Library Management System

Course: BTECH CSE

Submitted By: Mehak Negi

Sap id: 590026050

Submitted To: Mr. Prashant Trivedi

University: University of Petroleum and Energy Studies

Abstract

The Library Management System is a simple and userfriendly software developed in the C programming language to help manage day-to-day activities of a library. Instead of using paper registers or notebooks, this system stores all information digitally using file handling.

The main purpose of this project is to make library operations faster, easier, and more accurate. The system provides two types of users: Admin and User. Admins can create profiles, log in, add new books to the collection, update existing book details, delete books, and view all users and their comments. Users can create their profile, log in, search for books, view all available books, and leave comments after reading a book.

All information-such as user details, admin details, book records, and comments-is stored permanently in files, so nothing is lost even after the program is closed. The project uses C structures, functions, loops, and file handling to create a modular and efficient system.

Overall, this project demonstrates how a simple consolebased program can replace manual work and improve the management of a library. It reduces human error, saves time, and provides a clear example of how programming can solve real-life problems.

Problem Definition

Libraries that use manual registers face difficulty in managing large numbers of books and users. Searching for a book in paper records takes too much time.

High chances of human errors such as wrong entry, missing data, or duplicate information.

Updating book details (edit/delete) becomes slow and complicated when done manually.

Users cannot easily check whether a book is available without asking the librarian.

No proper system to store and track user feedback or comments.

Manual methods make the library less efficient and more time-consuming.

There is no quick way to validate user login or store passwords securely in a notebook.

As the number of books grows, maintaining accurate records becomes difficult.

Libraries need a simple digital system to store, search, update, and manage information automatically.

Algorithm

1. Start the program and show the main menu with options for Admin, User, or Exit.

2. If the Admin chooses their menu, they can create a profile or log in.

3. Once the Admin logs in, they can manage the library by adding books, editing them, deleting them, viewing all users, or checking comments.

4. After finishing any task, the Admin is taken back to the Admin menu.

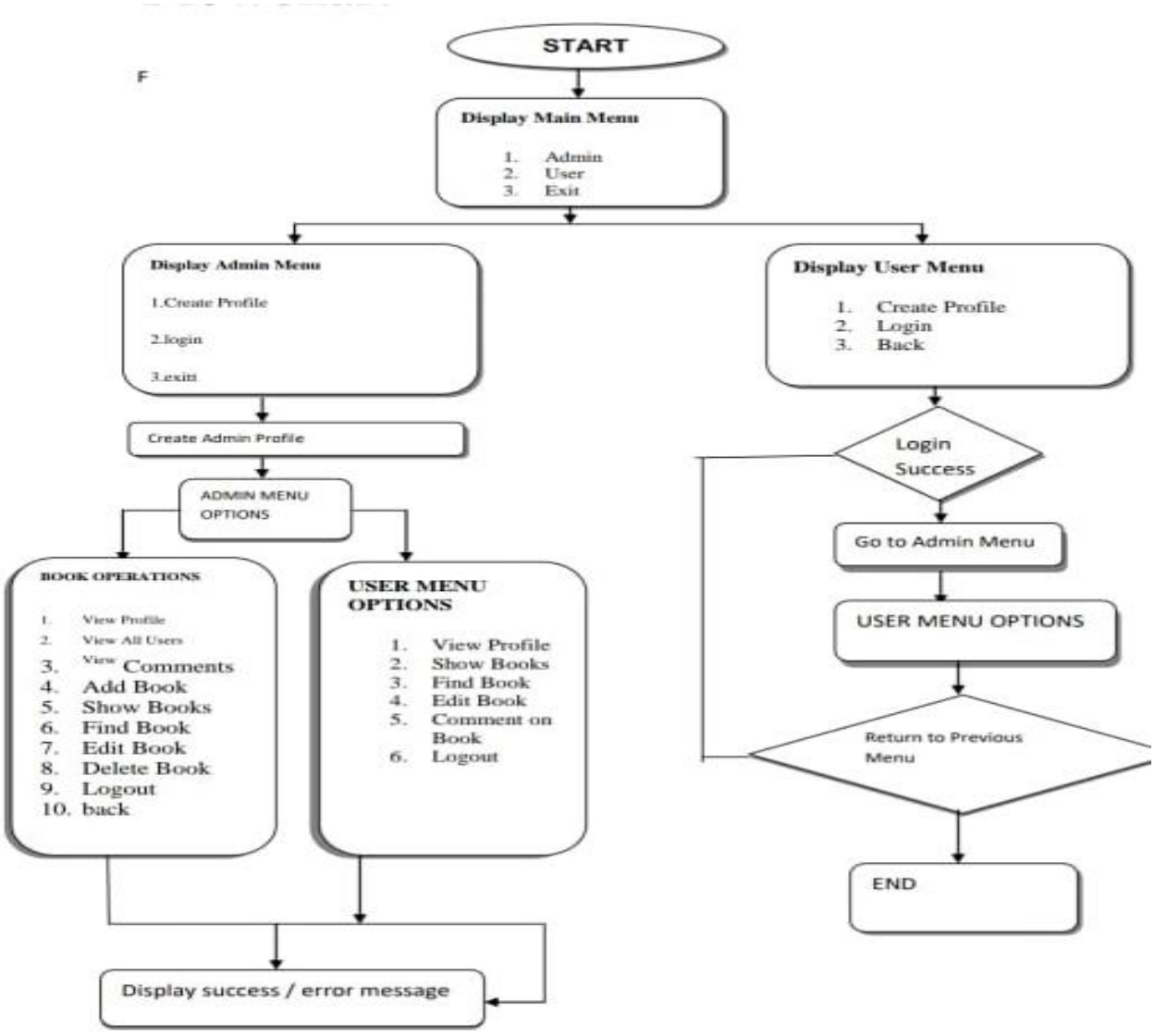
5. If the User chooses their menu, they can create a profile or log in.

6. After a User logs in, they can see their profile, search for books, view all books, or leave comments on books they have read, return and borrow books.

7. After completing any action, the User returns to the User menu. 8. If the user selects Exit from the main menu, the program closes.

9. End of the program.

Flowchart


START
Display Main Menu
1. Admin
2. User
3. Exit
Display Admin Menu
1.Create Profile
2.login
3.exitt
Create Admin Profile
ADMIN MENU OPTIONS
BOOK OPERATIONS
1. View Profile
2. View All Users
3. View Comments
4. Add Book
5. Show Books
6. Find Book
7. Edit Book
8. Delete Book
9. Logout
10. back
USER MENU OPTIONS
1. View Profile
2. Show Books
3. Find Book
4. Edit Book
5. Comment on Book
6. Logout
Display success / error message
Display User Menu
1. Create Profile
2. Login
3. Back
Login Success
Go to Admin Menu
USER MENU OPTIONS
Return to Previous Menu
END

Testing & Results

```c
void heading() {
    system ( "cls" );     //clear screen
    int d;
    printf ( "\n\n\n\t\t\t   Library System \n" );    //main menu
    printf ( "\n\t\t ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^ " );
    printf ( "\n\n\t\t    1. Admin" );
    printf ( "\n\t\t    2. User " );
    printf ( "\n\n\t\t    Enter you choice: " );
    scanf ( "%d", &d );
    if ( d == 1 ){
        admin();    }
    else if ( d == 2 ) {
        user();         }
        else {
            printf ( "\n\n\t\t \a  Wrong choice!\n\t\t    try again!" );
            fflush ( stdin );
            getchar();
            heading();
        }}
```

```
Library System

^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

1. Admin
2. User

Enter you choice:
```

```c
void admin_profile() {
    system ( "cls" );
    printf ( "\n\n\t\t     Admin Profile\n" );
    printf ( "\n\t\t ****************************** " );
    loginF2 = fopen ( "long2.txt", "rb+" );
    while ( fread ( &la, sizeof ( la ), 1, loginF2 ) == 1 ) {
        if ( pwd_ceck == la.id ) {
            printf ( "\n\n\t\tName: %s", la.name );
            printf ( "\n\t\tPassword: %s", la.password );       }   }
    printf ( "\n\n\t\t press any key.... " );
    fflush ( stdin );
    getchar();
    admin_menu();               }
```

C:\Users\mehak\Desktop\apt\project\19303001 Library_system.exe

```
Admin Profile

******************************

Name: mehak
Password: mehak1

press any key....
```

```c
// see all user function
void see_all_user() {
    int i = 1;
    system ( "cls" );
    printf ( "\n\n\t\t     All User Profile\n" );
    printf ( "\n\t\t ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^ " );
    loginF = fopen ( "long.txt", "rb+" );    //open file in read mode
    while ( fread ( &la, sizeof ( la ), 1, loginF ) == 1 ) {    /*read data from file*/
        printf ( "\n\n\t\t   Serial: %d\n", i++ );
        printf ( "\n\t\tName: %s", la.name );
        printf ( "\n\t\tDate: %s", la.date );
        printf ( "\n\t\tPassword: %s", la.password );       }
    printf ( "\n\n\t\t press any key.... " );
    fflush ( stdin );
    getchar();
    admin_menu();                       }
```

```
All User Profile

^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

Serial: 1

Name: oro
Date: 16/11/25
Password: oro3

Serial: 2

Name: aroma
Date: 21/11/2025
Password: aroma4

press any key....
```

```c
// User part bellow
void user() {
    system ( "cls" );
    int d;
    printf ( "\n\n\t\t ==========> User <========= " );
    printf ( "\n\n\t\t    1. Create Profile" );
    printf ( "\n\t\t    2. Login" );
    printf ( "\n\t\t    0. Back" );
    printf ( "\n\n\t\t    Enter you choice: " );
    scanf ( "%d", &d );
    if ( d == 0 ) {heading();}
    else if ( d == 1 ) {
        create_profile_user();  }
    else if ( d == 2 ) {
        user_login();       }
    else {
        printf ( "\n\n\t\t \a  wrong choice!\n\t\t    try agin!" );
        fflush ( stdin );
        getchar();
        user();             }}
```

```
==========> User <=========

1. Create Profile
2. Login
0. Back

Enter you choice: 1
```

```c
// user profile function
void create_profile_user() {
    system ( "cls" );
    loginF = fopen ( "long.txt", "a" );
    printf ( "\n\n\t\t ************ Create A User Profile ************" );
    printf ( "\n\n\t\t  Enter ID: " );
    scanf ( "%d", &la.id );
    fflush ( stdin );
    printf ( "\n\t\t  Enter Name: " );
    gets ( la.name );
    fflush ( stdin );
    printf ( "\n\t\t  Enter Date: " );
    fflush ( stdin );
    gets ( la.date );
    fflush ( stdin );
    printf ( "\n\t\t  Enter password: " );
    scanf ( "%s", la.password );
    fwrite ( &la, sizeof ( la ), 1, loginF );
    printf ( "\n\n\t\t Profile Create Successfully!" );
    fclose ( loginF );
    fflush ( stdin );
    getchar();
    user();          }
int check = 1;    //check for wrong attempts
```

```
************ Create A User Profile ************

 Enter ID: 3

 Enter Name: oro

 Enter Date: 16/11/25

 Enter password: oro3


Profile Create Successfully!
```

```c
// show book function
void showbook() {
    system ( "cls" );
    int count = 0;
    printf ( "\n\n\t\t\t\tAll BooKs are Available\n" );
    printf ( "\t\t\t*****************************************\n\n" );
    printf ( "\t\t\t ID\tName\tAuthor\tQunt\tRack\n\n" );
    file = fopen ( "books.dat", "rb" );
    while ( fread ( &a, sizeof ( a ), 1, file ) == 1 ) {
        printf ( "\t\t\t %d", a.id );
        printf ( "\t%s", a.name );
        printf ( "\t%s", a.author );
        printf ( "\t%d", a.quantity );
        printf ( "\t%d\n", a.rack );
        count = count + a.quantity;                }
    fclose ( file );
    printf ( "\n\t\t\t\t Total Books: %d ", count );
    fflush ( stdin );
    getchar();
    if ( t == 1 ) { admin_menu(); }
    else if ( t == 2 ) {
            user_menu();        }}
```

```
        All BooKs are Available
*****************************************

 ID     Name     Author  Qunt     Rack

 1001   NO_LONGER_HUMAN OSAMU_DAZAI     2       1
 1002   THE_SETTING_SUN OSAMU_DAZAI     2       2
 1003   TO_KILL_A_MOCKINGBIRD   HARPER_LEE      1       3
 1004   WHITE_NIGHTS    FYODOR_DOSTOYEVSKY      2       4
 1006   Pride_and_Prejudice     Jane_Austen     1       6

         Total Books: 8
```

```c
// find book function
void findbook() {
    int d, count = 0;
    system ( "cls" );
    printf ( "\n\n\t\t ************ Find Book in Library ***************\n" );
    printf ( "\n\t\t searching.....\n\n" );
    file = fopen ( "books.dat", "rb" );
    printf ( "\n\t\t Enter ID: " );
    scanf ( "%d", &d );
    while ( fread ( &a, sizeof ( a ), 1, file ) == 1 ) {
        if ( d == a.id ) {
            printf ( "\n\n\t\t Book is Found! \n" );
            printf ( "\t\t ID: %d", a.id );
            printf ( "\n\t\t Name: %s", a.name );
            printf ( "\n\t\t Author: %s", a.author );
            printf ( "\n\t\t quantity: %d", a.quantity );
            printf ( "\n\t\t Rack: %d\n", a.rack );
            count = 1;      }   }
    if ( count == 0 ) {
        printf ( "\n\n\t\t Book is Not Found ! " );     }
    printf ( "\n\n\t\t  press any key...." );
    fflush ( stdin );
    getchar();
    if ( t == 1 ) { admin_menu(); }
    else if ( t == 2 ) {
        user_menu();        }}
```

```
************ Find Book in Library ***************

searching.....


Enter ID: 1002


Book is Found!
ID: 1002
Name: THE_SETTING_SUN
Author: OSAMU_DAZAI
quantity: 2
Rack: 2


 press any key....
```

```c
void editbook() {
    system ( "cls" );
    int d, count = 0;
    printf ( "\n\n\t\t ************ edit book ************\n\n" );
    file = fopen ( "books.dat", "rb+" );
    printf ( "\n\t\t Enter ID: " );
    scanf ( "%d", &d );
    while ( fread ( &a, sizeof ( a ), 1, file ) == 1 ) {
        if ( d == a.id ) {
            printf ( "\n\t\t Book is Availble!\n" );
            printf ( "\n\n \t\t ID: %d", a.id );
            printf ( "\n\t\t Enter New Name: " );
            scanf ( "%s", a.name );
            printf ( "\n\t\t Enter New Author: " );
            scanf ( "%s", a.author );
            printf ( "\n\t\t Enter New quantity: " );
            scanf ( "%d", &a.quantity );
            printf ( "\n\t\t Enter New Rack: " );
            scanf ( "%d", &a.rack );
            fseek ( file, ftell ( file ) - sizeof ( a ), 0 );
            fwrite ( &a, sizeof ( a ), 1, file );   //writes one full structure (a) into a file
            fclose ( file );
            count = 1;      }   }
    if ( count == 0 ) {
        printf ( "\n\n\t\t Book is Not Found!" );   }
    printf ( "\n\n\t\t press any key...." );
    fflush ( stdin );
    getchar();
    admin_menu();           }
```

```
************ edit book ************

Enter ID: 1003

Book is Availble!

ID: 1003
Enter New Name: TO_KILL_A_MOCKINGBIRD

Enter New Author: HARPER_LEE

Enter New quantity: 1

Enter New Rack: 3

press any key....
```

```c
// Admin profile function
void admin_profile() {
    system ( "cls" );
    printf ( "\n\n\t\t     Admin Profile\n" );
    printf ( "\n\t\t ****************************" );
    loginF2 = fopen ( "long2.txt", "rb+" );
    while ( fread ( &la, sizeof ( la ), 1, loginF2 ) == 1 ) {
        if ( pwd_ceck == la.id ) {
            printf ( "\n\n\t\tName: %s", la.name );
            printf ( "\n\t\tPassword: %s", la.password );       }   }
    printf ( "\n\n\t\t press any key.... " );
    fflush ( stdin );
    getchar();
```

```
********** Login page for Admin **********

Enter ID: 1

Enter password: mehak1
```

```c
//ALL COMMENT function
void  see_all_comment() {
    int i = 1;
    system ( "cls" );
    printf ( "\n\n\t\t  See All comment \n" );
    printf ( "\n\t\t***************************************" );
    fcomment = fopen ( "comment.txt", "rb+" );
    while ( fread ( &c, sizeof ( c ), 1, fcomment ) == 1 ) {
        printf ( "\n\n\t\t     Comment no: %d", i++ );
        printf ( "\n\t\t User Name: %s", c.userName );
        printf ( "\n\t\t Book Name: %s", c.nameBook );
        printf ( "\n\t\t comment:\n\t\t %s\n", c.comment ); }
    fflush ( stdin );
    getchar();
    admin_menu();     }
```

```
Comment on a Book

^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

if you read a book from the Library then Make a comment.....

Enter ID: 1001


Book is Found!

Name: NO_LONGER_HUMAN
Author: OSAMU_DAZAI

Enter Your comment: The_nature_of_this_novel_is_very_dark


comment add Successfully!
```

Conclusion & Future Work

Conclusion

The Library Management System successfully demonstrates how a simple C program can automate common tasks in a library. By using file handling, the system stores book records, user profiles, admin data, and comments in an organized way without relying on manual registers. The project reduces human errors, speeds up searching, and makes the process of adding or updating books more efficient. It also ensures that users have access to important features such as viewing books, checking details quickly, and sharing their feedback. This project clearly shows the importance of computerization in daily operations and highlights how even a basic console application can improve accuracy, save time, and make the library easier to manage. Overall, the system meets its goals of providing a simple, reliable, and easy-to-use digital solution for library management.

Future Work

Although the current system works well as a basic library manager, there are several improvements that can make it more powerful and user-friendly in the future:

1. Issue and Return System Add a complete borrowing and returning module where users can issue books, check due dates, and track borrowed books.

2. Fine Calculation Introduce automatic fine calculation for overdue books to make the system more realistic and useful.

3. Database Integration Instead of using files, integrate a database like MySQL or SQLite for faster, safer, and more scalable data management.

4. Online/Cloud-Based System Convert the system into a web or cloud-based application so users can check books from anywhere.

5. Search Filters Provide advanced search options like search by author, category, year, or keywords. 6.Admin Dashboard Add statistics like number of users, total books, mostread books, and user activity reports.

7. Multi-user Access Make the system capable of handling multiple users at the same time.

References

Online C Documentation (https://www.w3schools.com / https://www.geeksforgeeks.org/)

Course Material Provided by teacher