LIBRARY MANAGEMENT SYSTEM

A PROJECT REPORT

submitted by

Lakshitha R E	312323205121
Kirubalini S I	312323205116

OF

BACHELOR OF TECHNOLOGY

IN

DEPARTMENT OF INFORMATION TECHNOLOGY

ST. JOSEPH'S COLLEGE OF ENGINEERING

(An Autonomous Institution)

St. Joseph's Group of Institutions

OMR, Chennai- 600119

TABLE OF CONTENTS

LIBRARY MANAGEMENT SYSTEM

ABSTRACT

- The Library Management System (LMS) is a software application designed to automate and streamline the processes of managing books. This system, developed using the C programming language, allows users to perform tasks such as adding new books, searching for books, deleting the books and viewing the books . The primary objective of the system is to enhance efficiency and accuracy in managing library resources. The system stores information about books (title, author, ISBN). Overall, this LMS provides a comprehensive solution to the management needs of small and medium-sized libraries, improving operations and user experience.

INTRODUCTION

- The Library Management System implemented in C provides a simple command line interface For managing a collection of books. This program allows users to perform key operations including adding new books, viewing the list of existing books, searching for books by title or author, and deleting books based on their ISBN. The data is stored in a text file, enabling persistence across program executions.

KEY FEATURES

1. Add Book: Users can input the title, author, and ISBN of a book, which is then appended to the library.txt file.

2. View Books: The system reads from the file and displays all stored books, showing their title, author, and ISBN.

3. Search Book: Users can search for books by entering a title or author name. The program checks for matches and displays relevant book information.

4. Delete Book: Users can remove a book by providing its ISBN. The program creates a temporary file to store all books except the one to be deleted, effectively removing it from the library.

IMPEMENTATION DETAILS

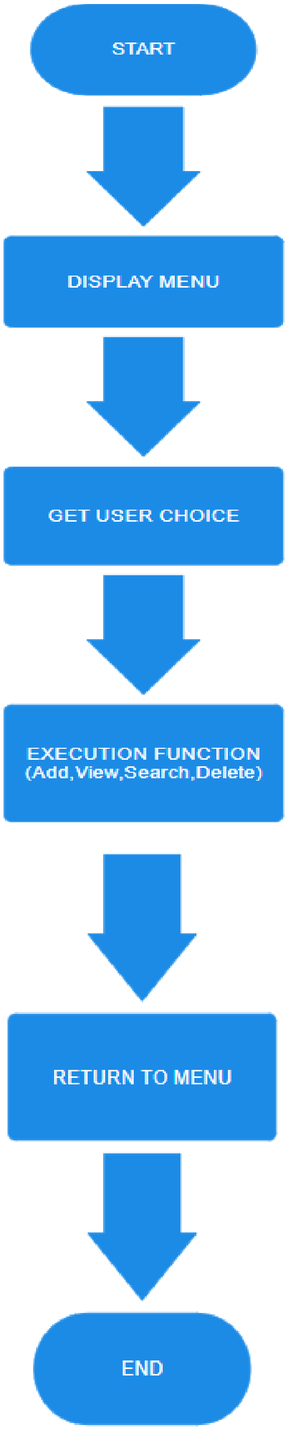
Data Structure: A Book structure holds the book's title, author, and ISBN.

File Handling: The program uses standard file operations to read from and write to a text file, ensuring that data is preserved between sessions.

User Interaction: The menu-driven interface provides a straightforward way for users to navigate through the available options.

This system serves as an educational example of file management, data structures, and basic user input handling in C, making it a useful tool for understanding fundamental programming concepts.

FLOWCHART


START
DISPLAY MENU
GET USER CHOICE
EXECUTION FUNCTION
(Add,View,Search,Delete)
RETURN TO MENU
END

LIBRARY MANAGEMENT SYSTEM CODE

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#define MAX_BOOKS 100
#define FILE_NAME "library.txt"
typedef struct {
    char title[100];
    char author[100];
    char isbn[20];
} Book;
void addBook();
void viewBooks();
void searchBook();
void deleteBook();
void saveBooks(Book *books, int count);
int main() {
    int choice;
    do {
        printf("\nLibrary Management System\n");
        printf("1. Add Book\n");
        printf("2. View Books\n");
        printf("3. Search Book\n");
        printf("4. Delete Book\n");
        printf("5. Exit\n");
        printf("Enter your choice: ");
        scanf("%d", &choice);
        switch (choice) {
            case 1: addBook(); break;
            case 2: viewBooks(); break;
            case 3: searchBook(); break;
            case 4: deleteBook(); break;
            case 5: printf("Exiting...\n"); break;
            default: printf("Invalid choice! Please try again.\n");
        }
    } while (choice != 5);
    return 0;
}
void addBook() {
    Book book;
    FILE *file = fopen(FILE_NAME, "a");
    if (!file) {
        printf("Could not open file for writing.\n");
        return;
    }
    printf("Enter book title: ");
    scanf(" %[^\n]", book.title);
    printf("Enter author: ");
    scanf(" %[^\n]", book.author);
    printf("Enter ISBN: ");
    scanf(" %[^\n]", book.isbn);
    fwrite(&book, sizeof(Book), 1, file);
    fclose(file);
    printf("Book added successfully!\n");
}
void viewBooks() {
    Book book;
    FILE *file = fopen(FILE_NAME, "r");
    if (!file) {
        printf("Could not open file for reading.\n");
        return;
    }
    printf("\nBooks in Library:\n");
    while (fread(&book, sizeof(Book), 1, file)) {
        printf("Title: %s, Author: %s, ISBN: %s\n", book.title, book.author,
book.isbn);
    }
    fclose(file);
}
void searchBook() {
    char search[100];
    Book book;
    FILE *file = fopen(FILE_NAME, "r");
    if (!file) {
        printf("Could not open file for reading.\n");
        return;
    }
    printf("Enter title or author to search: ");
    scanf(" %[^\n]", search);
    int found = 0;
    while (fread(&book, sizeof(Book), 1, file)) {
        if (strstr(book.title, search) || strstr(book.author, search)) {
            printf("Found: Title: %s, Author: %s, ISBN: %s\n", book.title, book.author, book.isbn);
            found = 1;
        }
    }
    if (!found) {
        printf("No books found matching the search.\n");
    }
    fclose(file);
}
void deleteBook() {
    char isbn[20];
    Book book;
    FILE *file = fopen(FILE_NAME, "r");
    FILE *tempFile = fopen("temp.txt", "w");
    if (!file || !tempFile) {
        printf("Could not open files for reading/writing.\n");
        return;
    }
    printf("Enter ISBN of the book to delete: ");
    scanf(" %[^\n]", isbn);
    int found = 0;
    while (fread(&book, sizeof(Book), 1, file)) {
        if (strcmp(book.isbn, isbn) != 0) {
            fwrite(&book, sizeof(Book), 1, tempFile);
        } else {
            found = 1;
        }
    }
    fclose(file);
    fclose(tempFile);
    remove(FILE_NAME);
    rename("temp.txt", FILE_NAME);
    if (found) {
        printf("Book deleted successfully!\n");
    } else {
        printf("No book found with the given ISBN.\n");
    }
}
```

CODE EXPLANATION

- **main() :**

  The entry point of the program. It presents a menu for the user to interact with the library management system. It uses a loop to repeatedly prompt the user for a choice until they decide to exit.

- **addBook() :**

1. Prompts the user for the book's title, author, and ISBN.
2. Opens the library.txt file in append mode.
3. Writes the new book's information to the file using fwrite.
4. Closes the file and confirms to the user that the book was added successfully.

- **viewBooks() :**

1. Opens the library.txt file in read mode.
2. Reads each book entry using fread in a loop.
3. Prints the title, author, and ISBN of each book.
4. Closes the file after displaying all books.

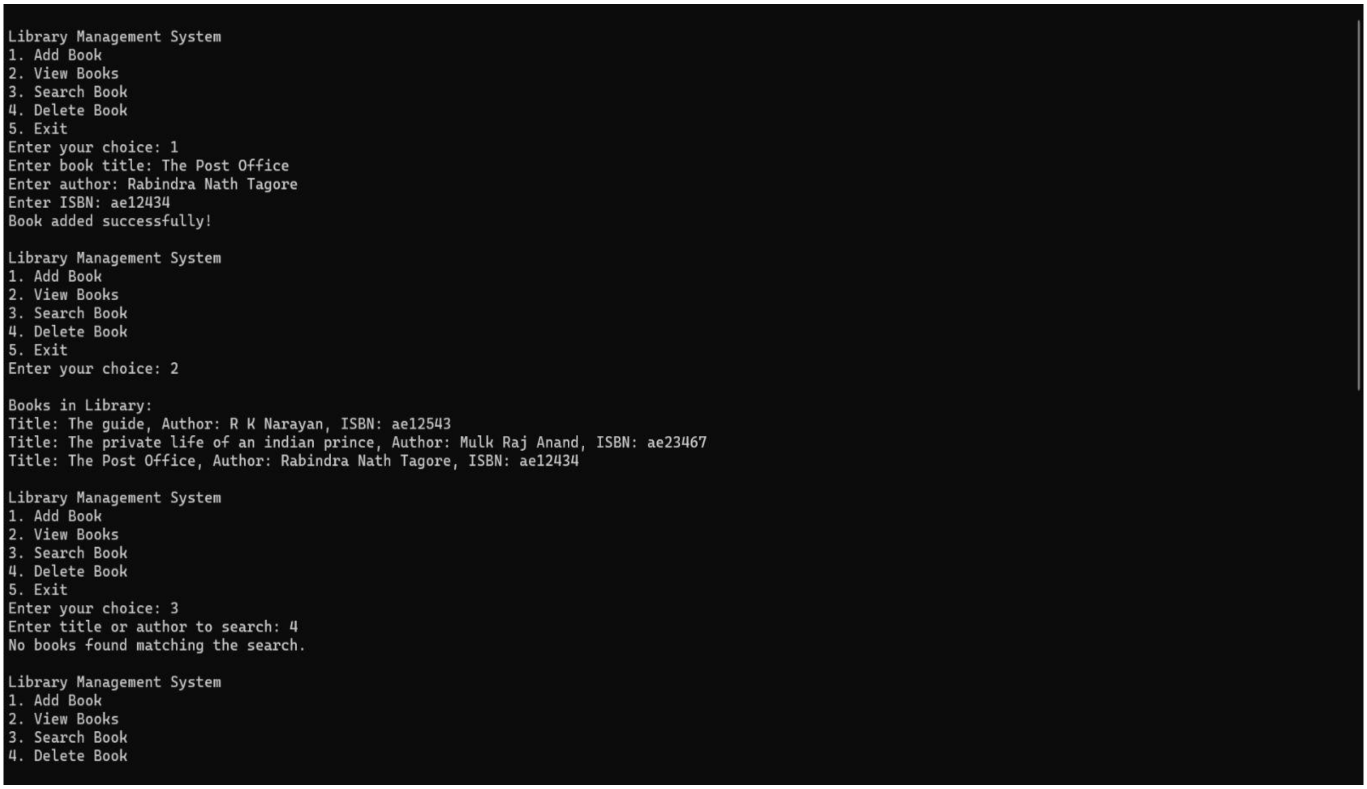
- **searchBook() :**

1. Prompts the user for a search term.
2. Opens the library.txt file in read mode.
3. Searches through each book entry using strstr to check if the title or author contains the search term.
4. Displays any matching books or informs the user if no matches are found.
5. Closes the file after completing the search.

- **deleteBook():**

1. Prompts the user for the ISBN of the book to be deleted.
2. Opens library.txt in read mode and creates a temporary file in write mode.
3. Reads each book; if the book's ISBN does not match the user’s input, it writes that book to the temporary file. If it matches, it sets a flag indicating the book was found.
4. Closes both files.
5. Replaces the original file with the temporary file.
6. Informs the user whether the book was successfully deleted or not.

OUTPUT



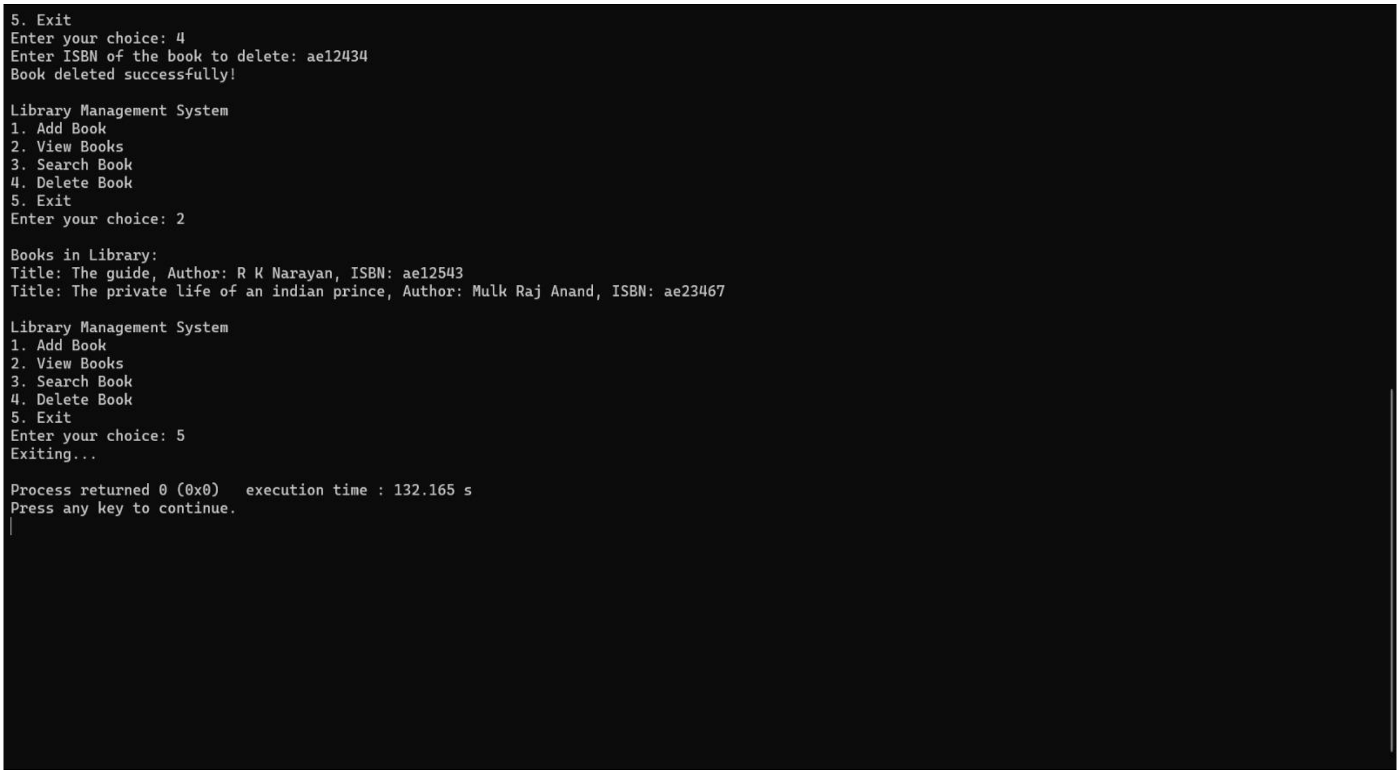
```
Library Management System
1. Add Book
2. View Books
3. Search Book
4. Delete Book
5. Exit
Enter your choice: 1
Enter book title: The Post Office
Enter author: Rabindra Nath Tagore
Enter ISBN: ae12434
Book added successfully!

Library Management System
1. Add Book
2. View Books
3. Search Book
4. Delete Book
5. Exit
Enter your choice: 2

Books in Library:
Title: The guide, Author: R K Narayan, ISBN: ae12543
Title: The private life of an indian prince, Author: Mulk Raj Anand, ISBN: ae23467
Title: The Post Office, Author: Rabindra Nath Tagore, ISBN: ae12434

Library Management System
1. Add Book
2. View Books
3. Search Book
4. Delete Book
5. Exit
Enter your choice: 3
Enter title or author to search: 4
No books found matching the search.

Library Management System
1. Add Book
2. View Books
3. Search Book
4. Delete Book
```



```
5. Exit
Enter your choice: 4
Enter ISBN of the book to delete: ae12434
Book deleted successfully!

Library Management System
1. Add Book
2. View Books
3. Search Book
4. Delete Book
5. Exit
Enter your choice: 2

Books in Library:
Title: The guide, Author: R K Narayan, ISBN: ae12543
Title: The private life of an indian prince, Author: Mulk Raj Anand, ISBN: ae23467

Library Management System
1. Add Book
2. View Books
3. Search Book
4. Delete Book
5. Exit
Enter your choice: 5
Exiting...

Process returned 0 (0x0)   execution time : 132.165 s
Press any key to continue.
```

CONCLUSION

- The Library Management System helps users manage books by allowing them to add, view, search, and delete entries. It uses a simple Book structure to keep information organized and stores records in a text file for easy access later.

- The command-line interface is user-friendly, guiding users through different options. This program demonstrates basic programming concepts in C, such as working with structures and file handling. Overall, it provides a good starting point for learning and can be improved with features like better error handling or a graphical interface.