Library Management System – Project Report

1. Introduction
This project is a Simple Library Management System implemented in the C programming language. It allows efficient tracking of books, enabling a user to add, search, issue, return, list, and remove books. The project features modular design by separating logic across multiple files for better readability and maintainability.

2. Objectives
• To manage library book records digitally.
• To provide efficient search and retrieval of books.
• To implement basic file handling for persistent storage.
• To demonstrate structured and modular programming in C.

3. Features
• Add new books with auto-generated ID.
• List all books with availability status.
• Search books by ID or substring in title/author.
• Issue and return books with availability validation.
• Remove outdated or unwanted book records.
• Persistent storage using binary file *library.dat*.

4. System Design
The system is divided into multiple modules:
• **main.c**: Entry point of the program.
• **library.c**: Core operations such as add, list, search, issue, return, and remove.
• **library.h**: Structure definitions and function declarations.
• **utils.c**: Helper functions like case-insensitive search and newline stripping.
• **utils.h**: Prototypes of utility functions.

5. Data Storage
The program uses a binary file **library.dat** to store:
• Total number of books.
• Details of each book (ID, title, author, total copies, available copies).
This ensures that book data persists even after the program is closed.

6. Implementation Details
Key concepts used:
• Structures for storing book data.
• File handling (fread, fwrite) for persistent storage.
• String manipulation and case-insensitive search functions.
• Modular C programming using header and source files.

7. Conclusion
This Library Management System serves as a simple yet effective example of modular programming, file handling, and record management in C. It demonstrates how real-life systems can be implemented through structured logic, efficient data storage, and clean code separation.

8. Future Enhancements
• Add user login system.
• Implement book categories.
• Include due dates and fines for issued books.

- Convert storage format to JSON or SQLite database.