

**MINI-PROJECT -REPORT**

**ON**

**LIBRARY MANAGEMENT SYSTEM**

*NAME: Ankit*

*SECTION: 3A*

*UID: 24BCA10448*

*SUBJECT: Object Oriented Programming Language*

*SUBMITTED TO: Miss. Jyoti Rani*

1. **Abstract**

The **Library Management System** is a simple console-based application developed in **C++** to automate basic library operations such as adding, deleting, issuing, and returning books.  
The project demonstrates the principles of **Object-Oriented Programming (OOP)** including classes, encapsulation, and data abstraction.  
It provides an easy-to-use menu interface for both **administrators** and **students**.

**2. Objectives**

 To design a basic system that can handle small-scale library operations.

 To implement **C++ OOP concepts** such as classes, objects, and member functions.

 To practice structured programming with control statements and loops.

 To provide a simple, text-based user interface for admin and students

1. **System Requirements**

**Software Requirements:**

* C++ Compiler (e.g., GCC, MinGW)
* Any IDE such as Code::Blocks, Turbo C++, Replit, or OnlineGDB.

**Hardware Requirements:**

* Minimum 2 GB RAM
* Any processor supporting a C++ compiler
* Basic console interface (no graphics required)

1. **Methodology**

The project is implemented using a single class (Library) that handles all book-related operations.

**Class Design**

**Class Name: Library**

**Data Members:**

* string books[100] – stores book names
* int copies[100] – total number of copies of each book
* int issued[100] – number of issued copies
* int count – number of books in the library

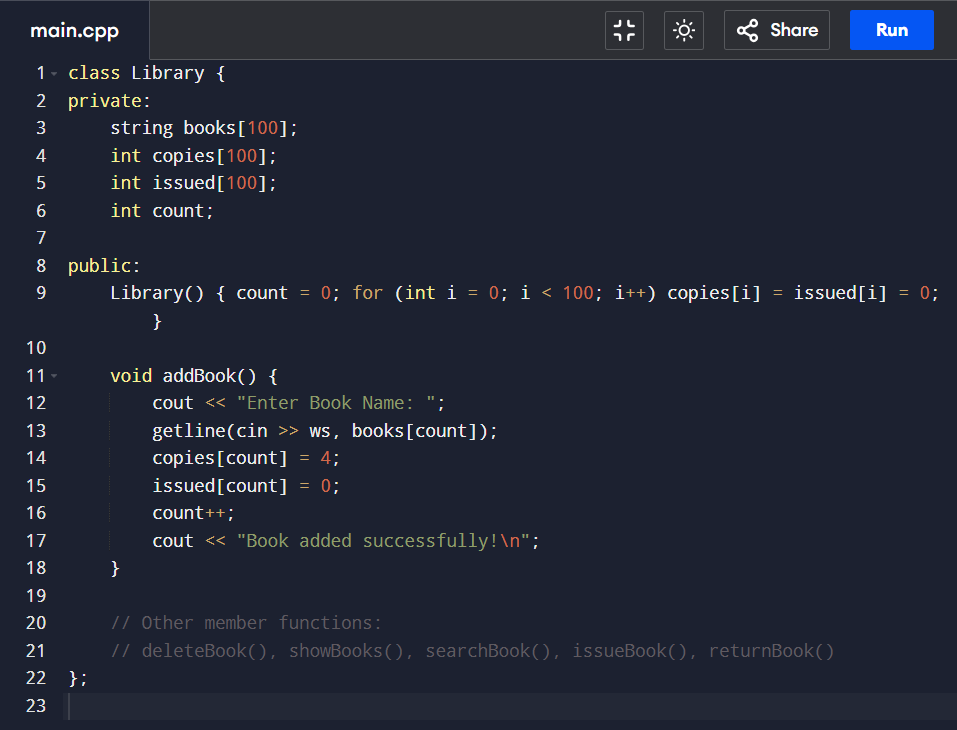
**Member Functions:**

1. addBook() – Adds a new book record
2. deleteBook() – Deletes an existing book
3. showBooks() – Displays all books
4. searchBook() – Searches for a book by name
5. issueBook() – Issues a book to a student
6. returnBook() – Returns an issued book

**6. Code Overview**

**Key Highlights:**

* Uses class-based structure for encapsulation
* Implements loops and conditionals for menu navigation
* Uses string arrays for managing book records
* No external file storage (memory-based only)



1. **Results**

* The system successfully handles book management tasks.
* Admins can add, delete, and view books.
* Students can search, issue, and return books.
* Program runs smoothly on online C++ compilers without additional dependencies.

**8. Advantages**

* Easy to use and maintain.
* Simple text-based interface for fast operation.
* Demonstrates OOP concepts effectively.
* Compact and works without external databases.

**9. Limitations**

* Data is **not saved permanently** (lost after program exit).
* No authentication for admin/student.
* Limited to 100 books.

**10. Future Enhancements**

* Add **file handling** for data persistence.
* Implement **login system** for admin and students.
* Use **linked lists or vectors** for dynamic data storage.
* Create a **GUI-based version** using Qt or JavaFX.

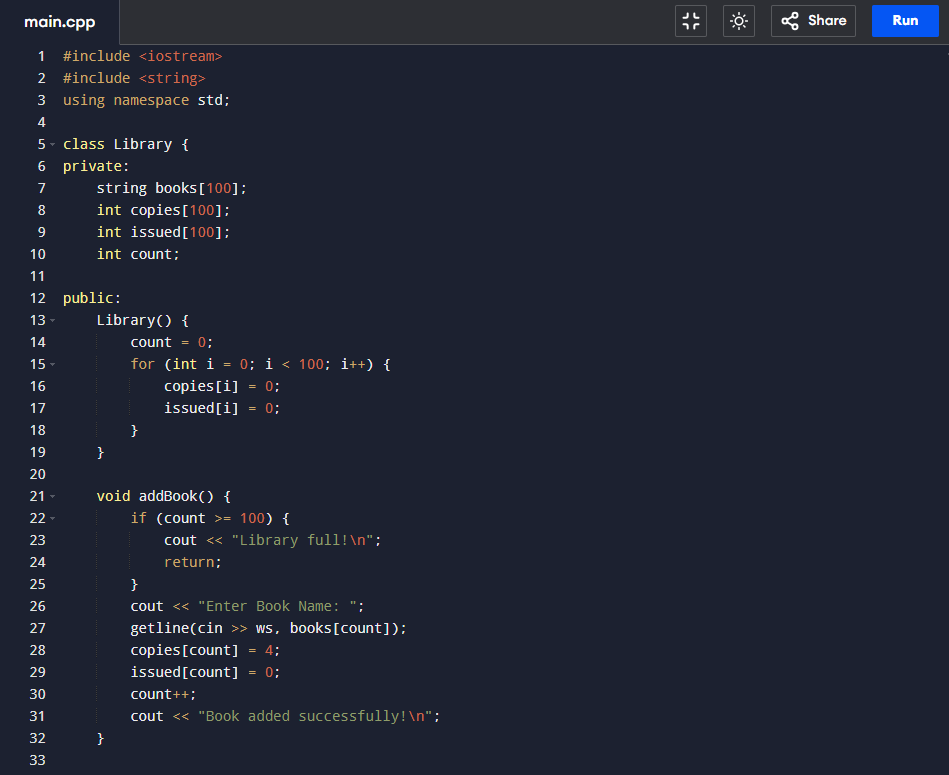
**11. Conclusion**

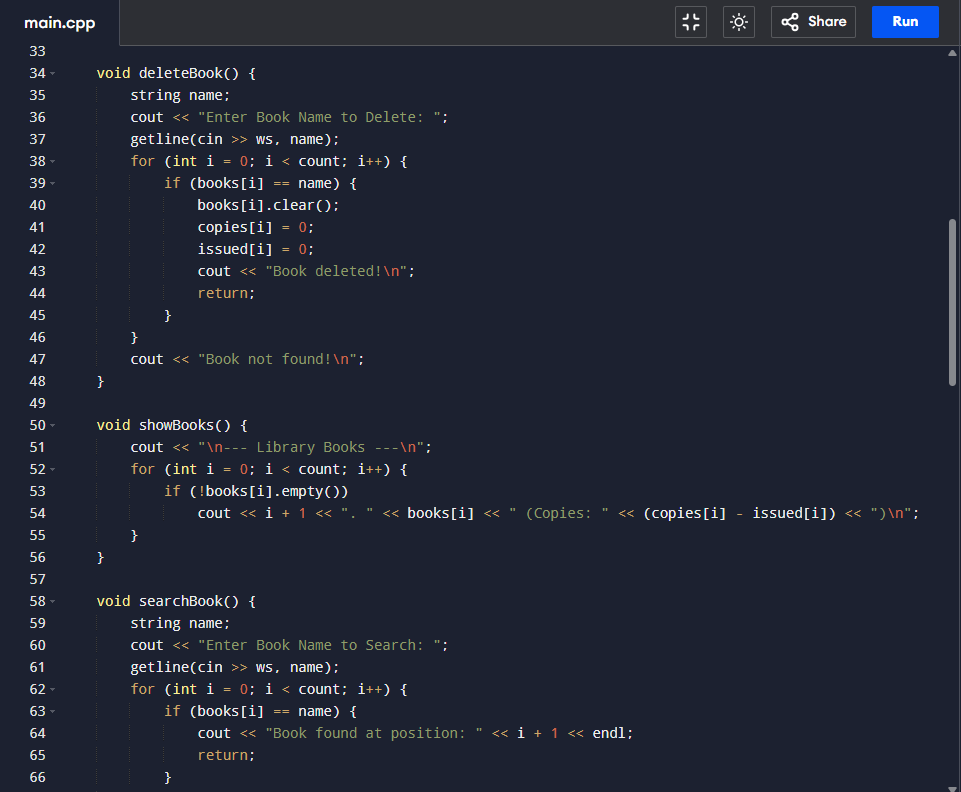
This project demonstrates the effective use of **C++ Object-Oriented Programming** to create a functional **Library Management System**.  
It successfully automates basic operations and provides a strong foundation for extending the system into a more advanced application with database integration and GUI.

**12. References**

1. E. Balagurusamy – *Object-Oriented Programming with C++*
2. Bjarne Stroustrup – *The C++ Programming Language*
3. TutorialsPoint – https://www.tutorialspoint.com/cplusplus
4. GeeksforGeeks – <https://www.geeksforgeeks.org/c-plus-plus/>

**13.Screenshot or Result**





A screenshot of a computer program

AI-generated content may be incorrect.

A screenshot of a computer program

AI-generated content may be incorrect.

A screen shot of a computer

AI-generated content may be incorrect.

