



: R. Purushotham



 : Bytexl pvt lmtd

  : M. Pranay

 : Library management system



(R&D, QC/QA, production, Marketing, IPR etc.)



(Molecular Biology, Immunology, Diagnostics etc)

: Programming technologies

: C programming

A Library Management System (LMS) is a software application that helps libraries manage their collections, patrons, and circulation processes. The primary goal of an LMS is to provide a centralized platform for managing all aspects of a library’s operations, making it easier for librarians to perform their duties efficiently.

Online library management system is a system which maintains the information about the books present in the library, their authors the members of library to whom books are issued ,library staff and all .this is very difficult organize manually . maintenance of all this information manually is a very complex task . owing advancement of technology organization of an online library becomes much simple .

The primary goal of a Library Management System is to reduce manual work and improve the efficiency of library operations. It allows for easy book searching, borrowing, and returning while maintaining accurate records of transactions.

A Library Management System in C simplifies the operations of a library by automating book management and record-keeping. It provides an organized way to manage library resources efficiently, making it an essential tool for educational institutions and public libraries. Library Management System (LMS) in C is a program designed to handle basic operations of a library, such as adding books, issuing books, returning books, and maintaining records of students and books

**2. INTRODUCTION TO LIBRARY MANGEMENT SYSTEM**

The “Library Management System has been developed to override the problems prevailing in the practicing manual system. This software supported to eliminate and, in some cases, reduce the hardships faced by this existing system. Moreover, this system is designed for the particular need of the company to cry out operations in a sosh and effective ma

The application in reduced as much as possible to avoid errors while emoring the data It also provides, error message while entering invalid data. No formal knowledge is needed for the user to use this system. Thus, by this all it is user friendly Library Management System as described above can lead amor free, secure, reliable and fast management system. It can assist the user to concentric on ther other activities rather to concentrate on the record keeping. Tawil help organization in better utilitiken of recourses

Every organistan, whether big or small, his challenges to overcome and managing the information of Books, Students, Librarian, Address, and Member Every Library Management System has different Staden needs; therefore we deign exchaive employce management system that is adapted to your managerial requirements. This is design to assist in strategic planning, and will help you closure that your organizations equipped with the right level of information and detail for your future goal. Also, for those busy executive who are always on the po, our systems come with remote access features, which allow you to manage your workforce anytime, at all times.

Library Management System is designed to make the everyday working of the library very easy and simple. It can be done wherever we are by just using the application. It keeps track of issued, returned, reserved, requested, added books in the library.

It helps to provide all the information to the user as well as the library administrator. It provides all the facilities to the user. It has three logins that are one for the Administrator, one for the Teacher and one for the Student. By using this application we can save our time and cost.

Components of the LMS in C

To implement the LMS in C, the following concepts are used:

1. Structures (struct) – To store book and user details.

File Handling – To store data persistently.

1. Functions – To modularize code for operations like adding, searching, and issuing books.
2. Loops and Conditionals – To manage user interactions dynamically.
3. Arrays or Linked Lists – To store multiple book records.

3. SYSTEM REQUIRMENTS

HARDWARE REQUIREMENTS:

Hardware - intel i3 Speed - 2.1 GHz RAM - 4GB

Hard Disk - 100 GB SSD Floppy Drive - 2.88 MB

Key Board - Standard Windows Keyboard Mouse - Two or Three-Button Mouse

SOFTWARE REQUIREMENTS:

Operating System: Windows 10, 11 Technology: C

Compiler: byteXL TechEd editor(https://bytexl.app/editor



#include <stdio.h> #include <string.h>

struct Student { int rollNo;

char name[100]; float marks;

};

struct Student students[100]; int count = 0;

void addStudent() { if (count < 100) {

printf("Enter Roll No: ");

scanf("%d", &students[count].rollNo); printf("Enter Name: ");

scanf(" %[^\n]", students[count].name); printf("Enter Marks: ");

scanf("%f", &students[count].marks); count++;

printf("Student added successfully!\n");

} else {

printf("Record is full!\n");

}

}

void viewStudents() { if (count == 0) {

printf("No students available!\n");

} else {

printf("Students:\n");

for (int i = 0; i < count; i++) {

printf("Roll No: %d, Name: %s, Marks: %.2f\n", students[i].rollNo, students[i].name, students[i].marks);

}

}

}

void searchStudent() { int rollNo;

printf("Enter Roll No to search: "); scanf("%d", &rollNo);

for (int i = 0; i < count; i++) {

if (students[i].rollNo == rollNo) { printf("Student found!\n");

printf("Roll No: %d, Name: %s, Marks: %.2f\n", students[i].rollNo, students[i].name, students[i].marks);

return;

}

}

printf("Student not found!\n");

}

int main() { int choice;

while (1) {

printf("\nStudent Record System\n"); printf("1. Add Student\n");

printf("2. View Students\n"); printf("3. Search Student\n"); printf("4. Exit\n"); printf("Enter your choice: "); scanf("%d", &choice);

switch (choice) { case 1:

addStudent(); break;

case 2:

viewStudents(); break;

case 3:

searchStudent(); break;

case 4:

printf("Exiting...\n"); return 0;

default:

printf("Invalid choice! Try again.\n");

}

}

return 0;

}

5. CODE EXPLANANTION

This C program implements a Student Record System that allows users to add, view, and search student records using a simple menu-based interface.

Code Breakdown and Explanation

1. Structure Definition (struct Student) struct Student {

int rollNo;

char name[100]; float marks;

};

Defines a structure Student with three fields:

rollNo (integer) - Student's roll number.

name[100] (character array) - Stores the student’s name. marks (float) - Holds the student’s marks.

1. Global Variables

struct Student students[100]; int count = 0;

students[100]: An array of Student structures that can hold up to 100 student records.

count: Keeps track of the number of students currently stored.

1. Functions in the Program
   1. Adding a Student (addStudent()) void addStudent() {

if (count < 100) { printf("Enter Roll No: ");

scanf("%d", &students[count].rollNo); printf("Enter Name: ");

scanf(" %[^\n]", students[count].name); printf("Enter Marks: ");

scanf("%f", &students[count].marks);

count++;

printf("Student added successfully!\n");

} else {

printf("Record is full!\n");

}

}

Takes input for roll number, name, and marks.

Uses scanf(" %[^\n]", students[count].name); to allow multi-word names.

Increments count to add a new student.

Prevents overflow by checking if the record is full (100 students).

* 1. Viewing All Students (viewStudents()) void viewStudents() {

if (count == 0) {

printf("No students available!\n");

} else {

printf("Students:\n");

for (int i = 0; i < count; i++) {

printf("Roll No: %d, Name: %s, Marks: %.2f\n", students[i].rollNo, students[i].name, students[i].marks);

}

}

}

If no students are added, it prints "No students available!"

Otherwise, it loops through the array and prints each student’s details.

* 1. Searching for a Student (searchStudent()) void searchStudent() {

int rollNo;

printf("Enter Roll No to search: "); scanf("%d", &rollNo);

for (int i = 0; i < count; i++) {

if (students[i].rollNo == rollNo) { printf("Student found!\n");

printf("Roll No: %d, Name: %s, Marks: %.2f\n", students[i].rollNo, students[i].name, students[i].marks);

return;

}

}

printf("Student not found!\n");

}

Takes Roll Number as input and searches for it in the student array. If found, it prints the student's details.

If not, it prints "Student not found!"

1. Main Function (main()) int main() {

int choice; while (1) {

printf("\nStudent Record System\n"); printf("1. Add Student\n");

printf("2. View Students\n"); printf("3. Search Student\n"); printf("4. Exit\n"); printf("Enter your choice: "); scanf("%d", &choice);

switch (choice) { case 1:

addStudent(); break;

case 2:

viewStudents(); break;

case 3:

searchStudent(); break;

case 4:

printf("Exiting...\n"); return 0;

default:

printf("Invalid choice! Try again.\n");

}

}

return 0;

}

Implements a menu-driven system using a while(1) loop. Reads user choice and calls the corresponding function:

1. → Calls addStudent()
2. → Calls viewStudents()
3. → Calls searchStudent()
4. → Exits the program

Uses a switch statement to handle choices.

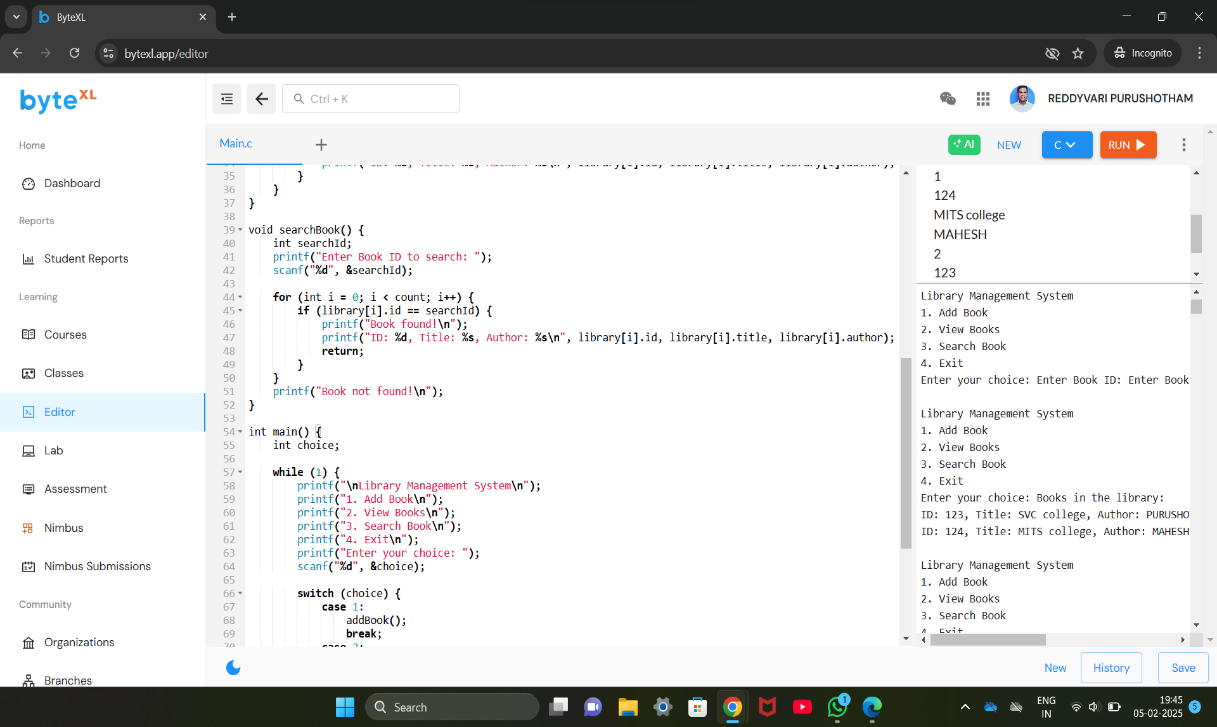
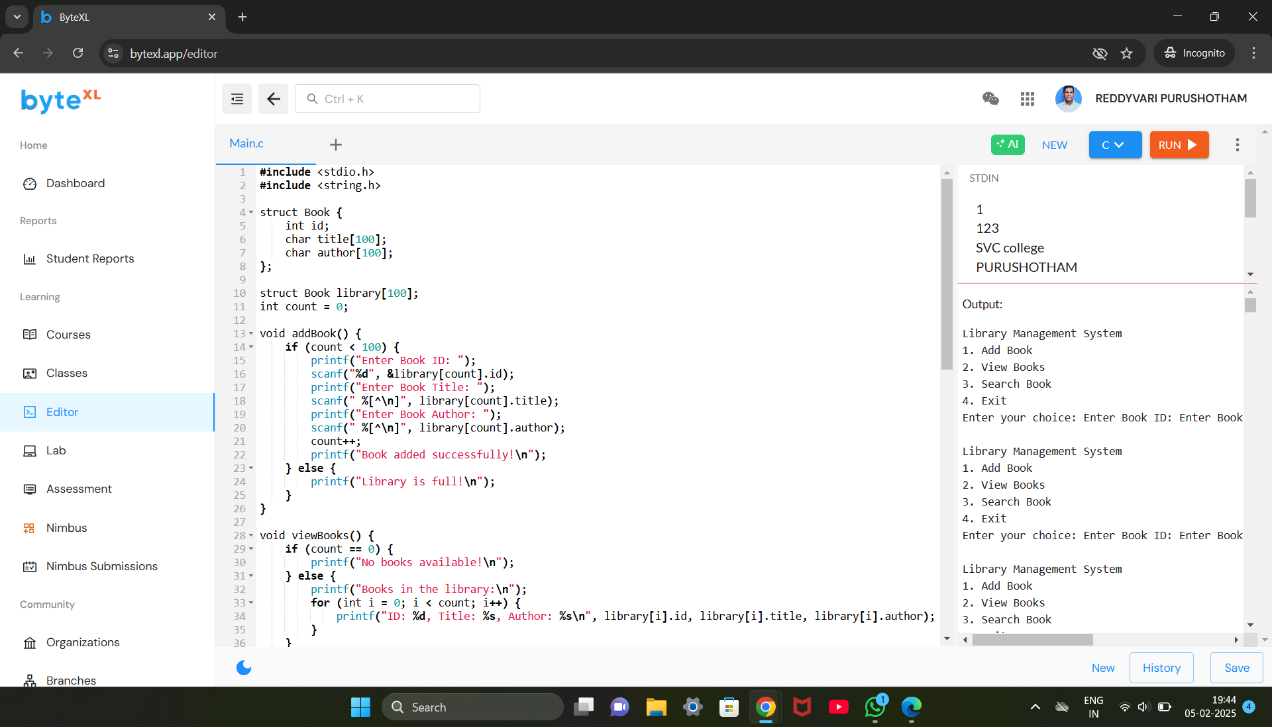
Summary

Stores student records (Roll No, Name, Marks).

Allows adding, viewing, and searching students using a menu-based system. Uses arrays and structures to organize data.

Prevents overflow by limiting records to 100 students. Uses loops for efficient searching and listing.

6. SCREEN SHOTS



7.

A Library Management System (LMS) is a software application designed to manage the operations of a library, helping to maintain a database of books, users, and transactions efficiently. The primary goal of an LMS is to facilitate smooth and organized management of library resources, including books, magazines, journals, and user data.

A Library Management System is an indispensable tool for modern libraries. It improves operational efficiency, enhances user experience, and helps librarians manage resources effectively. By automating tasks and keeping the library organized, it creates a seamless and productive environment for both users and administrators.

With advancements in technology, such systems can integrate with other digital services, such as online catalogs and e-books, further expanding the functionality and reach of libraries