PROJECT REPORT
Library Management System

COURSE: BTECH CSE
SUMBITTED BY: *ADITI DONGRE*
SAP ID: *590026244*
BATCH: 38

1. Project Overview

- This project is based on managing books in a digital format using the C programming language. The idea is to replace manual register-based working and make storing, updating, and searching books easier. The system keeps book details such as book ID, title, author, and availability. It also supports issuing and returning books and updates records accordingly.
- The records are saved in a file, so the data stays even after the program is closed. The project mainly uses topics like file handling, modular code structure, conditional statements, structures, and loops.

2. **Why This System Was Made**

- **Libraries usually record books manually, which becomes difficult when the number of books increases. Some common problems include:**
- **Hard to locate a specific book quickly**
- **Mistakes in counting issued or returned books**
- **No proper backup of records**
- **Time-consuming updates**
- **So, the main purpose of the system is to:**
- **Make library tasks quicker**
- **Reduce human errors**
- **Store data digitally**
- **Retrieve information faster**

3. **Features Included**

The Library System currently supports the following:

1)Functionality

2)Adding new books

3)Viewing all stored records

4)Searching a book by its ID

5)Editing book details

6)Removing a record

7)Issuing and returning books

8)Permanent file storage

CODE :

code.c

```c
#include <stdio.h>
#include <string.h>

struct Book {
    int id;
    char title[50];
    char author[50];
    int issued;
};

struct Book library[100];
int count = 0;

void flush() {
    int c;
    while ((c = getchar()) != '\n' && c != EOF);
}

void addBook() {
    printf("\nEnter Book ID: ");
    scanf("%d", &library[count].id);
    flush();

    printf("Enter Book Title: ");
    fgets(library[count].title, 50, stdin);

    printf("Enter Author Name: ");
    fgets(library[count].author, 50, stdin);

    library[count].issued = 0;
    count++;

    printf("\nBook Added Successfully!\n");
}

void viewBooks() {
    if (count == 0) {
        printf("\nNo books available.\n");
        return;
    }

    printf("\n--- All Books ---\n");
    for (int i = 0; i < count; i++) {
        printf("\nID: %d", library[i].id);
        printf("Title: %s", library[i].title);
        printf("Author: %s", library[i].author);
        printf("Status: %s\n", library[i].issued ? "Issued" : "Available");
    }
}
```

```c
void searchBook() {
    int id;
    printf("\nEnter Book ID to Search: ");
    scanf("%d", &id);

    for (int i = 0; i < count; i++) {
        if (library[i].id == id) {
            printf("\nBook Found!\n");
            printf("Title: %s", library[i].title);
            printf("Author: %s", library[i].author);
            printf("Status: %s\n", library[i].issued ? "Issued" : "Available");
            return;
        }
    }
    printf("Book Not Found.\n");
}

void deleteBook() {
    int id;
    printf("\nEnter Book ID to Delete: ");
    scanf("%d", &id);

    for (int i = 0; i < count; i++) {
        if (library[i].id == id) {
            for (int j = i; j < count - 1; j++)
                library[j] = library[j + 1];
            count--;
            printf("Book Deleted Successfully!\n");
            return;
        }
    }
    printf("Book Not Found.\n");
}

void issueBook() {
    int id;
    printf("\nEnter Book ID to Issue: ");
    scanf("%d", &id);

    for (int i = 0; i < count; i++) {
        if (library[i].id == id) {
            if (library[i].issued == 1) {
                printf("Book already issued.\n");
                return;
            }
            library[i].issued = 1;
            printf("Book Issued Successfully!\n");
            return;
        }
    }
    printf("Book Not Found.\n");
}
```

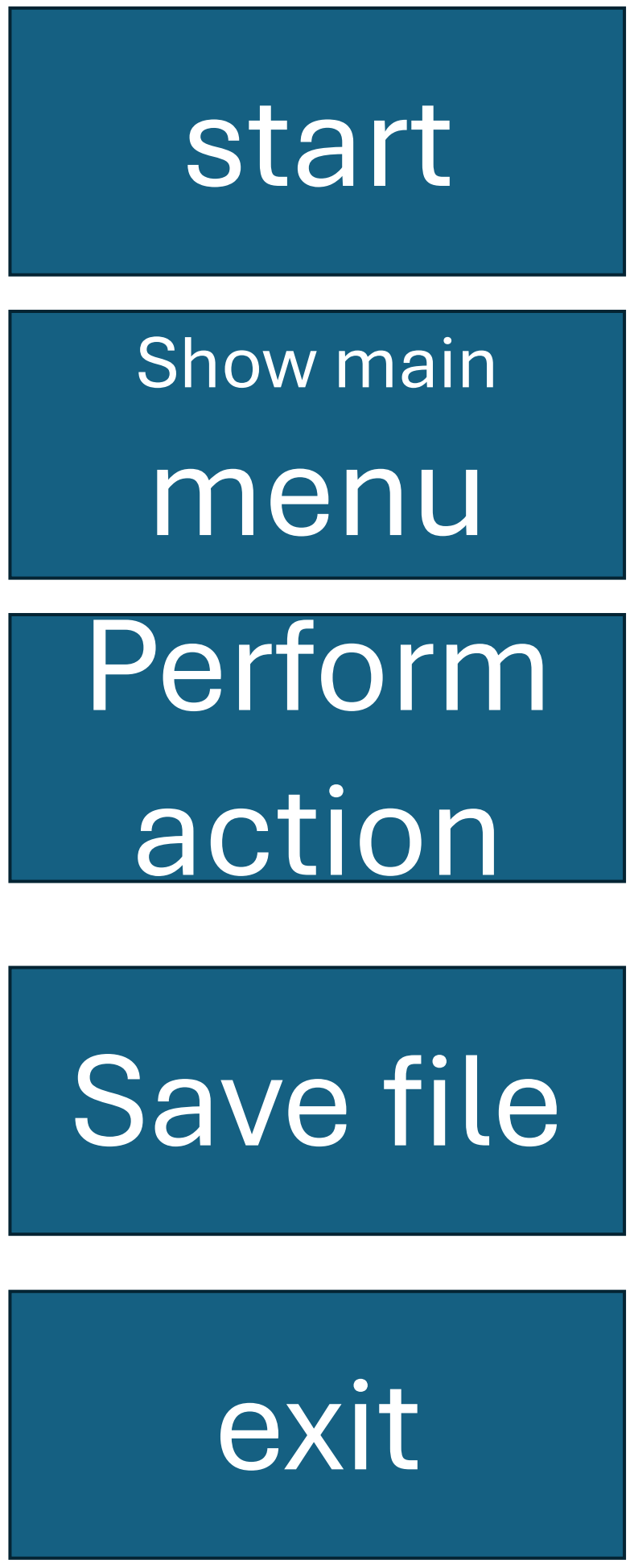
```c
void returnBook() {
    int id;
    printf("\nEnter Book ID to Return: ");
    scanf("%d", &id);

    for (int i = 0; i < count; i++) {
        if (library[i].id == id) {
            if (library[i].issued == 0) {
                printf("Book is not issued.\n");
                return;
            }
            library[i].issued = 0;
            printf("Book Returned Successfully!\n");
            return;
        }
    }
    printf("Book Not Found.\n");
}

int main() {
    int choice;

    while (1) {
        printf("\n\n=== Library 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. Issue Book\n");
        printf("6. Return Book\n");
        printf("7. Exit\n");
        printf("Enter your choice: ");

        if (scanf("%d", &choice) != 1) {
            printf("Invalid input! Enter numbers only.\n");
            flush();
            continue;
        }

        switch (choice) {
            case 1: addBook(); break;
            case 2: viewBooks(); break;
            case 3: searchBook(); break;
            case 4: deleteBook(); break;
            case 5: issueBook(); break;
            case 6: returnBook(); break;
            case 7: printf("Bye \n"); return 0;
            default: printf("Invalid Choice!\n");
        }
    }
}
```

4. **Development Approach**

- **The program is broken into multiple files instead of putting everything into one large .c file.**
  This makes the code easier to read, test, and update.

File Name	Purpose
`main.c`	The control center of the program
`menu.c`	Only displays choices
`operations.c`	Handles add, delete, search, update, issue, return functions
`database.c`	File read/write operations
`.h files`	Store structure definition and function declarations

5. **How the Program Works**



7. Algorithm :

1)Start the program

2)Create the data file if not present

3)Display the menu

4)Take user input

5)Based on the choice:

- Add / Update / Search / Delete / Issue / Return

6)Save changes to file

7)Repeat until Exit

8)Stop program

8. Implementation Summary

main.c	Control program flow
Menu.c	Shows the menu
Operations.c	Perform tasks like add/delete
Database.c	Reads/write record from file
Headers.h	Fnuction declaration and str

9. Testing

Test Case	Expected Result	Status
Add a book	Record saved	✓
Search book	Book details displayed	✓
Issue book	Available copies decrease	✓
Delete book	Record removed	✓
Duplicate ID	Show error	✓

10. Sample Output

```
=== Library Management System ===
1. Add Book
2. View Books
3. Search Book
4. Delete Book
5. Issue Book
6. Return Book
7. Exit
Enter your choice: 1

Enter Book ID: 1234
Enter Book Title: alchemist
Enter Author Name: holly

Book Added Successfully!


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

--- All Books ---

ID: 2532Title: alcemist
Author: holly
Status: Issued

ID: 1234Title: alchemist
Author: holly
Status: Available


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

Enter Book ID to Search: 1234

Book Found!
Title: alchemist
Author: holly
Status: Available



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

Enter Book ID to Issue: 1234
Book Issued Successfully!


=== Library Management System ===
1. Add Book
2. View Books
3. Search Book
4. Delete Book
5. Issue Book
6. Return Book
7. Exit
Enter your choice: 6

Enter Book ID to Return: 1234
Book Returned Successfully!


=== Library Management System ===
1. Add Book
2. View Books
3. Search Book
4. Delete Book
5. Issue Book
6. Return Book
7. Exit
Enter your choice: 7
Bye
PS F:\Aditi Dongre\programming in c\project c>
```

11. Conclusion

- **This project helped in understanding how real applications use file handling and structures in C. The system works well for small-size library operations like storing books and issuing them. The experience also improved programming logic, debugging skills, and modular coding practice.**

12. Future Enhancements

- Fine system for late returns
- User login authentication
- Sorting books alphabetically
- Backup and restore feature
- GUI/Web interface using another language

13. References

- NPTEL Programming in C
- Notes and class material
- Example programs from GeeksforGeeks