N.B.K.R INSTITUTE OF SCIENCE AND TECHNOLOGY

Vidyanagar, Tirupathi Dist

Library Management System [Team –2]

Team Members:

1.Apshin [24KB1A05HC]

2. CH.sucharitha [24KB1A05C2]

3.ch.keerthi [24KB1A05A1]

4.Jyotshna Priya [24KB1A05D4]

course: Data Structures

Department: CSE

Date: 29 - 04 – 25

Guide: Ashok selva kumar E

Certificate

This is to certify that [Apshin (24KB1A05HC) , Sucharitha 2 (24KB1A05C2) , keerthi(24KB1A05A1)],Jyotshna Priya[24KB1A05D4] students of [CSE], Section [C], at [N.B.K.R INSTITUTE OF SCIENCE AND TECHNOLOGY], have successfully completed the project titled "Library Management System in C" under my

supervision. This project is a partial fulfillment of the curriculum of [1st B.Tech 2nd Semester].

Guide Name: Ashok selva kumar E

(Signature)

Date: 29 – 04 – 25

Acknowledgment

I would like to express my deep gratitude to [ASHOK selva kumar E], my project guide, for their valuable support, guidance, and encouragement throughout this project. I am also thankful to my family and friends who have helped me throughout the development of this project.

Abstract

This project demonstrates the creation of a basic Library Management System using the C programming language. The system allows adding, searching, issuing, returning, and deleting books. The project focuses on fundamental concepts like structures, arrays, string handling, and conditional logic.

INTRODUCTION

The Library Management System is a C-based application designed to manage library records efficiently. It allows users to add, search, issue, return, and delete books through a simple menu-driven interface. The project idea aims to replace manual book tracking with a faster, error-free digital solution. We chose this topic to apply data structures in a real-world scenario and improve our programming skills. It also provides a foundational understanding of system design and user interaction.

Objective

1. To develop a simple and functional system for managing library books using C programming.
2. To implement operations like adding, issuing, returning, and deleting books efficiently.

3. To practice structured programming concepts such as arrays, structures, and functions.
4. To ensure secure access through password-based authentication.
5. To enhance user experience through a clear, menu-driven interface.

Software & Hardware Requirements

Software:
- GCC Compiler (CodeBlocks / Turbo C / Dev C++)
- Text Editor (Notepad / VS Code)

Hardware:
- Standard PC or Laptop
- Windows/Linux Operating System

Methodology

Use of structures to define book records, arrays to store them, and functions for each feature. Password protection via simple string comparison.

Project Description

Problem Statement

Managing library books manually is inefficient and error-prone. There is a need for a simple, computerized system.

Solution

A C-based command-line system that manages library operations like adding, issuing, returning, and deleting books.

Key Features

- Secure access via password authentication

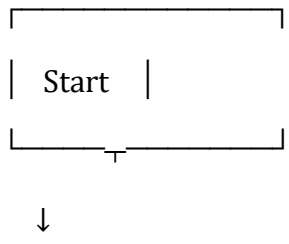
- Add/Search/Issue/Return/Delete books

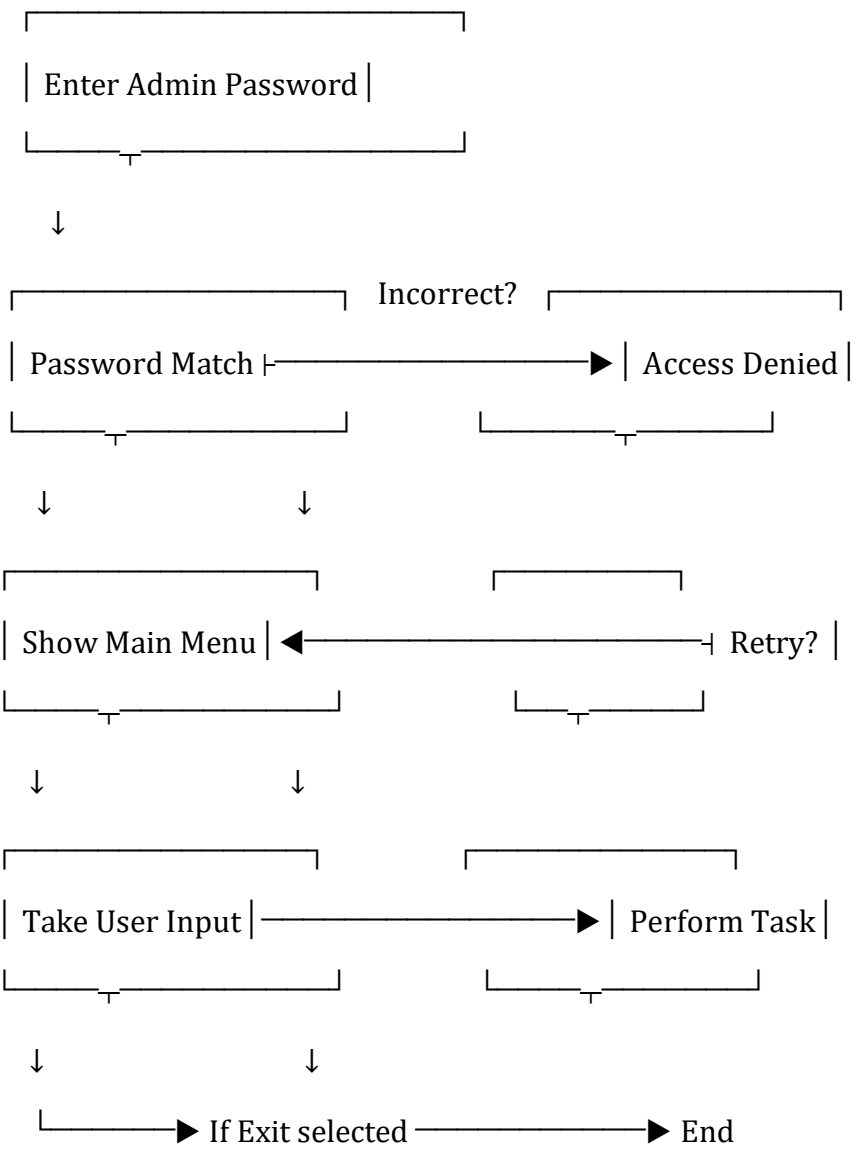
- Display all book interface

Algorithm

1. **Start**
2. Prompt for **Admin Password**
3. If password is correct:
    - Display **Main Menu**:
        1. Add Book
        2. Search Book
        3. Issue Book
        4. Return Book
        5. Delete Book
        6. Display All Books
        7. Exit
4. Take user input for menu choice.
5. Based on the choice, perform corresponding operation:
    - Use arrays and structures to manage book data.
    - For invalid input, show error and re-display menu.
6. If **Exit** is chosen, terminate the program.
7. **End**

📊 Flowchart (Text Representation)





Source code:

```
#include <stdio.h>

#include <stdlib.h>

#include <string.h>


struct Book {

   int id;
```

```
    char title[100];
    int available;  // 1 = available, 0 = issued
};

void addBook() {
    FILE *fp = fopen("library.dat", "ab");
    struct Book b;

    printf("Enter Book ID: ");
    scanf("%d", &b.id);
    printf("Enter Book Title: ");
    getchar(); // clear buffer
    fgets(b.title, 100, stdin);
    b.title[strcspn(b.title, "\n")] = 0; // remove newline
    b.available = 1;

    fwrite(&b, sizeof(b), 1, fp);
    fclose(fp);

    printf("Book added successfully.\n");
}

void displayBooks() {
    FILE *fp = fopen("library.dat", "rb");
    struct Book b;
```

```
    printf("\nAvailable Books:\n");
    printf("----------------------------\n");
    while (fread(&b, sizeof(b), 1, fp)) {
        printf("ID: %d\nTitle: %s\nStatus: %s\n\n", b.id, b.title, b.available ?
"Available" : "Issued");
    }
    fclose(fp);
}

void issueBook() {
    FILE *fp = fopen("library.dat", "rb+");
    struct Book b;
    int id, found = 0;

    printf("Enter Book to ID issue: ");
    scanf("%d", &id);

    while (fread(&b, sizeof(b), 1, fp)) {
        if (b.id == id && b.available) {
            b.available = 0;
            fseek(fp, -sizeof(b), SEEK_CUR);
            fwrite(&b, sizeof(b), 1, fp);
            found = 1;
            printf("Book issued successfully.\n");
```

```
            break;
        }
    }

    if (!found) {
        printf("Book not available or does not exist.\n");
    }

    fclose(fp);
}

void returnBook() {
    FILE *fp = fopen("library.dat", "rb+");
    struct Book b;
    int id, found = 0;

    printf("Enter Book ID to return: ");
    scanf("%d", &id);

    while (fread(&b, sizeof(b), 1, fp)) {
        if (b.id == id && !b.available) {
            b.available = 1;
            fseek(fp, -sizeof(b), SEEK_CUR);
            fwrite(&b, sizeof(b), 1, fp);
            found = 1;
```

```
            printf("Book returned successfully.\n");
            break;
        }
    }

    if (!found) {
        printf("Book not issued or does not exist.\n");
    }

    fclose(fp);
}

int main() {
    int choice;

    do {
        printf("\nLibrary Management System\n");
        printf("1. Add Book\n");
        printf("2. Display Books\n");
         printf("3. Issue Book\n");
        printf("4. Return Book\n");
        printf("5. Exit\n");
        printf("Enter choice: ");
        scanf("%d", &choice);
```

```
        switch (choice) {
            case 1: addBook(); break;
            case 2: displayBooksbreak;
            case 3: issueBook(); beaak;
            case 4: returnBook(); break;
            case 5: printf("Exiting program.\n"); break;
            default: printf("Invalid choice.\n");
        }
    } while (choice != 5);


    return 0;
}
```

Code Explanation

- Uses structures to represent book records
- Arrays store multiple book entries
- Password authentication and menu-driven functions

Testing and Validation

All functionalities such as add, search, issue, return, and delete were tested with multiple inputs. The system responded correctly and handled errors gracefully.

Output

Sample Outputs:
- Book added successfully
- Book issued

- Book returned
- Book deleted
- Invalid input handled

Limitations

- No database or file storage; data lost after program ends

- No prevention of duplicate book entries

- Simple password protection only

Conclusion

This project successfully simulates a Library Management System with basic functionalities and user interaction, providing a hands-on understanding of structured programming in C.

Future Scope

- Add file or database storage for data persistence

- Enhance security and user management

- Develop a graphical user interface (GUI)

- Prevent duplicate entries and enhance search functionality

References

- C Programming Language by E. Balagurusamy

- Online tutorials and documentation on C programming

- GCC Compiler Documentation