TITLE OF PROJECT: LABRARY MANAGEMENT SYSTEM

Group participants:

ONDAL
SHIK ROY
ARTYA DHAR
RITA CHAKROBARTYA
EYA RANI GHOSH

Table of contents

1. Introduction

The Library Management System is a console-based application developed in the C

programming language. It is designed to manage a collection of books through a linked list data structure. This program allows users to add, view, update, delete, and search for books efficiently. It demonstrates basic principles of data structures and file-less data handling in C, providing a foundational project for students .

2. Objectives

To implement a simple and efficient system for managing book records.

To practice the use of linked lists for dynamic data handling.

To develop a user-friendly interface for basic CRUD operations (Create, Read, Update, Delete).

To reinforce concepts of memory allocation and pointer manipulation in C.

To provide a prototype system that can be expanded to include file storage or database integration in the future.

3. System Architecture

System Architecture

Language: C (Procedural Programming)

Data Structure: Singly Linked List

Core Components:

Struct Definition: A Book structure containing ID, title, author, year, genre, and pointer to the next book.

Dynamic Memory: Books are dynamically allocated using malloc to allow scalable storage.

Functional Modules:

createBook(): Allocates memory and initializes a book.

addBook(): Adds a book to the end of the list.

displayBooks(): Prints all books in the list.

deleteBook(): Removes a book by ID.

searchBook(): Finds books with matching titles.

updateBook(): Modifies an existing book's details.

Main Menu Loop: Provides interactive options for users to perform actions.

7. Key Feature: Down Bidding System

•Add Book: Users can add a new book by providing all details including ID, title, author, year,

and genre.

•

•Display All Books: Lists all books currently stored in the linked list.

•

•Delete Book: Deletes a book by its ID, freeing the allocated memory.

•

•Search Book: Searches for books by title (substring match supported).

•

•Update Book: Allows modification of existing book details by ID.

•

•User-Friendly Console Interface: Clear prompts guide the user through each function.

. Future Scope

1. File Handling for Persistent Storage

Implement file input/output operations to save and load book data across sessions.

Use text or binary files to store book records permanently

2. Graphical User Interface (GUI)

Develop a GUI using libraries like GTK (for C) or migrate to higher-level languages (like Python with Tkinter) for a more user-friendly interface.

3. Database Integration

Replace the linked list with a database like SQLite or MySQL for more efficient data management and scalability.

4. Search Optimization

Improve search capabilities to include author names, genres, or year filters.

Add sorting options (by title, year, author, etc.).

5. User Authentication and Role Management

Add login systems for admin and general users.

Restrict certain operations (like deletion or updating) to authorized users.

. Conclusion

The Library Management System developed in C using linked lists serves as a practical implementation of dynamic data handling and memory management. It successfully demonstrates how a basic library can be managed through a series of operations such as adding, deleting, searching, updating, and displaying book records. By avoiding static data structures like arrays, the system becomes more flexible and scalable, handling an arbitrary number of books at runtime.

This project reinforces foundational programming concepts such as pointers, dynamic memory allocation, and modular coding practices. While the system currently runs in a console environment without file storage, it lays the groundwork for future enhancements such as persistent data storage, user authentication, or