PROJECT BASED LEARNING REPORT

On

LIBRARY CATALOG SYSTEM

Submitted by:
Phalguni 1/24/SET/BCS/130

Under the Guidance of:
Faculty Name: Dr. Lenin Narengbam

Department of Computer Science and Engineering
Manav Rachna International Institute of Research and Studies

Academic Year: 2025–2026

ABSTRACT

This project implements a Library Catalog System using Data Structures and Algorithms (DSA) in the C programming language. The system allows users to store book details, sort books by title or author, and perform efficient searches using Binary Search. Sorting is handled using the Merge Sort algorithm, making the search operation fast and optimized for larger datasets.
Book details such as title, author, ISBN, and publication year are stored in an array of structures. Additional helper functions handle case-insensitive comparisons and string normalization. The system provides essential functionalities such as adding new books, displaying catalog details, sorting records, and performing exact-match searches.
This project demonstrates the practical application of DSA concepts such as structures, arrays, merge sort, binary search, string handling, and menu-driven program design, making it a useful academic tool for learning structured programming.

INTRODUCTION

Libraries contain a large number of books that need to be efficiently stored, retrieved, sorted, and searched. Traditional manual management becomes slow and inefficient as the number of books grows. To handle such tasks programmatically, a Library Catalog System has been built using C and DSA techniques.

In this system, books are stored in a structure-based array, enabling fast access and manipulation. Sorting is performed using Merge Sort, known for its stability and O(n log n) efficiency. Searching is done through Binary Search, which requires prior sorting but provides O(log n) search time. Case-insensitive comparison functions allow uniform sorting and searching.

The project helps learners understand how DSA concepts can be applied in real-world data organization and management problems.

OBJECTIVES

- To design a Library Catalog System using DSA in C.
- To use structures for storing book details systematically.
- To implement Merge Sort for sorting books by title or author.
- To use Binary Search for efficient book lookup.
- To develop a clean, user-friendly, menu-driven program.
- To demonstrate string handling, file-less storage, and optimized algorithms.

SYSTEM ANALYSIS

Existing System

- Manual maintenance of book records.
- Difficulty in sorting and searching through large catalogs.
- High chances of errors and inefficiencies.

Proposed System

The C-based Library Catalog System automates book management using arrays, structures, and algorithms:

- Sorted book catalog (title/author).
- Fast searching through Binary Search.
- Easy data entry and display.
- Case-insensitive string operations for accurate comparisons.

Users

- Librarians: Add books, sort, search, and manage catalog.
- Students (optional): Quickly look up books by title/author

ER DIAGRAM AND SCHEMA DESIGN

Entities

Book

- title
- author
- isbn
- year

Catalog

- Array of Book structures
- Maximum size = 1000

IMPLEMENTATION

Component	Description
Language Used	C Programming
Data Structure	Array of struct Book
Sorting Algorithm	Merge Sort (efficient O(n log n))
Search Algorithm	Binary Search (O(log n))
String Handling	Custom lowercase conversion for accurate comparison
Core Modules	Add Book, Display Books, Sort (Title/Author), Search (Title/Author)
Menu System	Simple switch-case menu for user interaction

Key Functions Based on Code

- addBook() → Adds new book to catalog
- displayAll() → Displays all books
- mergeSort() / merge() → Sorts books by selected field
- binarySearch() → Performs fast exact-match search
- compareByTitle() & compareByAuthor() → Case-insensitive comparison
- toLowerStr() → Converts strings to lowercase

RESULTS AND DISCUSSION

- Books can be added and displayed easily.
- Sorting by both title and author works accurately due to Merge Sort.
- Searching through Binary Search returns correct results instantly.
- Case-insensitive string processing ensures consistency during sorting & searching.

CONCLUSION AND FUTURE WORK

The system effectively demonstrates how Data Structures and Algorithms in C can be used to manage library catalogs. It provides fast sorting and searching capabilities and reduces manual workload significantly.

Future Enhancements

- Add file handling to save catalog permanently.
- Implement Linked Lists or Trees for dynamic memory handling.
- Use Hashing for faster author/title lookups.
- Create a graphical or web-based interface.
- Add more search options like partial or keyword-based search.

LEARNING OUTCOME

- Understood practical implementation of DSA concepts in C.
- Learned to use structures and arrays for data management.
- Implemented Merge Sort and Binary Search in a real application.
- Improved skills in string handling and modular coding.
- Gained hands-on experience with menu-driven program design.
- Learned how algorithms improve efficiency in real-world problems.

REFERENCES

- Yashavant Kanetkar – *Let Us C*
- Reema Thareja – *Data Structures Using C*
- Mark Allen Weiss – *Data Structures and Algorithm Analysis*
- GeeksforGeeks C Programming – https://www.geeksforgeeks.org/c-programming
- C Standard Library Documentation – https://cplusplus.com/reference/clibrary/