LIBRARY MANAGEMENT SYSTEM PROJECT
X REPORT X

SCHOOL OF COMPUTER SCIENCE
COMPUTER SCIENCE ENGINEERING (CSEG1041)
(BATCH – 2025-2029)

SUBMITTED BY:
Arpit Singh
SAP ID: 590027565
ENROLLMENT NUMBER: 25010103156
SEMESTER: 1st

SUBMITTED TO:
Dr. Prashant Trivedi

2. ABSTRACT:

This Library Management System is a console-based C program designed to simplify and automate the management of books within a library. The system eliminates manual record-keeping by maintaining information such as book titles, authors, IDs, and quantities. It incorporates core concepts of C such as structures, arrays, functions, and modular programming to deliver an efficient and organized way of storing and retrieving book information.

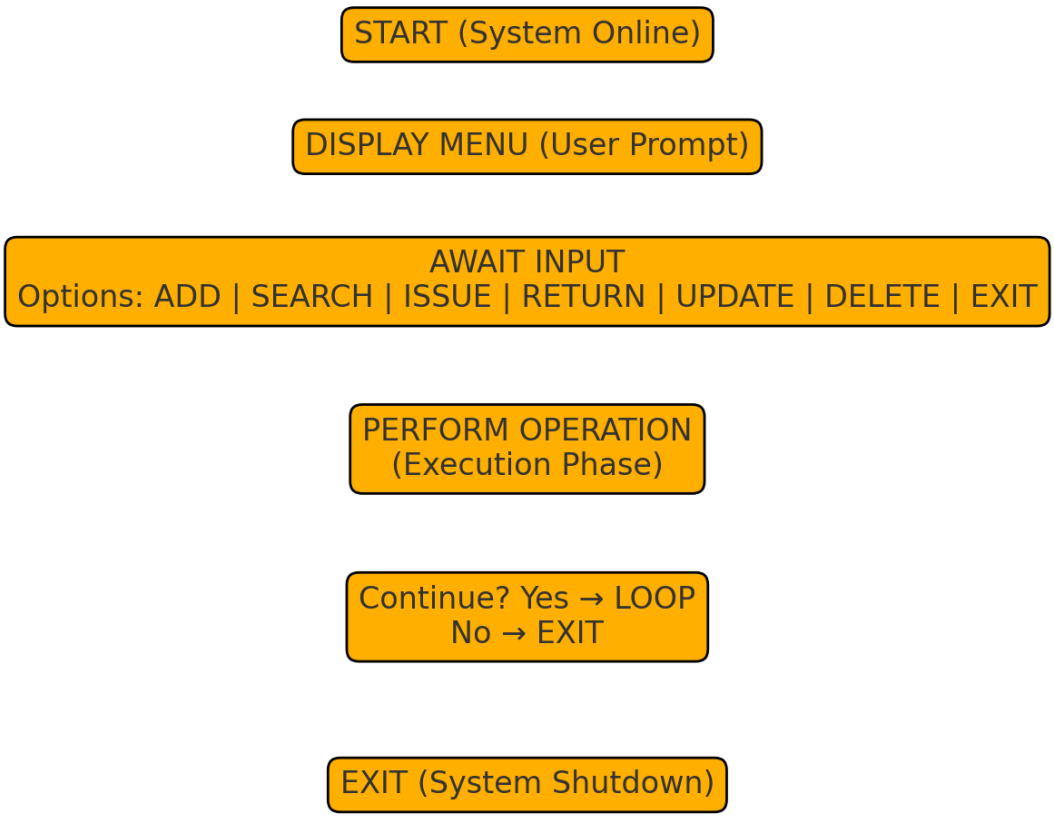
Automated functions such as adding new books, updating details, issuing and returning books, deleting records, and searching through the library database ensure operational accuracy and minimize human error. The system also prevents duplicate entries through validation checks for both book ID and title. This project demonstrates the practical application of C programming in solving real-world information management problems and forms a strong foundation for future integration of GUI, databases, and more complex features.

3. PROBLEM DEFINITION:

Manual management of library books often leads to errors, redundancy, and inefficiency. Problems include improper record-keeping, difficulty searching for books, errors in issuing/returning books, and challenges in tracking availability. This project aims to develop a digital Library Management System that automates these tasks by storing structured book details, checking availability, preventing duplicates, and enabling fast data retrieval. The system ensures accuracy, improves workflow, and supports essential library operations through a structured C-based implementation.

4. SYSTEM DESIGN:

4.1 Flowchart:



4.2 ALGORITHMS:

Algorithm: Add Book

1. Accept book ID from user.
2. Check if ID already exists.
3. Accept title; verify duplicate.
4. Accept author and quantity.
5. Insert record into library array.
6. Display confirmation.

Algorithm: Issue Book
1. Input book ID.
2. Search record.
3. If quantity > 0, decrement it.
4. Else notify user.

Algorithm: Return Book
1. Input book ID.
2. Search record.
3. Increment quantity.

Algorithm: Delete Book
1. Input ID.
2. Find record; shift array left.
3. Decrease count.

Algorithm: Search Book
1. Input ID.
2. Scan array and show details.

5. IMPLEMENTATION DETAILS:

5.1 Key Data Structure Used:

```
// Structure to store book information
struct Book {
    int id;
    char title[50];
    char author[50];
    int quantity;
};
```

5.2 Code Snippet – Adding a Book:

```c
// Function to add a new book
void addBook() {
    if (count >= MAX) {
        printf("Library is full!\n");
        return;
    }

    struct Book b;

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

    if (bookExists(b.id)) {
        printf("Book ID already exists! Choose another ID.\n");
        return;
    }

    printf("Enter Title: ");
    fgets(b.title, sizeof(b.title), stdin);
    b.title[strcspn(b.title, "\n")] = '\0';

    if (titleExists(b.title)) {
        printf("Book title already exists! Choose another title.\n");
        return;
    }

    printf("Enter Author: ");
    fgets(b.author, sizeof(b.author), stdin);
    b.author[strcspn(b.author, "\n")] = '\0';

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

    library[count] = b;
    count++;

    printf("Book added successfully!\n");
}
```

5.3 Code Snippet – Issuing a Book:

```c
// Function to issue a book
void issueBook() {
    int id, found = 0;
    printf("Enter Book ID to issue: ");
    scanf("%d", &id);

    for (int i = 0; i < count; i++) {
        if (library[i].id == id) {
            found = 1;
            if (library[i].quantity > 0) {
                library[i].quantity--;
                printf("Book issued successfully!\n");
            } else {
                printf("Book not available!\n");
            }
            break;
        }
    }
    if (!found)
        printf("Book not found!\n");
}
```

5.4 Code Snippet – Displaying a Book:

```c
// Function to display all books
void displayBooks() {
    if (count == 0) {
        printf("No books available!\n");
        return;
    }

    printf("\n%-10s %-25s %-20s %-10s\n", "ID", "Title", "Author", "Qty");
    printf("-----------------------------------------------------------------\n");
    for (int i = 0; i < count; i++) {
        printf("%-10d %-25s %-20s %-10d\n",
               library[i].id, library[i].title, library[i].author, library[i].quantity);
    }
}
```

5.5 Code Snippet – Deleting a Book:

```c
// Delete book
void deleteBook() {
    int id;
    printf("Enter 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");
}
```

5.6 Code Snippet – Searching a Book:

```c
// Function to search a book by ID
void searchBook() {
    int id, found = 0;
    printf("Enter 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\n", library[i].title);
            printf("Author: %s\n", library[i].author);
            printf("Quantity: %d\n", library[i].quantity);
            found = 1;
            break;
        }
    }
    if (!found)
        printf("Book not found!\n");
}
```

5.7 Code Snippet – Issue a Book:

```c
// Function to issue a book
void issueBook() {
    int id, found = 0;
    printf("Enter Book ID to issue: ");
    scanf("%d", &id);

    for (int i = 0; i < count; i++) {
        if (library[i].id == id) {
            found = 1;
            if (library[i].quantity > 0) {
                library[i].quantity--;
                printf("Book issued successfully!\n");
            } else {
                printf("Book not available!\n");
            }
            break;
        }
    }
    if (!found)
        printf("Book not found!\n");
}
```

5.8 Code Snippet – Updating a Book:

```c
// Update book details
void updateBook() {
    int id;
    printf("Enter Book ID to update: ");
    scanf("%d", &id);
    getchar();

    for (int i = 0; i < count; i++) {
        if (library[i].id == id) {
            printf("Enter new Title: ");
            fgets(library[i].title, sizeof(library[i].title), stdin);
            library[i].title[strcspn(library[i].title, "\n")] = '\0';

            printf("Enter new Author: ");
            fgets(library[i].author, sizeof(library[i].author), stdin);
            library[i].author[strcspn(library[i].author, "\n")] = '\0';

            printf("Enter new Quantity: ");
            scanf("%d", &library[i].quantity);

            printf("Book updated successfully!\n");
            return;
        }
    }
    printf("Book not found!\n");
}
```

6. TESTING AND RESULTS:

Test Case 1:

```
===== Library Management System =====
1. Add Book
2. Display All Books
3. Search Book
4. Issue Book
5. Return Book
6. Update Book
7. Delete Book
8. Exit
Enter your choice: 1
Enter Book ID: 101
Enter Title: C Programming
Enter Author: Dr Prashant Trivedi
Enter Quantity: 100
Book added successfully!
```

Test Case 2:

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

ID         Title                    Author               Qty
-----------------------------------------------------------------
101        C Programming            Dr Prashant Trivedi  100
```

Test Case 3:

```
===== Library Management System =====
1. Add Book
2. Display All Books
3. Search Book
4. Issue Book
5. Return Book
6. Update Book
7. Delete Book
8. Exit
Enter your choice: 4
Enter Book ID to issue: 101
Book issued successfully!
```

Test Case 4:

```
===== Library Management System =====
1. Add Book
2. Display All Books
3. Search Book
4. Issue Book
5. Return Book
6. Update Book
7. Delete Book
8. Exit
Enter your choice: 7
Enter Book ID to delete: 101
Book deleted successfully!
```

7. CONCLUSION AND FUTURE WORK:

Conclusion

The Library Management System developed in C successfully demonstrates how structured programming, modular design, and fundamental data structures can be used to automate essential library functions. Through features such as adding books, issuing and returning books, updating records, and preventing duplicate entries, the system effectively reduces manual effort and minimizes common human errors.

The program ensures accuracy through validation checks, improves efficiency by enabling quick retrieval of book details, and organizes data in a clean and manageable format. The use of structures and arrays provides a strong foundation for systematic data handling, while the menu-driven interface offers simplicity and ease of use. Overall, this project proves how C programming can be applied to real-world scenarios and serves as a stepping stone for more advanced library automation systems

Future Work

To enhance the system further, the following improvements can be implemented:

- Add file handling to save records permanently.
- Provide advanced search (by title/author) and sorting options.
- Implement user login roles (Admin/Student).
- Shift to dynamic data structures or database integration.
- Develop a graphical user interface for better usability.

8. REFERENCES:

1. Online C Language Resources (GeeksForGeeks, TutorialsPoint)
2. The C Programming Language – Kernighan & Ritchie