Abhishek Kumar
Reg no-12219287

Project: Simple Library Management System

https://replit.com/@abhishekjunior1/Simple-Library-Management-System

```
#include <iostream>
#include <vector>
#include <algorithm>
#include <string>
using namespace std;

class Book {
public:
    int id;
    string title;
    string author;
    bool isIssued;

    Book(int id, string title, string author)
        : id(id), title(title), author(author),
isIssued(false) {}

    void displayBook() const {
        cout << "ID: " << id << "\nTitle: " <<
title
             << "\nAuthor: " << author
             << "\nStatus: " << (isIssued ?
"Issued" : "Available") << "\n\n";
    }
};
```

```
class Library {
private:
    vector<Book> books;

    int findBookIndexById(int id) {
        for (size_t i = 0; i <
books.size(); ++i) {
            if (books[i].id == id) {
                return i;
            }
        }
        return -1;
    }

public:
    void addBook(int id, const string&
title, const string& author) {
        if (findBookIndexById(id) == -1) {
            books.push_back(Book(id, title,
author));
            cout << "Book added
successfully.\n";
        } else {
            cout << "A book with this ID
```

```
 void searchBookById(int id) {
        int index = findBookIndexById(id);
        if (index != -1) {
            books[index].displayBook();
        } else {
            cout << "Book not found.\n";
        }
    }

    void searchBookByTitle(const string&
title) const {
        bool found = false;
        for (const auto& book : books) {
            if (book.title == title) {
                book.displayBook();
                found = true;
                break;
            }
        }
        if (!found) {
            cout << "Book not found.\n";
        }
    }
```

```
 void issueBook(int id) {
        int index = findBookIndexById(id);
        if (index != -1 &&
!books[index].isIssued) {
            books[index].isIssued = true;
            cout << "Book issued
successfully.\n";
        } else {
            cout << "Book not available for
issue.\n";
        }
    }

    void returnBook(int id) {
        int index = findBookIndexById(id);
        if (index != -1 &&
books[index].isIssued) {
            books[index].isIssued = false;
            cout << "Book returned
successfully.\n";
        } else {
            cout << "Book was not
```

```
 void listAllBooks() const {
        vector<Book> sortedBooks = books;
        sort(sortedBooks.begin(),
sortedBooks.end(), [](const Book& a, const
Book& b) {
            return a.id < b.id;
        });
        for (const auto& book :
sortedBooks) {
            book.displayBook();
        }
    }

    void deleteBook(int id) {
        int index = findBookIndexById(id);
        if (index != -1) {
            books.erase(books.begin() +
index);
            cout << "Book deleted
successfully.\n";
        } else {
            cout << "Book not found.\n";
        }
```

```
void displayMenu() {
    cout << "\nLibrary Management
System\n";
    cout << "1. Add New Book\n";
    cout << "2. Search Book by ID\n";
    cout << "3. Search Book by Title\n";
    cout << "4. Issue Book\n";
    cout << "5. Return Book\n";
    cout << "6. List All Books\n";
    cout << "7. Delete Book\n";
    cout << "8. Exit\n";
    cout << "Enter your choice: ";
}
```

```
int main() {
    Library library;
    int choice, id;
    string title, author;

    while (true) {
        displayMenu();
        cin >> choice;

        switch (choice) {
            case 1:
                cout << "Enter book ID: ";
                cin >> id;
                cin.ignore();  // Clear the
newline character from the buffer
                cout << "Enter book title:
";
                getline(cin, title);
                cout << "Enter book author:
";
                getline(cin, author);
                library.addBook(id, title,
author);
```

```
 case 2:
                cout << "Enter book ID to
search: ";
                cin >> id;
                library.searchBookById(id);
                break;
            case 3:
                cout << "Enter book title to
search: ";
                cin.ignore();  // Clear the
newline character from the buffer
                getline(cin, title);

library.searchBookByTitle(title);
                break;
            case 4:
                cout << "Enter book ID to
issue: ";
                cin >> id;
                library.issueBook(id);
                break;
            case 5:
                cout << "Enter book ID to
return: ";
                cin >> id;
```

```
 case 7:
                cout << "Enter book ID to
delete: ";
                cin >> id;
                library.deleteBook(id);
                break;
            case 8:
                cout << "Exiting the
system.\n";
                return 0;
            default:
                cout << "Invalid choice.
Please try again.\n";
        }
    }
}
```

Thankyou.