⭐ LIBRARY MANAGEMENT SYSTEM

(C Programming Project)

Submitted By:

- **Prashant Sah, 590025295**
- **Aditya Singh Bundela, 590023903**

Course:

- B.Tech

Subject:

- Programming in C

Submitted To:

- Mr. Vinod Kumar

Academic Year:

- 2024 – 2025

Department:

- Computer Science

College Name:

- University Of Petroleum And Energy Studies

Problem Definition

The project **Library Management System** is designed to store and manage book records using the C programming language.
It allows the user to:

- Add new book details
- Display all books
- Search books by ID
- Delete book records
- Issue a book (reduce quantity)

The main aim is to replace manual record-keeping with a simple computerized system using **file handling** and **dynamic memory allocation** in C.

MAIN MENU ALGORITHM

1. Start
2. Display menu
3. Read user choice
4. If input invalid → show error → repeat
5. Based on choice:
    - Call Add Book
    - Call Display Books
    - Call Search Book
    - Call Delete Book
    - Call Issue Book
    - Exit

6. Return to menu
7. Stop

ADD BOOK ALGORITHM

1. Open file in append mode
2. Input Book ID
3. Input Title (into temp)
4. Allocate memory using malloc for Title
5. Input Author (into temp)
6. Allocate memory using malloc for Author
7. Input Quantity
8. Write Book ID to file
9. Write Quantity to file
10. Write Title using writeString()
11. Write Author using writeString()
12. Close file
13. Free allocated memory
14. End

DISPLAY BOOKS ALGORITHM

1. Open file in read mode
2. While records available:
    - Read Book ID
    - Read Quantity
    - Read Title using readString()

    - Read Author using readString()
    - Display values
3. Close file
4. End

SEARCH BOOK ALGORITHM

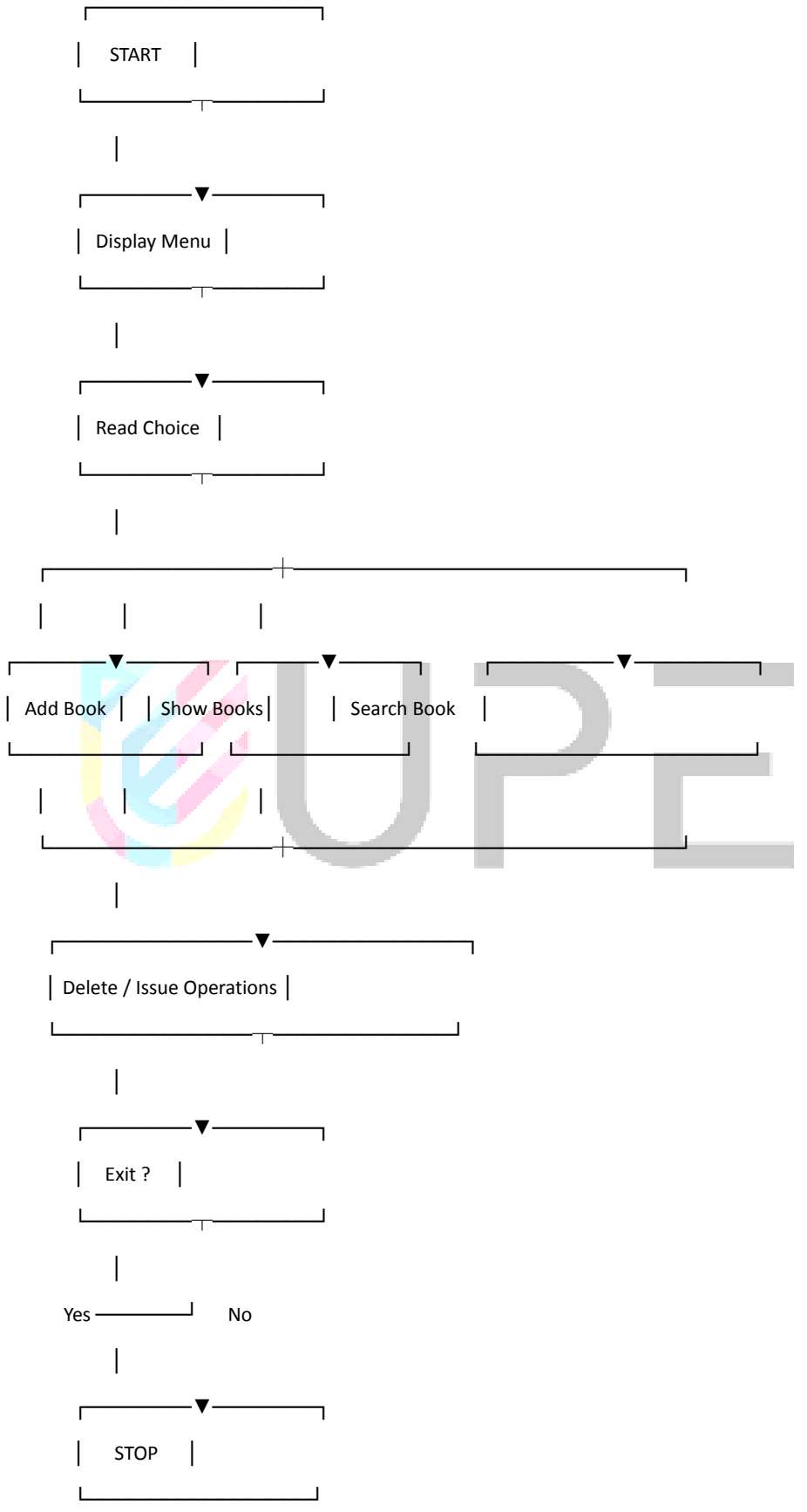
1. Input Book ID to search
2. Open file
3. Loop through all records
4. If Book ID matches:
    - Display details
    - Mark found
5. If not found → show message
6. Close file

DELETE BOOK ALGORITHM

1. Input Book ID to delete
2. Open main file (read)
3. Open temp file (write)
4. Read each book
5. If ID matches → skip writing (delete)
6. Else write to temp
7. Close files
8. Replace main file with temp
9. End

ISSUE BOOK ALGORITHM

1. Input Book ID
2. Open main file
3. Open temp file
4. For each record:
    - If ID matches & quantity > 0 → decrease quantity
    - Write updated record to temp
5. Replace main file with temp
6. End


START
Display Menu
Read Choice
Add Book
Show Books
Search Book
Delete / Issue Operations
Exit ?
Yes
No
STOP

THE CODE

Library_Managment.c > ...

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

struct Book
{
    int bookId;
    char *title;
    char *author;
    int quantity;
};

void addBook();
void showAllBooks();
void searchBook();
void deleteBook();
void issueBook();

void writeString(FILE *fp, char *str);
char *readString(FILE *fp);

int main()
{

    int choice;

    while (1)
    {

        printf("\n LIBRARY MANAGMENT SYSTEM UPES \n");
        printf("1. Add New Book\n");
        printf("2. Display All Books\n");
        printf("3. Search Book\n");
        printf("4. Delete Book\n");
        printf("5. Issue Book\n");
        printf("6. Exit\n");
        printf("Enter Choice: ");

        if (scanf("%d", &choice) != 1)
        {
            printf("Invalid Input\n");
            while (getchar() != '\n')
                ;
            continue;
        }

        switch (choice)
        {
        case 1:
            addBook();
            break;
        case 2:
            showAllBooks();
            break;
        case 3:
            searchBook();
            break;
        case 4:
            deleteBook();
            break;
        case 5:
            issueBook();
            break;
        case 6:
            exit(0);
        default:
            printf("Invalid Choice\n");
        }
    }
}

void writeString(FILE *fp, char *str)
{
```

```c
void writeString(FILE *fp, char *str)
{
    int len = strlen(str);
    fwrite(&len, sizeof(int), 1, fp);
    fwrite(str, sizeof(char), len, fp);
}

char *readString(FILE *fp)
{
    int len;
    fread(&len, sizeof(int), 1, fp);
    char *s = malloc(len + 1);
    fread(s, sizeof(char), len, fp);
    s[len] = '\0';
    return s;
}

void addBook()
{

    struct Book b;
    char temp[200];

    FILE *fp = fopen("library.dat", "ab");

    if (!fp)
    {
        printf("File Error\n");
        return;
    }

    printf("Enter Book ID: ");
    scanf("%d", &b.bookId);

    printf("Enter Title: ");
    scanf("%s", temp);
    b.title = malloc(strlen(temp) + 1);
```

```
    strcpy(b.title, temp);

    printf("Enter Author: ");
    scanf("%s", temp);
    b.author = malloc(strlen(temp) + 1);
    strcpy(b.author, temp);

    printf("Enter Quantity: ");
    scanf("%d", &b.quantity);

    fwrite(&b.bookId, sizeof(int), 1, fp);
    fwrite(&b.quantity, sizeof(int), 1, fp);
    writeString(fp, b.title);
    writeString(fp, b.author);

    fclose(fp);

    printf("Book Added Successfully\n");

    free(b.title);
    free(b.author);
}

void showAllBooks()
{

    FILE *fp = fopen("library.dat", "rb");

    if (!fp)
    {
        printf("No Book Records Found\n");
        return;
    }

    printf("\n--------- BOOK LIST ---------\n");

    int id, qty;
```

```
    while (fread(&id, sizeof(int), 1, fp))
    {

        fread(&qty, sizeof(int), 1, fp);
        char *title = readString(fp);
        char *author = readString(fp);

        printf("ID: %d\n", id);
        printf("Title: %s\n", title);
        printf("Author: %s\n", author);
        printf("Quantity: %d\n", qty);
        printf("---------------------------\n");

        free(title);
        free(author);
    }

    fclose(fp);
}

void searchBook()
{

    int searchId;
    int found = 0;

    printf("Enter Book ID: ");
    scanf("%d", &searchId);

    FILE *fp = fopen("library.dat", "rb");

    if (!fp)
    {
        printf("No Records Available\n");
        return;
    }
```

```
    int id, qty;

    while (fread(&id, sizeof(int), 1, fp))
    {

        fread(&qty, sizeof(int), 1, fp);
        char *title = readString(fp);
        char *author = readString(fp);

        if (id == searchId)
        {
            printf("\nBook Found:\n");
            printf("ID: %d\n", id);
            printf("Title: %s\n", title);
            printf("Author: %s\n", author);
            printf("Quantity: %d\n", qty);
            found = 1;
        }

        free(title);
        free(author);
    }

    if (!found)
        printf("Book Not Found\n");

    fclose(fp);
}

void deleteBook()
{

    int deleteId;
    int deleted = 0;
```

```c
printf("Enter Book ID to Delete: ");
scanf("%d", &deleteId);

FILE *fp = fopen("library.dat", "rb");
FILE *temp = fopen("temp.dat", "wb");

if (!fp || !temp)
{
    printf("File Error\n");
    return;
}

int id, qty;

while (fread(&id, sizeof(int), 1, fp))
{

    fread(&qty, sizeof(int), 1, fp);
    char *title = readString(fp);
    char *author = readString(fp);

    if (id != deleteId)
    {
        fwrite(&id, sizeof(int), 1, temp);
        fwrite(&qty, sizeof(int), 1, temp);
        writeString(temp, title);
        writeString(temp, author);
    }
    else
    {
        deleted = 1;
    }

    free(title);
    free(author);
}
```

```
    fclose(fp);
    fclose(temp);

    remove("library.dat");
    rename("temp.dat", "library.dat");

    if (deleted)
        printf("Book Deleted\n");
    else
        printf("Book Not Found\n");
}

void issueBook()
{

    int issueId;
    int issued = 0;

    printf("Enter Book ID to Issue: ");
    scanf("%d", &issueId);

    FILE *fp = fopen("library.dat", "rb");
    FILE *temp = fopen("temp.dat", "wb");

    if (!fp || !temp)
    {
        printf("File Error\n");
        return;
    }

    int id, qty;

    while (fread(&id, sizeof(int), 1, fp))
    {
```

```
    while (fread(&id, sizeof(int), 1, fp))
    {

        fread(&qty, sizeof(int), 1, fp);
        char *title = readString(fp);
        char *author = readString(fp);

        if (id == issueId && qty > 0)
        {
            qty--;
            issued = 1;
            printf("Book Issued\n");
        }

        fwrite(&id, sizeof(int), 1, temp);
        fwrite(&qty, sizeof(int), 1, temp);
        writeString(temp, title);
        writeString(temp, author);

        free(title);
        free(author);
    }

    fclose(fp);
    fclose(temp);

    remove("library.dat");
    rename("temp.dat", "library.dat");

    if (!issued)
        printf("Book Not Found or Out of Stock\n");
}
```

THE OUTPUT

```
PS D:\100 day of coding\C Project> cd "d:\100 day of coding\C Project\" ; if ($?) { gcc Library_Managment.c -o Library_Managment } ; if ($?) { .\Library_Managment }

 LIBRARY MANAGMENT SYSTEM UPES
1. Add New Book
2. Display All Books
3. Search Book
4. Delete Book
5. Issue Book
6. Exit
Enter Choice: 1
Enter Book ID: 1234
Enter Title: C Programming
Enter Author: Enter Quantity: 4
Book Added Successfully

 LIBRARY MANAGMENT SYSTEM UPES
1. Add New Book
2. Display All Books
3. Search Book
4. Delete Book
5. Issue Book
6. Exit
Enter Choice: 1
Enter Book ID: 2345
Enter Title: Advance Engineering
Enter Author: Enter Quantity: 4
Book Added Successfully

 LIBRARY MANAGMENT SYSTEM UPES
1. Add New Book
2. Display All Books
3. Search Book
4. Delete Book
5. Issue Book
6. Exit
Enter Choice: 2

--------- BOOK LIST ---------
ID: 1234
Title: HARRY
Author: POTTER
Quantity: 4
-----------------------------
```

```
 LIBRARY MANAGMENT SYSTEM UPES
1. Add New Book
2. Display All Books
3. Search Book
4. Delete Book
5. Issue Book
6. Exit
Enter Choice: 5
Enter Book ID to Issue: 2345
Book Issued

 LIBRARY MANAGMENT SYSTEM UPES
1. Add New Book
2. Display All Books
3. Search Book
4. Delete Book
5. Issue Book
6. Exit
Enter Choice: 2

--------- BOOK LIST ---------
ID: 2345
Title: Advance
Author: Engineering
Quantity: 3
-----------------------------
```

```
 LIBRARY MANAGMENT SYSTEM UPES
1. Add New Book
2. Display All Books
3. Search Book
4. Delete Book
5. Issue Book
6. Exit
Enter Choice: 3
Enter Book ID: 2345

Book Found:
ID: 2345
Title: Advance
Author: Engineering
Quantity: 4
```

```
 LIBRARY MANAGMENT SYSTEM UPES
1. Add New Book
2. Display All Books
3. Search Book
4. Delete Book
5. Issue Book
6. Exit
Enter Choice: 4
Enter Book ID to Delete: 1234
Book Deleted

 LIBRARY MANAGMENT SYSTEM UPES
1. Add New Book
2. Display All Books
3. Search Book
4. Delete Book
5. Issue Book
6. Exit
Enter Choice: 2

--------- BOOK LIST ---------
ID: 2345
Title: Advance
Author: Engineering
Quantity: 4
-----------------------------
```

PROBLEM FACED

1. Understanding File Handling:

Initially, reading and writing binary data using fwrite() and fread() was confusing.

Solution: Broke down operations into simple steps.

2. Dynamic Memory Allocation:

Using malloc() for title and author strings created issues like segmentation faults.

Solution: Always allocated strlen(temp) + 1 bytes and freed memory properly.

3. Infinite Loop with scanf:

When invalid input was given, the program went into an infinite loop.

Solution: Input buffer cleared using while(getchar()!='\n');.

4. File Corruption:

Deleting or issuing books sometimes corrupted the file.

Cause: Writing records incorrectly.

Solution: Used temp file method.

5. Understanding Binary Files:

Binary files do not display readable text, which was confusing.

Solution: Focused on reading the same structure format during retrieval.