

A MINI PROJECT ON
LIBRARY MANAGEMENT SYSTEM

Batch no:18,19,20

BATCH MEMBERS :

| | |
|-----------------------------|------------------------------|
| K.HASITHA SAI(17121A1252) | M.SAI GRYSHYANTH(17121A1257) |
| L.GOPI (17121A1253) | M.VISHNU(17121A1258) |
| M.MYTHILI(17121A1254) | M.V.RANGASWAMY(17121A1259) |
| M.SATHISH KUMAR(17121A1255) | M.HEPSIBAH(17121A1260) |
| M.SAI NITHYA(17121A1256) | |



SREE VIDYANIKETHAN ENGINEERING COLLEGE

(autonomous)

(Affiliated to JNTUA, Ananthapur, Approved by AICTE, Accredited by NBA & NAAC)
Sree Sainath Nagar, Tirupati – 517 102, A.P., INDIA

2018 -2019

TABLE OF CONTEXT

| | |
|---------------------|---|
| • INTRODUCTION | 1 |
| • MODULES | |
| • REQUIREMENTS | |
| • FUTURE INFERENCES | |
| • CONCLUSION | |

ABSTRACT

- The main objective of the mini project library management system is to provide service to the library in maintaining records of the books present in library. It handles the entire activity of a Library.
- It can perform some operations adding a book , searching a book , deleting a book record , updating a book record ,issuing a book record , viewing a book record.
- Each book holds details like category , book name , author , id , quantity , price ,rack no,
- Each issued book contains details like student name , student roll no , issued date month and year , return date , month , year .
- The records here are maintained in files .
- The software keeps track of all the information about the books and their complete details.

INTRODUCTION



- In the era of digitalisation it is very hard to maintain the huge record of books manually. This project helps us to maintain the data regarding books in a library more effectively.
- Generally, it is very difficult to maintain records of books present in library such as book category, name, quantity , etc., physically in a book.
- Library Management System provides user friendly environment and helps in effective maintenance of Library.
- A digitalized maintenance of records would be a best solution to make the task easier .
- In order to store the information of books present in library and performing the more operations adding an additional benefits , a library management system is designed.
- This project works with objective of providing a best platform for the maintenance of information regarding books.

MODULES OF THE PROJECT

The code initially provides user with a main menu which allows user to perform different kinds of operations. The operations of the Library Management are as follows:

- ADDING BOOKS
- SEARCHING BOOKS
- UPDATION OF BOOKS
- VIEWING OF BOOK LIST
- DELETION OF BOOKS
- ISSUING BOOKS

```
+++++
1.Enter a record
2.view all records
3.Search a book record
4.Update a book record
5.Delete a book record
6.Issue a Book
0.exit
Enter your choice
+++++
```

Module-1(ADDITION OF BOOKS):

This module is helpful for inserting new books to the record of the Library. The book is stored in the record based on some criteria like branches. Different books are In this module the addition operation can be done which saves the added book details in the file holding each and every record of books entered.

```

9.exit
Enter your choice
+++++
1
Categories are :
1.Computer
2.Electronics
3.Electrical
4.Civil
5.Mechanical
6.Architecure
Enter the category number
1
enter the book ID
101
enter the book name
Drillingc
Enter the Book Price
120.4
Enter the author name of the book
Richard
Enter the quantity
4
Enter the rack number
2
Do you want to enter any other record of a book
1.Yes 0.no0

```

As shown above, this module initially asks for the branch in which it has to be kept available. Later, it takes other details like book name, book ID.....

- INPUT: option the specifies to which branch the book belongs and the details of book like book title, author, price, quantity, rack no.
- OUTPUT: The detail given as input is stored in a file.

Module-2(Searching books):

This operation is handy for the users who wants a particular book or to know whether a particular book is available at that instant or not. In this module the books entered into record is searched whether it is present in the book list if present returns the book details to standard output screen. The books can be searched based on book name, book ID.....which are unique for different books.

```
0.exit
Enter your choice
+++++
3
Enter the Book title to search
Drillingc

-----
category is :: Computer
Book ID :: 101
Book Title :: Drillingc
Book Price :: 120.400002
Book Author :: Richard
Number of Books :: 4
Book rack number :: 2
-----
```

- INPUT : Book name to be searched is given as a input
- OUTPUT: if input book name is present in the list the book details are displayed , else an searched book doesn't exist message is displayed.

Module-3(Updation of books):

This module is useful for the administrator to maintain the records of books up to date by updating their details. In this module the updating or editing of book details present in the book list is done. To update the details of book, firstly we have to enter the name of the book and then we can edit other details.

```

6.Issue a Book
0.exit
Enter your choice
+++++
4
Enter the book title to update
Drillingc
The book catogery is Computer
enter the book ID
101
enter the book name
Drillingc++
Enter the Book Price
123.4
Enter the author name of the book
RichardThies
Enter the quantity
3
Enter the rack number
1

```

- INPUT : book name is given as an input for which the details need to be updated. If present then the details you wanted to update is given as input.
- OUTPUT : the output of this module is , updations of details of book present is saved in file if the input book name entered is present in the book list , else a error message is displayed.

Module-4(View book record):

The books issued to different students are stored in a record. In this module the book details present in the record is viewed at the standard output screen.

```
6.Issue a Book
0.exit
Enter your choice
+++++
2

Records are ::

-----
category is :: Computer
Book ID :: 101
Book Title :: Drillingc
Book Price :: 120.400002
Book Author :: Richard
Number of Books :: 4
Book rack number :: 2
-----
```

As shown above this operation displays all the details and availability of the books issued.

- INPUT : here no input is taken for this module after the selection of this operation the output is displayed
- Output : the output here is the details of each and every book record entered into the list displayed on the standard output screen.

Module-5(deletion of book):

This module helps the administrator to maintain the record more effectively by updating the book record based on their availability by the deleting the details of the book .The entire record of specified book given as input is deleted from the record.

- INPUT : the input here is the name of the book you wanted to delete
- OUTPUT : If the book name entered is present in book list the file content is updated by deleting the existing book record , if not an error message is displayed allowing to delete further more if u want.

Module-6(issuing of books):

This function holds all the operations regarding the issue of books and maintains data.

Using this function , we can view, edit, and search the books that has been issued.

In this module performs 4 operations say,

- Issuing a book
- Viewing issued book
- Deleting issued book
- Searching issued book

```
+++++
1.Enter a record
2.view all records
3.Search a book record
4.Update a book record
5.Delete a book record
6.Issue a Book
0.exit
Enter your choice
+++++
6
=====
1.Issue a Book
2.View Issued Books
3.Search Issued Books
4.Remove a Issued book
0.exit
=====
Enter your choice
```

Above figure shows the menu of **Issue a Book** operation, in which we can select particular operation.

```

=====
1.Issue a Book
2.View Issued Books
3.Search Issued Books
4.Remove a Issued book
0.exit
=====
Enter your choice
1
Enter the book id to be issued
101
Book record available
There are 3 books available
Book name is Drillingc++
Enter student name and Roll number
sarvash
15121a1255
Enter today's date, month and year
9
10
2018
Book to returned on date : 24 - 10 - 2018
=====

```

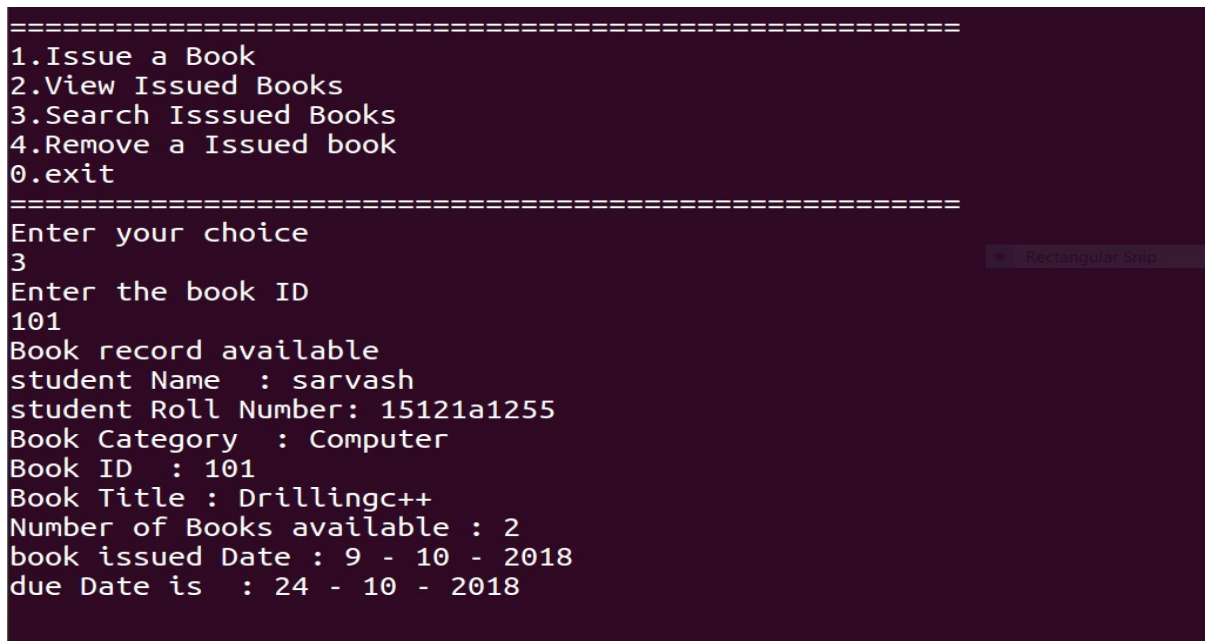
To issue a book this module initially requires book id. Then it shows whether the book is available or not. If the book is available it takes student name & roll number and issues the book. It also returns the date on which it has to be returned.

```

=====
1.Issue a Book
2.View Issued Books
3.Search Issued Books
4.Remove a Issued book
0.exit
=====
Enter your choice
2
student Name : sarvash
student Roll Number: 15121a1255
Book Category : Computer
Book ID : 101
Book Title : Drillingc++
Number of Books available : 2
book issued Date : 9 - 10 - 2018
due Date is : 24 - 10 - 2018
=====

```

View issued books operations is helpful to see the record of all the books that have been issued. The above image shows how the book record is displayed.



```

=====
1.Issue a Book
2.View Issued Books
3.Search Issued Books
4.Remove a Issued book
0.exit
=====
Enter your choice
3
Enter the book ID
101
Book record available
student Name : sarvash
student Roll Number: 15121a1255
Book Category : Computer
Book ID : 101
Book Title : Drillingc++
Number of Books available : 2
book issued Date : 9 - 10 - 2018
due Date is : 24 - 10 - 2018

```

Search issued books option is handy to know whether a particular book has been issued or not. If issued, it shows to whom and when it has been issued.

Remove a issued book option is useful to remove the records of the books that have been written and maintained the data up to date.

- INPUT : option for specific operation to perform is given as input
 1. Takes book name, student name and student roll no. date of book issued as input.
 2. No input
 3. Takes book name as input
 4. Takes book name as input
- OUTPUT :
 1. Stores the details entered into a file
 2. Views all the books and details present in the record
 3. Deletes the book name entered and updates the file
 4. Displays the details of specific book searched if present

PROJECT CODE : #include<stdio.h>

#include<stdlib.h>

#include<string.h>

#define RETURNTIME 15

struct date

```
{  
  
    int mm,dd,yy;  
  
};
```

struct book

```
{  
  
    int bookid;  
  
    char title[19];  
  
    float price;  
  
    char stuname[19];  
  
    char rollnumber[19];  
  
    char author[19];  
  
    int quantity;  
  
    int count;  
  
    int rackno;  
  
    char *cat;
```

```
    struct date issued;

    struct date duedate;

};

struct book b1;

char
catagories[][15]={"computer","electronics","electrical","civil","mechanical","ar
chitecture"};

void getbookdata()

{

    int s,c;

    FILE *fp;

    fp=fopen("book.txt","a+");

    c=1;

    while(c==1)

    {

        printf("1.computer\n2.electronics\n3.electrical\n4.civil\n5.mechanical\n6.
architecture\n");

        printf("enter the category number\n");

        scanf("%d",&s);

        b1.cat=catagories[s-1];

        printf("enter the book ID\n");
```

```
scanf("%d",&b1.bookid);

printf("enter the book name\n");

fflush(stdin);

scanf("%s",b1.title);

printf("enter book price\n");

scanf("%f",&b1.price);

printf("enter the author name of the book\n");

fflush(stdin);

scanf("%s",b1.author);

printf("enter the quantity\n");

fflush(stdin);

scanf("%d",&b1.quantity);

printf("enter the rack number\n");

scanf("%d",&b1.rackno);

fwrite(&b1,sizeof(b1),1,fp);

printf("Do you want to enter any other record of a
book\n\t\t1.yes\t\t0.no");

fflush(stdin);

scanf("%d",&c);

}

fclose(fp);
```



```
}
```

```
void viewallbooks()
```

```
{
```

```
    FILE *fp;
```

```
    fp=fopen("book.txt","r");
```

```
    if(fp==NULL)
```

```
    {
```

```
        printf("error in opening the file may be it not exist\n");
```

```
        exit(1);
```

```
    }
```

```
    rewind(fp);
```

```
    while((fread(&b1,sizeof(b1),1,fp))>0)
```

```
    {
```

```
        printf("%s\t\t",b1.cat);
```

```
        printf("%d\t\t%s\t\t%f\n",b1.bookid,b1.title,b1.price);
```

```
        printf("%s\t\t%d\t\t%d\n\n",b1.author,b1.quantity,b1.rackno);
```

```
    }
```

```
    fclose(fp);
```

```
}
```

```
void searchbookrecord(char *t)
```

```
{
```

```
    int counter=0;
```

```
    FILE *fp;
```

```
    fp=fopen("book.txt","r");
```

```
    if(fp==NULL)
```

```
    {
```

```
        printf("error in opening the file may be it not exist\n");
```

```
        exit(1);
```

```
    }
```

```
    rewind(fp);
```

```
    while((fread(&b1,sizeof(b1),1,fp))>0)
```

```
    {
```

```
        if(!strcmp(t,b1.title))
```

```
        {
```

```
            printf("%s\t\t",b1.cat);
```

```
            printf("%d\t\t%s\t\t%f\n",b1.bookid,b1.title,b1.price);
```

```
            printf("%s\t\t%d\t\t%d\n",b1.author,b1.quantity,b1.rackno);
```

```
        counter++;

    }

}

if(counter==0)

printf("the book you are searching for %s not found\n",t);

fclose(fp);

}

void updateabookrecord(char *t)

{

    long c;

    FILE *fp;

    fp=fopen("book.txt","r+");

    if(fp==NULL)

    {

        printf("error in opening the file may be it not exist\n");

        exit(1);

    }

    rewind(fp);

    while((fread(&b1,sizeof(b1),1,fp))>0)
```

```
{  
  
    if(!strcmp(t,b1.title))  
  
    {  
  
        printf("the book category is %s",b1.cat);  
  
        printf("enter the book ID\n");  
  
        scanf("%d",&b1.bookid);  
  
        printf("enter the book name\n");  
  
        fflush(stdin);  
  
        scanf("%s",b1.title);  
  
        printf("enter book price\n");  
  
        scanf("%f",&b1.price);  
  
        printf("enter the author name of the book\n");  
  
        fflush(stdin);  
  
        scanf("%s",b1.author);  
  
        printf("enter the quantity\n");  
  
        fflush(stdin);  
  
        scanf("%d",&b1.quantity);  
  
        printf("enter the rack number\n");  
  
        scanf("%d",&b1.rackno);  
  
        c=ftell(fp) ;
```

```
        fseek(fp,c,0);

        fwrite(&b1,sizeof(b1),1,fp);

        break;

    }

}

fclose(fp);

}
```

```
void deleteabookrecord(char *t)

{

    FILE *fp,*tem;

    tem=fopen("temp.txt","w+");

    if(tem==NULL)

    {

        printf("error in opening the file may be it not exist\n");

        exit(1);

    }

    fp=fopen("book.txt","r");

    if(fp==NULL)

    {
```

```
        printf("error in opening the file may be it not exist\n");

        exit(1);

    }

    rewind(fp);

    while((fread(&b1,sizeof(b1),1,fp))>0)

    {

        if(strcmp(t,b1.title))

        {

            fwrite(&b1,sizeof(b1),1,tem);

        }

    }

    fclose(fp);

    fclose(tem);

    remove("book.txt");

    rename("temp.txt","book.txt");

}

void issueabook()

{

    int test,c=0;
```

```

int ch;

FILE *fp,*tem,*iss;

while(1)

{

    printf("*****
*****");

    printf("1.issue a book\n 2.view issued book\n 3.search issued
books\n 4.remove issued book\n0.exit\n");

    printf("enter your choice\n");

    scanf("%d",&ch);

    switch(ch)

    {

    case1 :

        printf("enter the book id to be issued \n");

        scanf("%d",&test);

        fp=fopen("book.txt","r");

        iss=fopen("issue.txt","a+");

        while((fread(&b1,sizeof(b1),1,fp))>0)

        {

            if(b1.bookid==test)

```

```
{  
  
    if(b1.quantity==0)  
  
    {  
  
        printf("all books are sold out\n");  
  
        break;  
  
    }  
  
    printf("book record available\n");  
  
    printf("there are %d books available\n",b1.quantity);  
  
    b1.quantity=1;  
  
    printf("book name is %s\n",b1.title);  
  
    printf("enter student name and roll number\n");  
  
    scanf("%s %d",b1.stuname,b1.rollnumber);  
  
    printf("enter todays date month and year\n");  
  
    scanf("%d%d%d",&b1.issued.dd,&b1.issued.mm,&b1.issued.yy);  
  
    b1.duedate.dd=b1.issued.dd+RETURNTIME;  
  
    b1.duedate.mm=b1.duedate.mm;  
  
    b1.duedate.yy=b1.issued.yy;  
  
    if(b1.duedate.dd>30)  
  
    {  
  
        b1.duedate.mm+=b1.duedate.dd/30;
```



```

        b1.duedate.dd-=30;

    }

    if(b1.duedate.mm>12)

    {

        b1.duedate.yy+=b1.duedate.mm/12;

        b1.duedate.mm-=12;

    }

    printf("book to returned on
date:%d%d%d\n",b1.duedate.dd,b1.duedate.mm,b1.duedate.yy);

    fwrite(&b1,sizeof(b1),1,iss);

    c=1;

    break;

}

}

fclose(fp);

fclose(iss);

if(c==0)

printf("\nbook not found with the id number %d\n",test);

break;

case2:

    iss=fopen("issue txt","r");

```

```

        rewind(iss);

        while((fread(&b1,sizeof(b1),1,iss))>0){

            printf("student name:%s\n",b1.stuname);

            printf("student roll number:%s\n",b1.rollnumber);

            printf("boook category:%s\n",b1.cat);

            printf("book id:%s\n",b1.bookid);

            printf("book title:%s\n",b1.title);

            printf("number of books available : %d\n",b1.quantity);

            printf("book issued date : %d - %d - %d
\n\n\n",b1.issued.dd,b1.issued.mm,b1.issued.yy);

            printf(" book return date : %d - %d - %d
\n\n\n",b1.duedate.dd,b1.duedate.mm,b1.duedate.yy);

        }

    fclose(iss);

    break;

    case 3 :

        iss=fopen("issue.txt","r");

        rewind(iss);

        printf("enter the book id\n");

        scanf("%d",&test);

        while((fread(&b1,size(b1),1,iss))>0)

```

```
{  
  
    if(b1.bookid==test)  
  
    {  
  
        printf("book record available \n");  
  
        printf("student name : %s\n",b1.stuname);  
  
        printf("student roll num : %s\n",b1.rollnumber);  
  
        printf("book category : %s\n",b1.cat);  
  
        printf("book id : %d\n",b1.bookid);  
  
        printf("book title : %s\n",b1.title);  
  
        printf("number of books available :  
%d\n",b1.quantity);  
  
    }  
  
}  
  
break;  
  
case 4:  
  
    tem=fopen("temp.txt","w+");  
  
    if(tem==NULL)  
  
    {  
  
        printf(" error in opening a file\n");  
  
        exit(1);  
  
    }
```

```
iss=fopen("issue.txt","r");

if(iss==NULL)

{

    printf("error in opening a file \n");

    scanf("%d",&test);

    rewind(iss);

    while((fread(&b1,sizeof(b1),1,iss))>0)

    {

        if(b1.bookid!=test)

        {

            fwrite(&b1,sizeof(b1),1,tem);

        }

    }

    fclose(iss);

    fclose(tem);

    remove("issue.txt");

    rename("temp.txt","issue.txt");

    break;

}

if(ch==0)
```

```
        break;

    }

}

int main()

{

    char title[19];

    int ch;

    while(1)

    {

        printf("1.enter a record\n2.view all record\n3.search a book\n4.view all\n5.add a\n6.delete a\n7.issue a\n8.view all\n9.exit\n");

        printf("4.update a book record\n5.delete a book record\n6.issue a book\n0.exit\n");

        printf("enter your choice : \n");

        scanf("%d",&ch);

        switch(ch)

        {

            case 1 : getbookdata();

                    break;

            case 2:

                    viewallbooks();
```

```
break;
```

```
case 3:
```

```
printf("enter the book title to search\n");
```

```
scanf("%s",title);
```

```
searchabookrecord(title);
```

```
break;
```

```
case 4 :
```

```
printf("enter the book title to update\n");
```

```
scanf("%s",title);
```

```
updateabookrecord(title);
```

```
break;
```

```
case 5 :
```

```
printf("enter the book title to delete\n");
```

```
scanf("%s",title);
```

```
deleteabookrecord(title);
```

```
break;
```

```
case 6 : issueabook();
```

```
break;
```

```
case 0 :
```

```
exit(0);
```

default :

```
        printf("invalid choice\n");

        break;

    }

}

}
```

REFERENCES:

- PROGRAMMING WITH C LANGUAGE , BALAGURUSWAMY.
- BEGGINNING OF LINUX PROGRAMING
- NEIL MATHEW AND RICHARD STONES , WILLEY DREAMTECH , FOURTH EDITION , 2008.
- ONLINE REFERENCES : CODE WITH C
- SOFTWARE DOWNLOADED :

<https://www.ubuntu.com/#download>.

Future inferences:

- The library management system mini project designed so far is more advantages for the admin to know the details and perform operations on books present in library
- In future we can update more operations that serves to be more advantageous to the user for getting information of the books present in library
- The operations may include like locking a book , claim for book , check availability.....etc.,

CONCLUSION:

- With this project we conclude that it provides a best platform for the maintenance of information regarding books present in the library.
- Taking consideration to the future references it can also provides services to the users to know the details of book , locking a book neededetc., operations.
- With the designing of this project the c concepts and Linux concepts were revised and