Assignment: Library Management System (C++)

Project Description
The system allows the user to manage books and users in a small library. It provides a menu where the user can add books, remove books, register users, borrow and return books, and view available records.

Design Choices
• Classes were used to represent real-world entities such as Book, User, and Library.
• Arrays were used instead of advanced data structures to keep the program easy to understand.
• A menu-driven approach was chosen so beginners can clearly follow program flow.

How to Run the Program
1. Open a terminal or command prompt.
2. Compile the program using g++ library.cpp -o library.
3. Run the program using ./library or library.exe on Windows.
4. Follow the menu options displayed on the screen.

Testing the Program
Testing is done manually by selecting menu options and checking the output. For example, adding a book should display a confirmation message, while searching for a borrowed book should show its status correctly.

Conclusion
This assignment helps beginners practice basic C++ programming skills, understand object-oriented design, and build confidence in writing simple but meaningful programs.

C++ Source Code

```
#include <iostream>
using namespace std;

class Book
{
public:
    string title;
    bool borrowed;

    Book()
    {
        borrowed = false;
    }
};

class User
{
public:
    string name;
};

class Library
{
private:
    Book books[10];
    User users[10];
    int bookCount;
    int userCount;

public:
    Library()
    {
        bookCount = 0;
```

```
        userCount = 0;
    }

    void addBook()
    {
        cout << "Enter book title: ";
        cin >> books[bookCount].title;
        books[bookCount].borrowed = false;
        bookCount++;
        cout << "Book added successfully
";
    }

    void removeBook()
    {
        string title;
        cout << "Enter book title to remove: ";
        cin >> title;

        for (int i = 0; i < bookCount; i++)
        {
            if (books[i].title == title)
            {
                books[i] = books[bookCount - 1];
                bookCount--;
                cout << "Book removed
";
                return;
            }
        }
        cout << "Book not found
";
    }

    void searchBook()
    {
        string title;
        cout << "Enter book title to search: ";
        cin >> title;

        for (int i = 0; i < bookCount; i++)
        {
            if (books[i].title == title)
            {
                cout << "Book found";
                if (books[i].borrowed)
                    cout << " (Borrowed)
";
                else
                    cout << " (Available)
";
                return;
            }
        }
        cout << "Book not found
";
    }

    void registerUser()
    {
        cout << "Enter user name: ";
        cin >> users[userCount].name;
        userCount++;
        cout << "User registered successfully
";
    }

    void borrowBook()
    {
        string title;
        cout << "Enter book title to borrow: ";
        cin >> title;

        for (int i = 0; i < bookCount; i++)
        {
            if (books[i].title == title)
            {
```

```
                if (books[i].borrowed)
                    cout << "Book already borrowed
";
                else
                {
                    books[i].borrowed = true;
                    cout << "Book borrowed successfully
";
                }
                return;
            }
        }
        cout << "Book not found
";
    }

    void returnBook()
    {
        string title;
        cout << "Enter book title to return: ";
        cin >> title;

        for (int i = 0; i < bookCount; i++)
        {
            if (books[i].title == title)
            {
                books[i].borrowed = false;
                cout << "Book returned successfully
";
                return;
            }
        }
        cout << "Book not found
";
    }

    void showBooks()
    {
        cout << "Books in library:
";
        for (int i = 0; i < bookCount; i++)
        {
            cout << books[i].title;
            if (books[i].borrowed)
                cout << " (Borrowed)";
            cout << endl;
        }
    }

    void showUsers()
    {
        cout << "Registered users:
";
        for (int i = 0; i < userCount; i++)
        {
            cout << users[i].name << endl;
        }
    }
};

int main()
{
    Library lib;
    int choice;

    do
    {
        cout << "\n--- Library Menu ---\n";
        cout << "1. Add Book\n";
        cout << "2. Remove Book\n";
        cout << "3. Search Book\n";
        cout << "4. Register User\n";
        cout << "5. Borrow Book\n";
        cout << "6. Return Book\n";
        cout << "7. Show Books\n";
        cout << "8. Show Users\n";
        cout << "0. Exit\n";
```

```
        cout << "Enter choice: ";
        cin >> choice;

        switch (choice)
        {
            case 1: lib.addBook(); break;
            case 2: lib.removeBook(); break;
            case 3: lib.searchBook(); break;
            case 4: lib.registerUser(); break;
            case 5: lib.borrowBook(); break;
            case 6: lib.returnBook(); break;
            case 7: lib.showBooks(); break;
            case 8: lib.showUsers(); break;
            case 0: cout << "Exiting...\n"; break;
            default: cout << "Invalid choice\n";
        }

    } while (choice != 0);

    return 0;
}
```