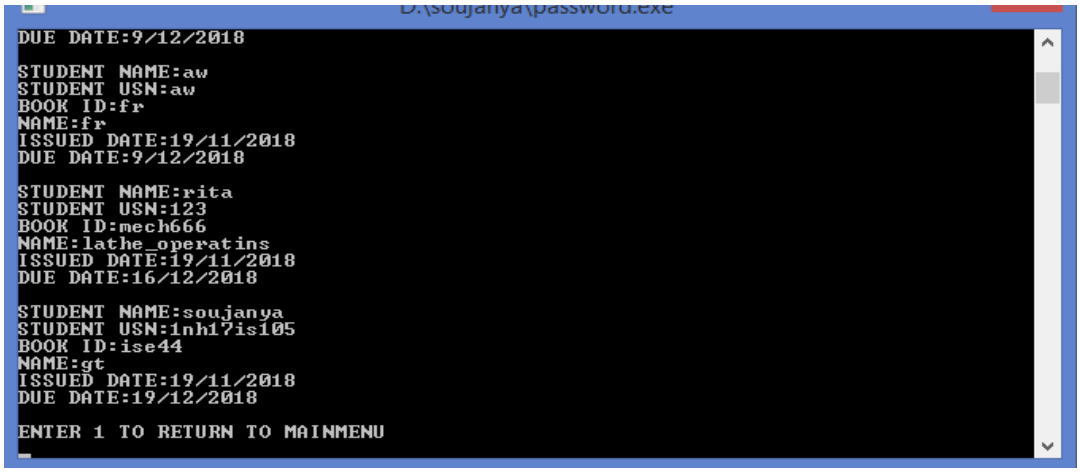

Fig 4.17 the issued books details are viewed.

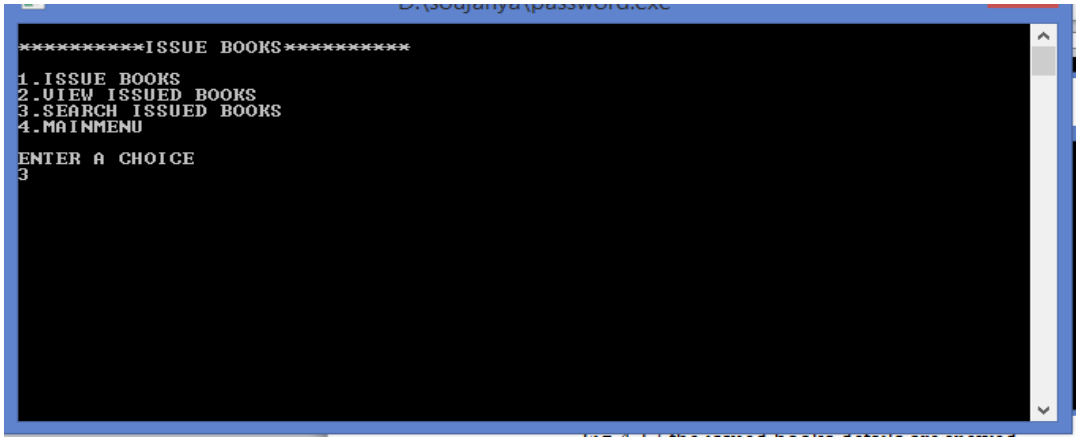

Fig 4.18 the user wants to search issued books

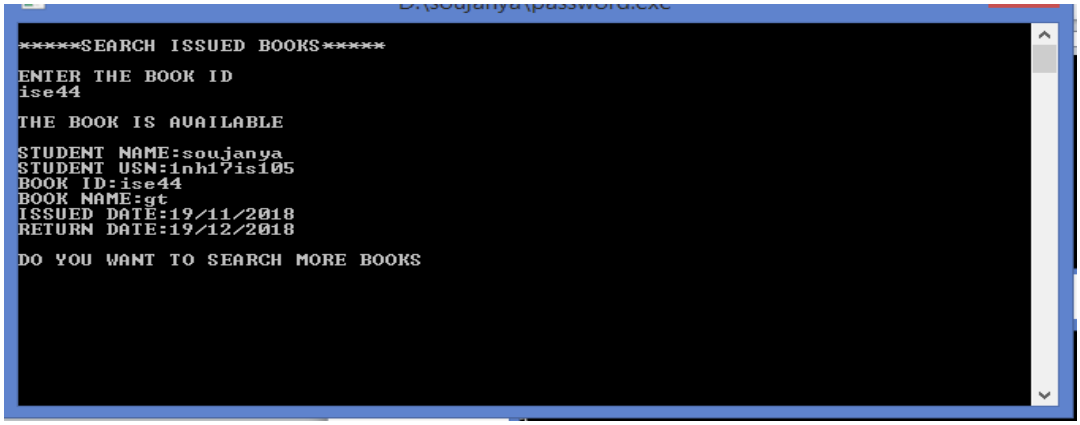

Fig 4.19 the book details of the bookid to be searched is displayed

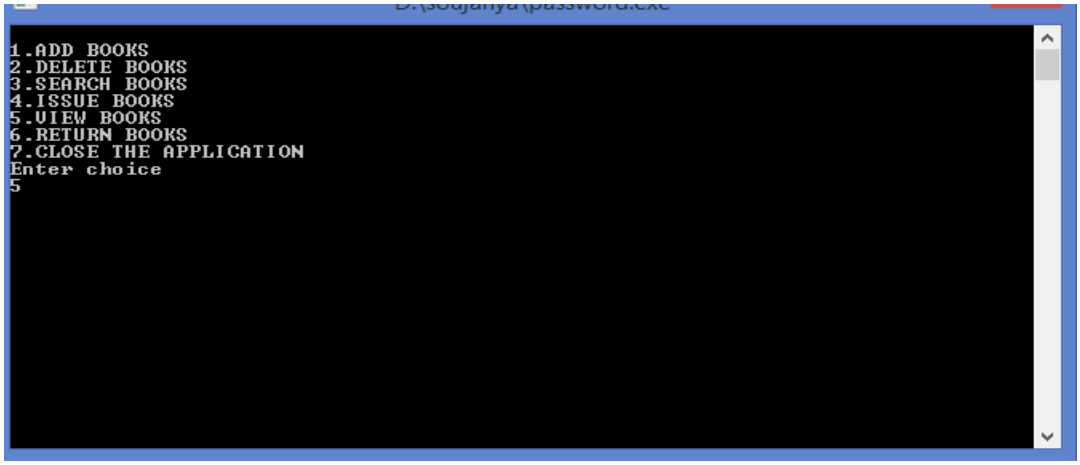

Fig 4.20 the user enters choice 5 and the view books function is called.

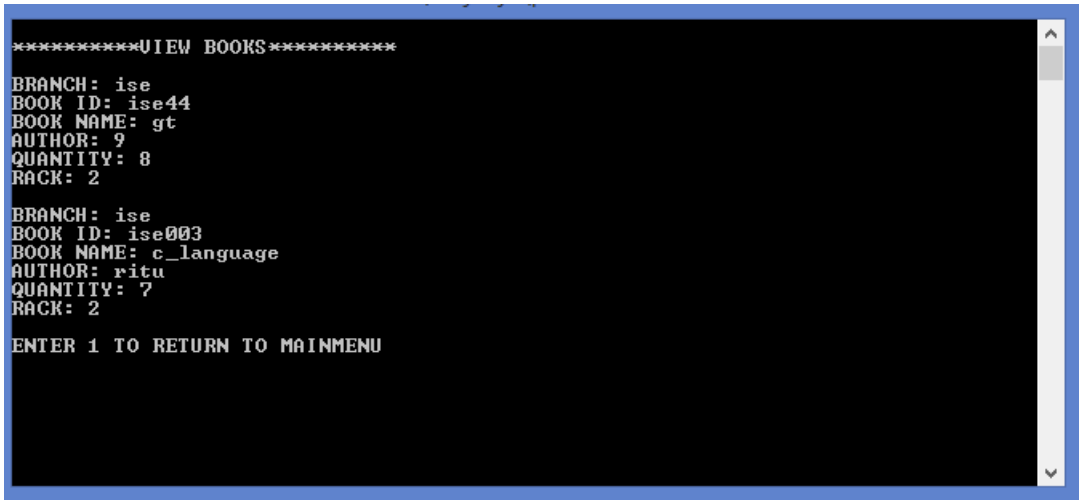

Fig 4.21 the book details are viewed

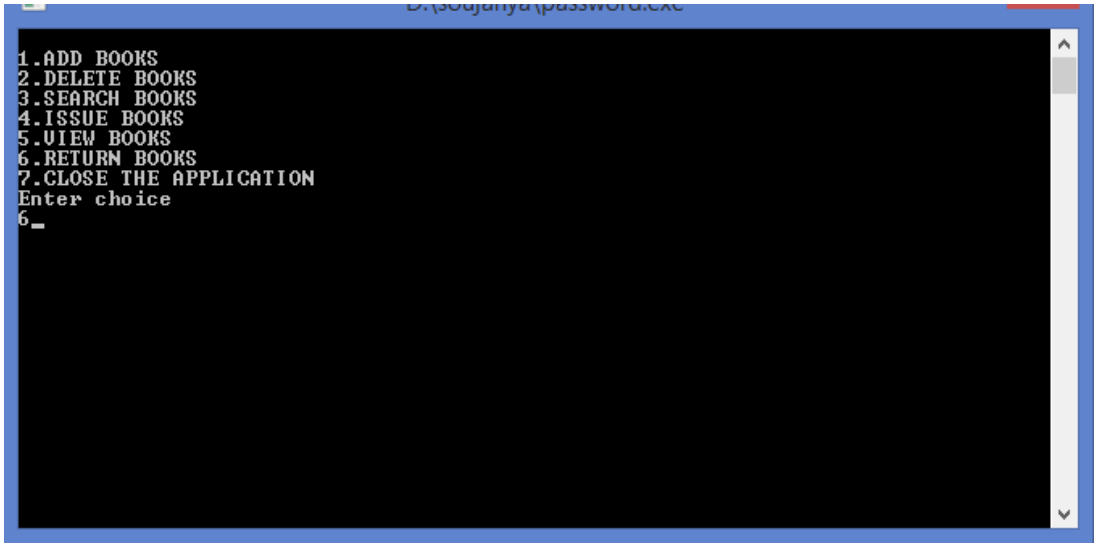

Fig 4.22 the user wants to return books

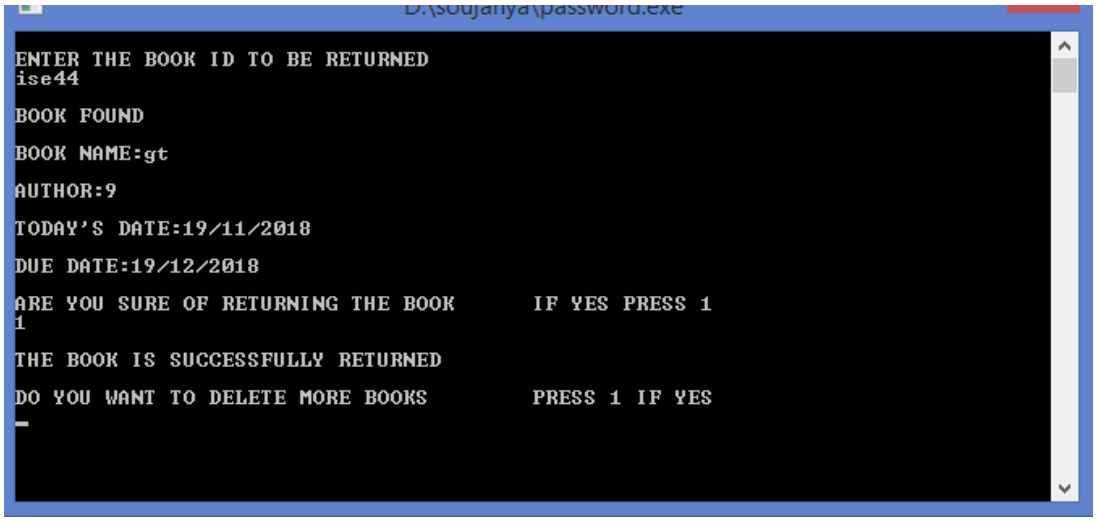

Fig 4.23 the book id to be returned is asked and the details is displayed

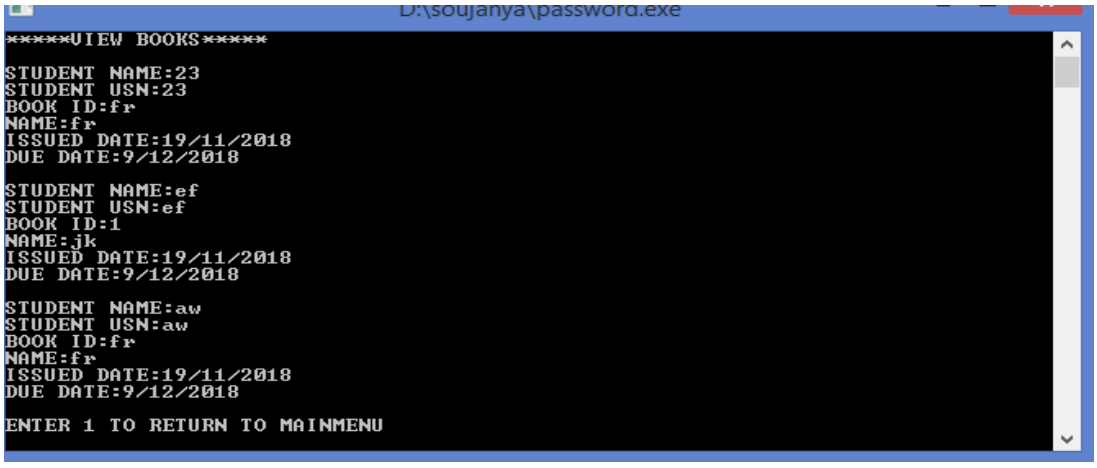

Fig 4.24 the book id returned is removed from the issuedbook.txt file

Chapter 05

CONCLUSION

As the existing manual library management system has its own set of disadvantages including lower efficiency, higher time consumption and requirement of higher manual labor. The above mentioned disadvantages can be eliminated by using the above proposed library management system. The library management system proves to be an user friendly application and has various advantages over the manual library management system which includes quick, easy and flexible generation of the book details, modification of the database is quick and effective, password protection which provides higher security to the application, safe storage of the information and many more.

REFERENCES

https://www.codewithc.com/mini-project-in-c-library-management-system/

http://www.ccodechamp.com/c-program-of-library-management-system-c-projects/

https://code-projects.org/library-management-system-c-programming-source-code/

https://www.geeksforgeeks.org/basics-file-handling-c/

https://fresh2refresh.com/c-programming/c-file-handling/

https://www.studytonight.com/c/file-input-output.php

https://www.programiz.com/c-programming/c-file-input-output

https://www.quora.com/

https://www.scribd.com/doc/53330583/Online-Library-System-project-report

https://www.google.co.in/search?rlz=1C1DFOC_enIN610IN610&ei=6bj2W7O-H8nRvgTPka6YAw&q=library+management+system+project+in+c+explained.&oq=library+management+system+project+&gs_l=psy-ab.3.0.35i39l2j0i67l4j0j0i67l3.4306.4306..6692...0.0..0.160.160.0j1.....0....1..gws-wiz.....0i71.aYhlPtBR8o0