

PROJECT REPORT

Project Title: Modular Library Book Tracking System

Course: Programming in C

Team Members: Vedansh Jain, Adarsh Chaudhary

SAP IDs: 590023901, 590025023

Date: 08 December 2025

1. Introduction

This project focuses on creating a modular Library Book Tracking System using the C programming language. The system maintains book records, supports operations such as adding, listing, searching, issuing, returning, and deleting books, and stores all data permanently using file handling.

To improve organization and maintainability, the entire codebase is divided into multiple modules such as **library.c**, **utils.c**, and respective header files.

2. Objectives

- Implement a structured and modular C application.
- Demonstrate file handling using binary files.
- Use structures to represent book data.
- Provide user-friendly menu-driven interaction.
- Enable persistent data storage across program runs.
- Ensure scalability by separating logic into header and source files.

3. Modular Structure

The project follows a clean modular architecture divided into several files:

main.c – Contains the main function and overall flow.

library.c – Handles book-related operations such as add, remove, issue, return, search, etc.

utils.c – Utility functions including ID generation and input helpers.

library.h – Structure definitions and function prototypes for library operations.

utils.h – Function prototypes for utility operations.

README – Instructions for compiling and running the project.

This modular breakdown improves readability, reusability, and debugging efficiency.

4. Features Implemented

- Add new book records
- Display complete list of books
- Search books using ID or substring
- Issue books (decrease available count)
- Return books (increase available count)
- Remove a book entry
- Store all book data in a binary file (library.dat)
- Auto-generate book IDs
- Handle up to 1000 book records

5. System Design

The system uses the following major design components:

Data Structure:

A *Book* structure containing ID, title, author, total copies, and available copies.

File Handling:

Binary file *library.dat* is used to store data persistently.

Menu System:

A loop-driven menu allowing user interaction for all operations.

6. Program Flow

1. Load data from file (if exists).
2. Display the menu.
3. Accept user choice and execute related operation.
4. Update the file after each modification.
5. Continue until user selects Exit.

7. Conclusion

This project successfully demonstrates modular programming, file handling, and structure usage in C. The system is robust, easy to extend, and shows how real-world applications such as library management systems can be built using fundamental programming concepts. Additional enhancements such as student records, due dates, and fine calculation can be integrated in the future.