

**A Project Report  
on**

**Library management system project in C**

*Submitted in partial fulfillment of the  
requirement for the award of the degree of*

**Bachelor of Technology in Computer Science and  
Engineering**



(Established under Galgotias University Uttar Pradesh Act No. 14 of 2011)

**Under the Supervision of  
Dr. Pallavi Jain  
Assistant Professor  
Department of Computer Science and Engineering**

**Submitted By  
19SCSE1010728- Babloo Kumar  
19SCSE1010523- Bably Kumari**

**SCHOOL OF COMPUTING SCIENCE AND ENGINEERING  
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING  
GALGOTIAS UNIVERSITY, GREATER NOIDA, INDIA  
DECEMBER - 2021**



**SCHOOL OF COMPUTING SCIENCE AND  
ENGINEERING  
GALGOTIAS UNIVERSITY, GREATER NOIDA**

**CANDIDATE'S DECLARATION**

I/We hereby certify that the work which is being presented in the thesis/project/dissertation, entitled "**Library management system project in C**" in partial fulfillment of the requirements for the award of the B.Tech submitted in the School of Computing Science and Engineering of Galgotias University, Greater Noida, is an original work carried out during the period of month, Year to Month and Year, under the supervision of Name... Designation, Department of Computer Science and Engineering/Computer Application and Information and Science, of School of Computing Science and Engineering, Galgotias University, Greater Noida

The matter presented in the thesis/project/dissertation has not been submitted by me/us for the award of any other degree of this or any other places.

Babloo Kumar – 19SCSE1010728

Bably Kumari – 19SCSE1010523

This is to certify that the above statement made by the candidates is correct to the best of my knowledge.

Supervisor Name

Designation

**CERTIFICATE**

The Final Thesis/Project/ Dissertation Viva-Voce examination of Babloo Kuma -19SCSE1010728,  
Bably Kumari – 19SCSE1010523 has been held on \_\_\_\_\_ and his/her work is  
recommended for the award of B. Tech.

**Signature of Examiner(s)**

**Signature of Supervisor(s)**

**Signature of Project Coordinator**

**Signature of Dean**

Date:

Place: Greater Noida

## ABSTRACT

The mini-project “Library management system project in C” is a console application using the C programming language. This project compiled in Visual Studio/Code Blocks with the GCC compiler. In this console application, you can do basic library management task like adding the book, view the added book, search the books, etc.

This application based on file handling in C, where I have used a file-related function like fopen, fread, fwrite, etc. Good thing is that “Library management system project” is password-protected, so only authorized person able to login in this application.

Also to increase the readability I have broken the application in different function. Each function of the project extensively use the **file handling function**, so it is also a great project to understand file handling.

**Keywords:** Management System, .Net, C Language, DBMS

## Table of Contents

<b>Title</b>	<b>PageNo.</b>
Candidates Declaration	I
Acknowledgement	II
Abstract	III
Contents	IV
List of Table	V
List of Figures	VI
Chapter 1 Introduction.....	8
Chapter 2 Literature Review.....	10
Chapter 3 Methodlogy .....	12
Chapter 4 ER diagram for library management system.	16
Chapter 5 Conclusion and Future Scope.....	17
5.1 Conclusion	
5.2 Future Scope	
Reference	
Publication/Copyright/Product	

### List of Figures

<b>S.No.</b>	<b>Title</b>	<b>Page No.</b>
<b>1</b>	<b>Block Schematic</b>	<b>12</b>
<b>2</b>	<b>Welcome page, menu for LMS</b>	<b>13</b>
<b>3</b>	<b>Add Book, Search Book, View Book of LMS</b>	<b>14</b>
<b>4</b>	<b>ER- diagram</b>	<b>16</b>

## **ACKNOWLEDGEMENT**

Primarily I might thank God for having the ability to finish this mission with success. Then I would love to thank my mission manual **Dr. Pallavi Jain**, whose treasured steering has been those that helped me patch this mission and make it complete evidence success, his pointers and his commands has served because the foremost contributor in the direction of the finishing touch of the mission.

Then I would love to thank my mother and father and friends who've helped me with their treasured pointers and steering has been beneficial in diverse levels of the finishing touch of the mission.

# **1.INTRODUCTION**

A library is a place where a huge collection of books and resources are available which can be accessible by the users. It acts as a brain for the institutions. It enhances the dissemination of knowledge and spiritual civilization among the students. The tons of books and research works are captivating the students to improvise their knowledge in all perspectives. It guides the students to promote their views differently. This knowledge optimizes the student to achieve a better result in academic as well as personal skill development. Improvisation in technology causes the demand for developing a way to enhance the traditional library set up to digital one. Numerous tedious processes reduce the efficiency of the library. For example, it always needs manual support to do any activities in the traditional library. The count and details of books are scribbled in the paper for reference. Each data is fetched in the notebook for future citations. To examine any data then they have to refer the notebooks. At the same time while distributing the books to the students they have to enter into the notebook where they need to represent the book id, distribution and renewal date, and student id. The admins/staff have to assign a tag for each book and provide an id for it. They have to align and arrange the books on the shelves and marked it. Missing or theft of the book builds a serious issue and confusion to the admins. While collecting the book from the students they have to verify the penalties of the books. Therefore it causes a monotonous among the staff. Consequently, it builds an uninteresting among the student due to the slow progress of the staff. To evoke the library into the technological era, we presented a system called the Library Management system (LMS). It is an automatic system that reduces the work burden of the staff/admins through a single click. It will manage, organize and oriented the library task. The LMS supports the admin to add/view/delete/update details from the library stock. Preliminarily the admin has to add student and book details into the database. After that he/she can view/delete/update those details through the Library Management system. On account of this, the user can access the library at any time. The admins can assist the data without any confusion. Each data are retrieved from the database. if he/she access any user details then it shows username, id, book details, and penalty details. They no need to write it on paper for any references. By editing the data



they can change the parameter in it. In spite of working on the manual, the admin can feel easy to handle the automatic system. It has more additional features such as admin can maintain library records, student's history of penalties and issues. It always tracks the count of the book in the library and issued book details. This causes a flexible service for admins and students. It is a user-friendly interface, so basic computer knowledge is enough to access the LMS. The system is a customizable and user-configurable one which causes it to use in different organizations. We represent the LMS with Admin module. We built the LMS in .Net Technology which is considered as the one of the upcoming technology in IT industries. By the integration of all the modules, it will be presented on the desktop of your computer.

As aforementioned the data's are stored and secured in the database. The related data are stored together and maintained properly. It allows the user to create their database as per the requirement. The database gets manipulated by the programs which provide an interface between the databases. The database management system (DBMS) receives the command from the administrator based on the instruction it changes the data in the database. This instruction may load, retrieve or modify the existing database. It is better to assign a DBMS as a centralized one which helps multiple users to access the database in a controlled manner at a different location. Based on the scheme of DBMS, the system can assign a view mode for each user like some people can see only some data and authorized one can see all the data existing in the database. It offers both logical and physical data independence. The Open database connectivity (ODBC) provides an application programming interface that allows the client-side program to call the DBMS on the server-side.

## 2.Literature Survey

A Library Management System (LMS) is a tool to help any libraries which are still using the old way to manage their library. The old way like searching for a book using manual work is hassle, fast report generation is not possible, information about issue/return of the books are not properly maintained, no central database can be created as information is not available in database. But by using the LMS, user can overcome all the problems mentioned above. This system can manage all the happenings of the library. Book transactions including book searching, availability of the book, details and appearance of the book, personal book borrowing history and etc. can be very easily handled by this system. This system is suitable for small to big libraries including medical and legal libraries, colleges, schools, universities, corporate houses and other academic resource centers. However, I would like to focus on LMS for colleges or universities.

### 2.1- RELATED WORK

Shasha et al[1], research on the library management system to upgrade the management to meet the need of the student demand. Honghai et al[2], proposed a paper where he represents the wasting of investment in CD which is attached along with the books. To save the cost of the library, he suggested cloud computing for data transfer. Bao et al [3] presented a paper on constructing the prediction model for the library. He introduced two models for predicting the process such as the co-efficient of simple determination and t-test. This analysis explains the strong relationship between lending and the number of readers. They mainly concentrate on library lending for designing the construction of the model library. Eraxiang et al [4], launched a paper where he highlighted the disadvantage of the traditional library management systems. He provided a solution for the disadvantage by utilizing struts and hibernate framework in MVC architecture. The MVC architecture is also called a multilayer tier where presentation, business, data persistence and database layer are available. These extra features improve the maintainability and reuse of the system. Zheng et al [5], introduced a paper based on

UML for the Library Management system. Due to the good application prospect of UML, the LMS is designed and model based on this concept. Case diagram and analysis diagram are drawn after the analysis of simple LMS. Hitchense et al [6], proposed a paper on flexible usage of classes. He suggested the reuse of classes for some similar conditions. Yang et al [7], explained a tedious work on the manual process of the admins. So he introduced an LMS through VB. Bretthauer et al [8] exposed information about the open-source software for libraries. He also explained the drawback of the open-source software. Brave et al [9], presented the various open-source software such as LMS, digital library software content, citation and knowledge, and journal management software, etc.. Albee et al [10], examined the staff satisfaction and attitude towards the open-source library. Singh et al [11], proposed a paper where comparison takes place between expectations and experience of the open-source library. Huang et al

[12] proposed a paper for learning the SQL database. It will provide students to test the SQL statement is working or not.

To improve the design and progress of the library management system, we represented an LMS in .Net technology which can easily access by the admins. It overcomes the drawbacks of the existing methods. The system is highly secure and provides an effective result to the users.

The rest of the paper is represented as follows. Section 3 briefly describes the architecture of the proposed system. Section 4, explain the conclusion part of the system.

### 3.Methodology

#### 3.1Block Schematic

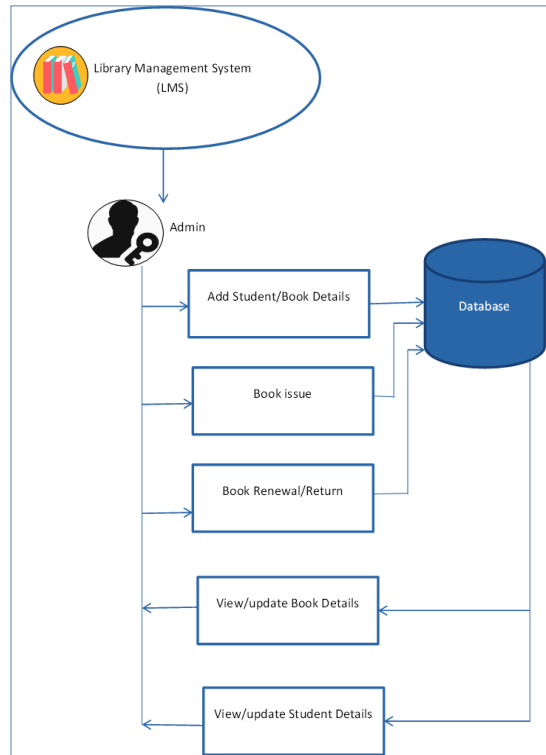


Fig : - 3.1: Block Diagram of Proposed

Fig 3.1 depicted the block diagram of the proposed Library Management system (LMS).The LMS contains an Admin module where it demonstrates the activates of the admin. Admin is considered as the authorized person to access the LMS system.

#### 3.2. welcomeMessage():

This function displays the first welcomes screen of the “Library management system project” and asks the user to press any key to access the library application.

```

#####
#####
#####   Library management System Project in C   #####
#####
#####
#####

-----
                           www.Guide by: Dr. Pallavi Jain
-----

**_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_***

-----
=                         =
=      WELCOME            =
=       TO                =
=      LIBRARY            =
=   MANAGEMENT           =
=      SYSTEM             =
=                         =
-----
**_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_*_***

```

Fig 3.2: Welcome Page of Library Management System

### 3.3. menu():

This function displays the library menu and asks the user to select the option. If the user selects 0, then the application will close.

```

#####
#####
#####   Library management System Project in C   #####
#####
#####
#####

-----
                           MAIN MENU
-----

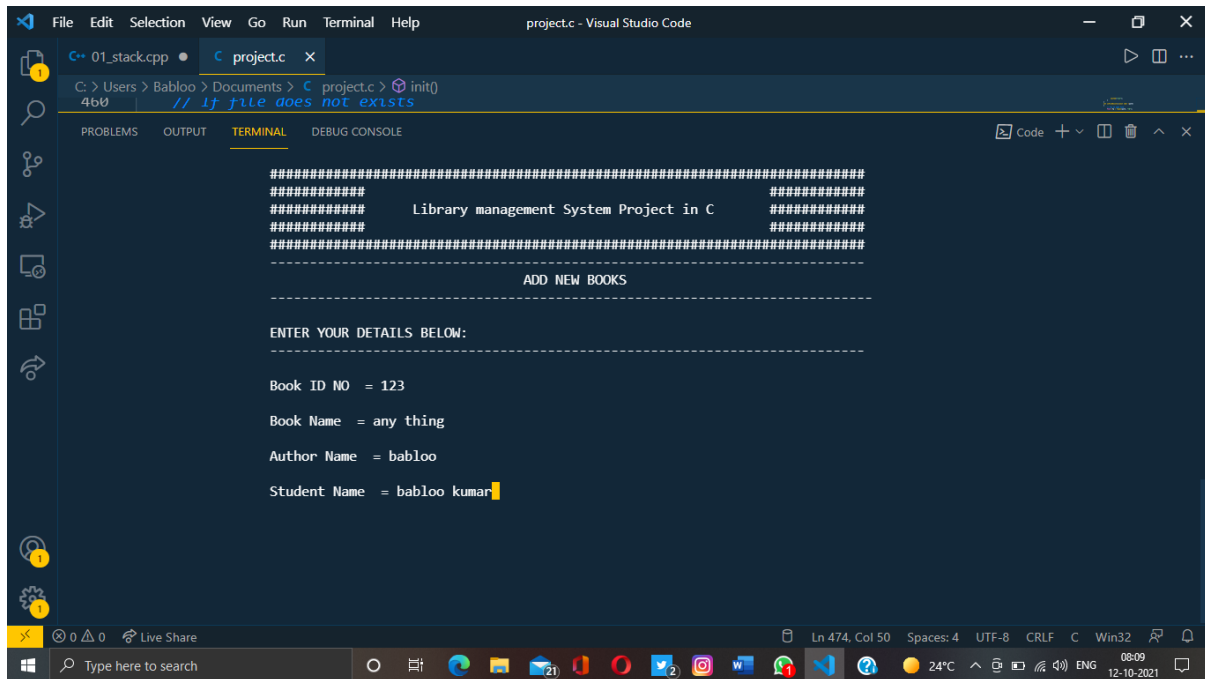
1.Add Books
2.Search Books
3.View Books
4.Delete Book
5.Update Password
0.Exit

Enter choice => █

```

### 3.4. addBookInDataBase():

This function opens the binary file in append mode and writes the book and the details.



```
File Edit Selection View Go Run Terminal Help project.c - Visual Studio Code
C:\Users> Babloo > Documents > C project.c > init()
4b0 // if file does not exists

#####
#####
##### Library management System Project in C #####
#####
#####
-----
                         ADD NEW BOOKS
-----

ENTER YOUR DETAILS BELOW:
-----

Book ID NO = 123
Book Name = any thing
Author Name = babloo
Student Name = babloo kumar
```

### 4.5. searchBooks():

This function opens the binary file in reading mode and asks the user to enter the book name which wants to search. If the book is not available in the list, it shows the message book not find in records.

```
C:\Users> Babloo > Documents > C project.c > init()
4b0 // if file does not exists

#####
#####
##### Library management System Project in C #####
#####
#####
-----
SEARCH BOOKS
-----

Enter Book Name to search:any thing

Book id = 123
Book name = any thing
Book authorName = babloo
Book issue date(day/month/year) = (12/11/2021)

Press any key to go to main menu....
```

4.6. **viewBooks()**: It opens the file in reading mode and read and display all the stored book details. If there is no book available in the records, then it displays the message record is empty.

```
C:\Users> Babloo > Documents > C project.c > isFileExists(const char *)
4b0 // if file does not exists

#####
#####
##### Library management System Project in C #####
#####
#####
-----
VIEW BOOKS DETAILS
-----

Book Count = 1

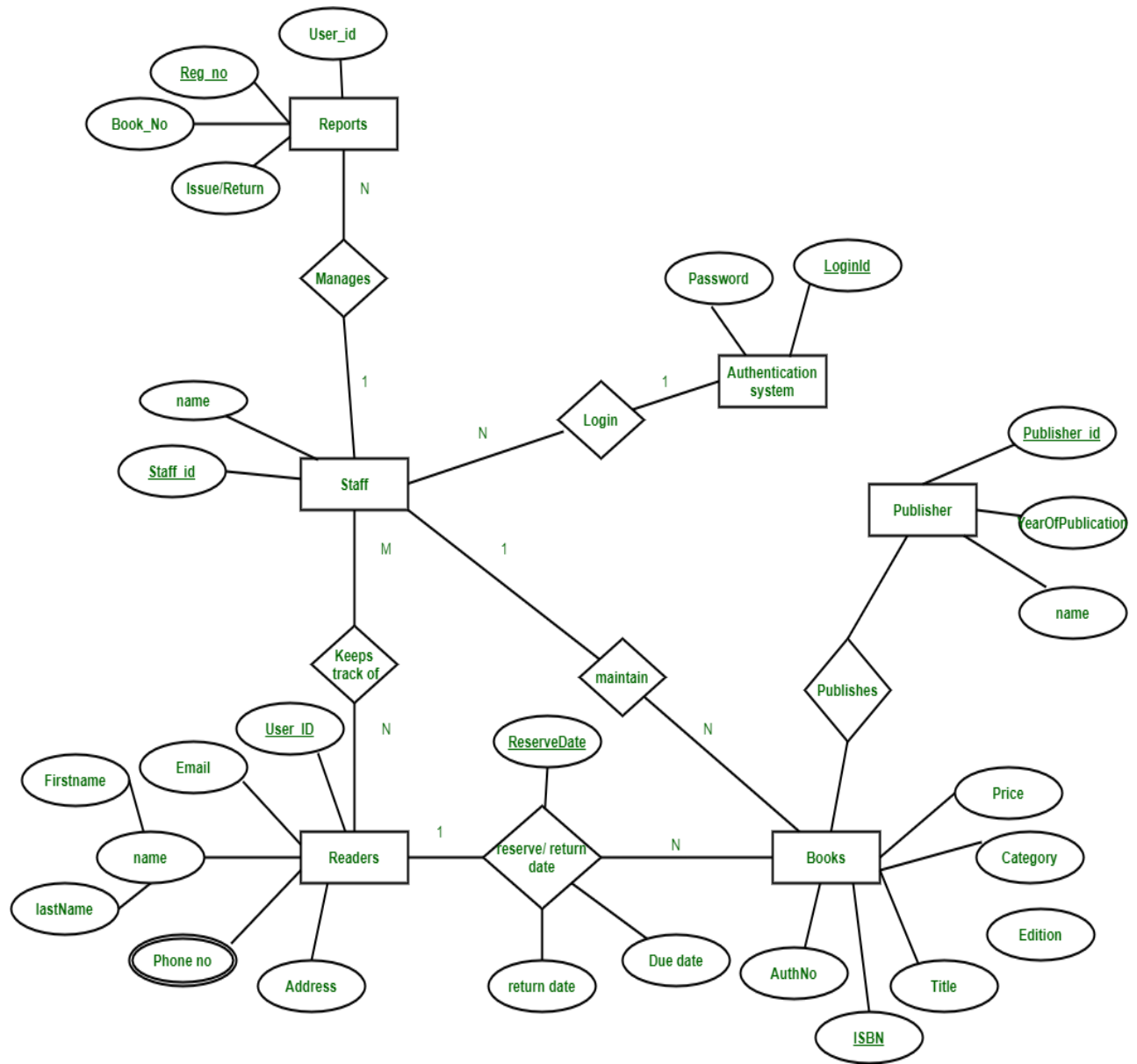
Book id = 123
Book name = any thing
Book authorName = babloo
Book issue date(day/month/year) = (12/11/2021)

Book Count = 2

Book id = 123
Book name = qwert
Book authorName = pooja
Book issue date(day/month/year) = (10/10/2020)

Book Count = 3
```

#### 4. ER diagram for library management system





## 5.CONCLUSION

The hindrance and issues of the traditional library are identified and promote it to easy access for the libraries. In the Library Management system, the admin can add/update/remove the student and book details into the database. The students have a Unique ID for accessing any book from the library. Through the ID, the admin can check the user details, fine payment, and book details. The LMS reduces labor work and makes the system efficient. In future work, we planned to enhance the LMS by integrating the LMS with Local area Network (LAN) which increases the efficiency of the system.

## 6.Complete code of Library management system project in C:

```
#include <stdio.h>
#include <time.h>6
#include <string.h>
#define MAX_YR 9999
#define MIN_YR 1900
#define MAX_SIZE_USER_NAME 30
#define MAX_SIZE_PASSWORD 20
#define FILE_NAME "LibBookS.bin"
// Macro related to the books info
#define MAX_BOOK_NAME 50
#define MAX_AUTHOR_NAME 50
#define MAX_STUDENT_NAME 50
#define MAX_STUDENT_ADDRESS 300
#define FILE_HEADER_SIZE sizeof(sFileHeader)
//structure to store date
typedef struct
{
    int yyyy;
```

```

    int mm;
    int dd;
} Date;
typedef struct
{
    char username[MAX_SIZE_USER_NAME];
    char password[MAX_SIZE_PASSWORD];
} sFileHeader;
typedef struct // to call in program
{
    unsigned int books_id; // declare the integer data type
    char bookName[MAX_BOOK_NAME]; // declare the character data type
    char authorName[MAX_AUTHOR_NAME]; // declare the character data type
    char studentName[MAX_STUDENT_NAME]; // declare the character data type
    char studentAddr[MAX_STUDENT_ADDRESS]; // declare the character data type
    Date bookIssueDate; // declare the integer data type
} s_BooksInfo;
void printMessageCenter(const char* message)
{
    int len =0;
    int pos = 0;
    //calculate how many space need to print
    len = (78 - strlen(message))/2;
    printf("\t\t\t");
    for(pos =0 ; pos < len ; pos++)
    {
        //print space
        printf(" ");
    }
    //print message
    printf("%s",message);
}
void headMessage(const char *message)
{
    system("cls");
    printf("\t\t\t#####");
    #####");
    printf("\n\t\t\t#####          #####");

```

```

printf("\n\t\t\t##### Library management System Project in C #####")
;
printf("\n\t\t\t#####");
printf("\n\t\t\t#####");
#####");
printf("\n\t\t\t-----\n");
printMessageCenter(message);
printf("\n\t\t\t-----");
}
void welcomeMessage()
{
headMessage("www.Guide by: Dr. Pallavi jain");
printf("\n\n\n\n");
printf("\n\t\t\t**_**_**_**_**_**_**_**_**_**_**_**_**_**_**_**_**_**\n");
printf("\n\t\t\t  =-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-");
printf("\n\t\t\t  =          WELCOME          =");
printf("\n\t\t\t  =          TO             =");
printf("\n\t\t\t  =          LIBRARY         =");
printf("\n\t\t\t  =          MANAGEMENT      =");
printf("\n\t\t\t  =          SYSTEM          =");
printf("\n\t\t\t  =-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-");
printf("\n\t\t\t**_**_**_**_**_**_**_**_**_**_**_**_**_**_**_**_**_**\n");
printf("\n\n\n\t\t\t\tEnter any key to continue.....");
getch();
}
int isValidName(const char *name)
{
int validName = 1;
int len = 0;
int index = 0;
len = strlen(name);
for(index =0; index <len ; ++index)
{
if(!isalpha(name[index])) && (name[index] != '\n') && (name[index] != ' ')
{
validName = 0;
break;
}
}
}
}

```

```

    return validName;
}
// Function to check leap year.
//Function returns 1 if leap year
int IsLeapYear(int year)
{
    return (((year % 4 == 0) &&
            (year % 100 != 0)) ||
            (year % 400 == 0));
}
// returns 1 if given date is valid.
int isValidDate(Date *validDate)
{
    //check range of year,month and day
    if (validDate->yyyy > MAX_YR ||
        validDate->yyyy < MIN_YR)
        return 0;
    if (validDate->mm < 1 || validDate->mm > 12)
        return 0;
    if (validDate->dd < 1 || validDate->dd > 31)
        return 0;
    //Handle feb days in leap year
    if (validDate->mm == 2)
    {
        if (IsLeapYear(validDate->yyyy))
            return (validDate->dd <= 29);
        else
            return (validDate->dd <= 28);
    }
    //handle months which has only 30 days
    if (validDate->mm == 4 || validDate->mm == 6 ||
        validDate->mm == 9 || validDate->mm == 11)
        return (validDate->dd <= 30);
    return 1;
}
// Add books in list
void addBookInDataBase()
{
    int days;

```

```

s_BooksInfo addBookInfoInDataBase = {0};
FILE *fp = NULL;
int status = 0;
fp = fopen(FILE_NAME,"ab+");
if(fp == NULL)
{
    printf("File is not opened\n");
    exit(1);
}
headMessage("ADD NEW BOOKS");
printf("\n\n\t\t\t\t\tENTER YOUR DETAILS BELOW:");
printf("\n\t\t\t\t\t-----\n");
printf("\n\t\t\t\t\tBook ID NO = ");
fflush(stdin);
scanf("%u",&addBookInfoInDataBase.books_id);
do
{
    printf("\n\t\t\t\t\tBook Name = ");
    fflush(stdin);
    fgets(addBookInfoInDataBase.bookName,MAX_BOOK_NAME,stdin);
    status = isNameValid(addBookInfoInDataBase.bookName);
    if (!status)
    {
        printf("\n\t\t\t\t\tName contain invalid character. Please enter again.");
    }
}
while(!status);
do
{
    printf("\n\t\t\t\t\tAuthor Name = ");
    fflush(stdin);
    fgets(addBookInfoInDataBase.authorName,MAX_AUTHOR_NAME,stdin);
    status = isNameValid(addBookInfoInDataBase.authorName);
    if (!status)
    {
        printf("\n\t\t\t\t\tName contain invalid character. Please enter again.");
    }
}
while(!status);

```

```

do
{
    printf("\n\t\t\tStudent Name = ");
    fflush(stdin);
    fgets(addBookInfoInDataBase.studentName,MAX_STUDENT_NAME,stdin);
    status = isNameValid(addBookInfoInDataBase.studentName);
    if (!status)
    {
        printf("\n\t\t\tName contain invalid character. Please enter again.");
    }
}
while(!status);
do
{
    //get date year,month and day from user
    printf("\n\t\t\tEnter date in format (day/month/year): ");
    scanf("%d/%d/%d",&addBookInfoInDataBase.bookIssueDate.dd,&addBookInfoInDataBas
e.bookIssueDate.mm,&addBookInfoInDataBase.bookIssueDate.yyyy);
    //check date validity
    status = isValidDate(&addBookInfoInDataBase.bookIssueDate);
    if (!status)
    {
        printf("\n\t\t\tPlease enter a valid date.\n");
    }
}
while(!status);
fwrite(&addBookInfoInDataBase,sizeof(addBookInfoInDataBase), 1, fp);
fclose(fp);
}
// search books
void searchBooks()
{
    int found = 0;
    char bookName[MAX_BOOK_NAME] = {0};
    s_BooksInfo addBookInfoInDataBase = {0};
    FILE *fp = NULL;
    int status = 0;
    fp = fopen(FILE_NAME,"rb");
    if(fp == NULL)

```

```

{
    printf("\n\t\t\tFile is not opened\n");
    exit(1);
}
headMessage("SEARCH BOOKS");
//put the control on books detail
if (fseek(fp,FILE_HEADER_SIZE,SEEK_SET) != 0)
{
    fclose(fp);
    printf("\n\t\t\tFacing issue while reading file\n");
    exit(1);
}
printf("\n\n\t\t\tEnter Book Name to search:");
fflush(stdin);
fgets(bookName,MAX_BOOK_NAME,stdin);
while (fread (&addBookInfoInDataBase, sizeof(addBookInfoInDataBase), 1, fp))
{
    if(!strcmp(addBookInfoInDataBase.bookName, bookName))
    {
        found = 1;
        break;
    }
}
if(found)
{
    printf("\n\t\t\tBook id = %u\n",addBookInfoInDataBase.books_id);
    printf("\t\t\tBook name = %s",addBookInfoInDataBase.bookName);
    printf("\t\t\tBook authorName = %s",addBookInfoInDataBase.authorName);
    printf("\t\t\tBook issue date(day/month/year) = (%d/%d/%d)",addBookInfoInDataBase.bo
kIssueDate.dd,
        addBookInfoInDataBase.bookIssueDate.mm, addBookInfoInDataBase.bookIssueDate.
yyyy);
}
else
{
    printf("\n\t\t\tNo Record");
}
fclose(fp);
printf("\n\n\n\t\t\tPress any key to go to main menu.....");

```

```

    getchar();
}
// v books function
void viewBooks()
{
    int found = 0;
    char bookName[MAX_BOOK_NAME] = {0};
    s_BooksInfo addBookInfoInDataBase = {0};
    FILE *fp = NULL;
    int status = 0;
    unsigned int countBook = 1;
    headMessage("VIEW BOOKS DETAILS");
    fp = fopen(FILE_NAME,"rb");
    if(fp == NULL)
    {
        printf("File is not opened\n");
        exit(1);
    }
    if (fseek(fp,FILE_HEADER_SIZE,SEEK_SET) != 0)
    {
        fclose(fp);
        printf("Facing issue while reading file\n");
        exit(1);
    }
    while (fread (&addBookInfoInDataBase, sizeof(addBookInfoInDataBase), 1, fp))
    {
        printf("\n\t\t\tBook Count = %d\n\n",countBook);
        printf("\t\t\tBook id = %u",addBookInfoInDataBase.books_id);
        printf("\n\t\t\tBook name = %s",addBookInfoInDataBase.bookName);
        printf("\t\t\tBook authorName = %s",addBookInfoInDataBase.authorName);
        printf("\t\t\tBook issue date(day/month/year) = (%d/%d/%d)\n\n",addBookInfoInDataBase.
bookIssueDate.dd,
        addBookInfoInDataBase.bookIssueDate.mm, addBookInfoInDataBase.bookIssueDate.
yyyy);
        found = 1;
        ++countBook;
    }
    fclose(fp);
    if(!found)

```



```

{
    printf("\n\t\t\tNo Record");
}
printf("\n\n\t\t\tPress any key to go to main menu.....");
fflush(stdin);
getchar();
}
// delete function
void deleteBooks()
{
    int found = 0;
    int bookDelete = 0;
    sFileHeader fileHeaderInfo = {0};
    char bookName[MAX_BOOK_NAME] = {0};
    s_BooksInfo addBookInfoInDataBase = {0};
    FILE *fp = NULL;
    FILE *tmpFp = NULL;
    int status = 0;
    headMessage("Delete Books Details");
    fp = fopen(FILE_NAME,"rb");
    if(fp == NULL)
    {
        printf("File is not opened\n");
        exit(1);
    }
    tmpFp = fopen("tmp.bin","wb");
    if(tmpFp == NULL)
    {
        fclose(fp);
        printf("File is not opened\n");
        exit(1);
    }
    fread (&fileHeaderInfo,FILE_HEADER_SIZE, 1, fp);
    fwrite(&fileHeaderInfo,FILE_HEADER_SIZE, 1, tmpFp);
    printf("\n\t\t\tEnter Book ID NO. for delete:");
    scanf("%d",&bookDelete);
    while (fread (&addBookInfoInDataBase, sizeof(addBookInfoInDataBase), 1, fp))
    {
        if(addBookInfoInDataBase.books_id != bookDelete)

```

```

    {
        fwrite(&addBookInfoInDataBase,sizeof(addBookInfoInDataBase), 1, tmpFp);
    }
    else
    {
        found = 1;
    }
}
(found)? printf("\n\t\tRecord deleted successfully....."):printf("\n\t\tRecord not found");
fclose(fp);
fclose(tmpFp);
remove(FILE_NAME);
rename("tmp.bin",FILE_NAME);
}
void updateCredential(void)
{
    sFileHeader fileHeaderInfo = {0};
    FILE *fp = NULL;
    unsigned char userName[MAX_SIZE_USER_NAME] = {0};
    unsigned char password[MAX_SIZE_PASSWORD] = {0};
    headMessage("Update Credential");
    fp = fopen(FILE_NAME,"rb+");
    if(fp == NULL)
    {
        printf("File is not opened\n");
        exit(1);
    }
    fread (&fileHeaderInfo,FILE_HEADER_SIZE, 1, fp);
    if (fseek(fp,0,SEEK_SET) != 0)
    {
        fclose(fp);
        printf("\n\t\tFacing issue while updating password\n");
        exit(1);
    }
    printf("\n\n\t\tNew Username:");
    fflush(stdin);
    fgets(userName,MAX_SIZE_USER_NAME,stdin);
    printf("\n\n\t\tNew Password:");
    fflush(stdin);

```

```

fgets(password,MAX_SIZE_PASSWORD,stdin);
strncpy(fileHeaderInfo.username,userName,sizeof(userName));
strncpy(fileHeaderInfo.password,password,sizeof(password));
fwrite(&fileHeaderInfo,FILE_HEADER_SIZE, 1, fp);
fclose(fp);
printf("\n\t\tYour Password has been changed successfully");
printf("\n\t\tLogin Again:");
fflush(stdin);
getchar();
exit(1);
}
void menu()
{
int choice = 0;
do
{
headMessage("MAIN MENU");
printf("\n\n\t\t1.Add Books");
printf("\n\t\t2.Search Books");
printf("\n\t\t3.View Books");
printf("\n\t\t4.Delete Book");
printf("\n\t\t5.Update Password");
printf("\n\t\t0.Exit");
printf("\n\n\t\tEnter choice => ");
scanf("%d",&choice);
switch(choice)
{
case 1:
addBookInDataBase();
break;
case 2:
searchBooks();
break;
case 3:
viewBooks();
break;
case 4:
deleteBooks();
break;

```

```

case 5:
    updateCredential();
    break;
case 0:
    printf("\n\n\n\t\t\tThank you!!!\n\n\n\n");
    exit(1);
    break;
default:
    printf("\n\n\n\t\t\tINVALID INPUT!!! Try again...");
} //Switch Ended
}
while(choice!=0); //Loop Ended
}
//login password
void login()
{
    unsigned char userName[MAX_SIZE_USER_NAME] = {0};
    unsigned char password[MAX_SIZE_PASSWORD] = {0};
    int L=0;
    sFileHeader fileHeaderInfo = {0};
    FILE *fp = NULL;
    headMessage("Login");
    fp = fopen(FILE_NAME,"rb");
    if(fp == NULL)
    {
        printf("File is not opened\n");
        exit(1);
    }
    fread (&fileHeaderInfo,FILE_HEADER_SIZE, 1, fp);
    fclose(fp);
    do
    {
        printf("\n\n\n\t\t\tUsername:");
        fgets(userName,MAX_SIZE_USER_NAME,stdin);
        printf("\n\t\t\tPassword:");
        fgets(password,MAX_SIZE_PASSWORD,stdin);
        if(!strcmp(userName,fileHeaderInfo.username) && !strcmp(password,fileHeaderInfo.password))
        {

```

```

        menu();
    }
    else
    {
        printf("\t\t\tLogin Failed Enter Again Username & Password\n\n");
        L++;
    }
}
while(L<=3);
if(L>3)
{
    headMessage("Login Failed");
    printf("\t\t\tSorry,Unknown User.");
    getch();
    system("cls");
}
}
int isFileExists(const char *path)
{
    // Try to open file
    FILE *fp = fopen(path, "rb");
    int status = 0;
    // If file does not exists
    if (fp != NULL)
    {
        status = 1;
        // File exists hence close file
        fclose(fp);
    }
    return status;
}
void init()
{
    FILE *fp = NULL;
    int status = 0;
    const char defaultUsername[] = "babloo\n";
    const char defaultPassword[] = "babl\n";
    sFileHeader fileHeaderInfo = {0};
    status = isFileExists(FILE_NAME);
}

```

```

if(!status)
{
    //create the binary file
    fp = fopen(FILE_NAME,"wb");
    if(fp != NULL)
    {
        //Copy default password
        strncpy(fileHeaderInfo.password,defaultPassword,sizeof(defaultPassword));
        strncpy(fileHeaderInfo.username,defaultUsername,sizeof(defaultUsername));
        fwrite(&fileHeaderInfo,FILE_HEADER_SIZE, 1, fp);
        fclose(fp);
    }
}
}
}
int main()
{
    init();
    welcomeMessage();
    login();
    return 0;
}

```

Source code for learn some function()

- How to use fgetc() in C?
- How to use fputc() in C?
- How to use fgets() in C?
- How to use fputs() in C?
- How to use fread() in C?
- How to use fwrite() in C?
- How to use fopen() in C?

