DSA Assignment: Library Management System

Name: M sai chetan

College: Lovely Professional University

Project Overview:

This project involves developing a simple library management system using C++. The primary objective is to create a system that allows librarians to manage and track the library's collection of books. The system provides functionalities to add new books, search for books, issue books to students, return books, list all books, and delete books from the system. This project serves as a practical application of various data structures and algorithms in C++.

Functionality:

1) Add New Books
2) Search for a Book
3) Issue a Book
4) Return a Book
5) List all Books
6) Delete a Book

Code Input:

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

struct Book {
    int id;
    string title;
    string author;
    bool isIssued;
    Book* next;

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

class Library {
private:
    Book* head;

    Book* searchById(int id) {
        Book* current = head;
        while (current != nullptr) {
            if (current->id == id)
                return current;
            current = current->next;
        }
        return nullptr;
    }

    Book* searchByTitle(string title) {
        Book* current = head;
        while (current != nullptr) {
            if (current->title == title)
```

```cpp
        Book* current = head;
        while (current != nullptr) {
            if (current->title == title)
                return current;
            current = current->next;
        }
        return nullptr;
    }

public:
    Library() : head(nullptr) {}

    void addBook(int id, string title, string author) {
        Book* newBook = new Book(id, title, author);
        if (head == nullptr) {
            head = newBook;
        } else {
            Book* current = head;
            while (current->next != nullptr) {
                current = current->next;
            }
            current->next = newBook;
        }
        cout << "Book added: " << title << endl;
    }

    void searchBook(int id) {
        Book* book = searchById(id);
        if (book) {
            cout << "Book found: " << book->title << " by " << book->author << endl;
        } else {
            cout << "Book not found." << endl;
        }
```

```
        } else {
            cout << "Book not found." << endl;
        }
    }

    void searchBook(string title) {
        Book* book = searchByTitle(title);
        if (book) {
            cout << "Book found: " << book->title << " by " << book->author << endl;
        } else {
            cout << "Book not found." << endl;
        }
    }

    void issueBook(int id, string studentName) {
        Book* book = searchById(id);
        if (book) {
            if (book->isIssued) {
                cout << "Book already issued." << endl;
            } else {
                book->isIssued = true;
                cout << "Book issued to " << studentName << endl;
            }
        } else {
            cout << "Book not found." << endl;
        }
    }

    void returnBook(int id) {
        Book* book = searchById(id);
        if (book) {
            if (book->isIssued) {
                book->isIssued = false;
```

```
        if (book) {
            if (book->isIssued) {
                Book* Library::returnBook::book
                cout << "Book returned." << endl;
            } else {
                cout << "Book was not issued." << endl;
            }
        } else {
            cout << "Book not found." << endl;
        }
    }

    void listBooks() {
        if (head == nullptr) {
            cout << "No books available." << endl;
            return;
        }
        Book* current = head;
        while (current != nullptr) {
            cout << current->id << ": " << current->title << " by " <<
            current->author << (current->isIssued ? " (Issued)" : "") << endl;
            current = current->next;
        }
    }

    void deleteBook(int id) {
        if (head == nullptr) {
            cout << "No books available." << endl;
            return;
        }
        if (head->id == id) {
            Book* toDelete = head;
            head = head->next;
```

```
        if (head->id == id) {
            Book* toDelete = head;
            head = head->next;
            delete toDelete;
            cout << "Book deleted." << endl;
            return;
        }
        Book* current = head;
        while (current->next != nullptr && current->next->id != id) {
            current = current->next;
        }
        if (current->next == nullptr) {
            cout << "Book not found." << endl;
        } else {
            Book* toDelete = current->next;
            current->next = current->next->next;
            delete toDelete;
            cout << "Book deleted." << endl;
        }
    }
};

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

    while (true) {
        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";
```

```
            break;
        case 4:
            cout << "Enter Book ID: ";
            cin >> id;
            cin.ignore();
            cout << "Enter Student Name: ";
            getline(cin, studentName);
            library.issueBook(id, studentName);
            break;
        case 5:
            cout << "Enter Book ID: ";
            cin >> id;
            library.returnBook(id);
            break;
        case 6:
            library.listBooks();
            break;
        case 7:
            cout << "Enter Book ID: ";
            cin >> id;
            library.deleteBook(id);
            break;
        case 8:
            return 0;
        default:
            cout << "Invalid choice. Try again." << endl;
        }
    }
    return 0;
}
```

Code output:

```
Library Management System
1. Add New Book
2. Search Book by ID
3. Search Book by Title
4. Issue Book
5. Return Book
6. List All Books
7. Delete Book
8. Exit
Enter your choice: 1
Enter Book ID: 8996
Enter Book Title: rich investments
Enter Book Author: sai chetan
Book added: rich investments
```

```
Library Management System
1. Add New Book
2. Search Book by ID
3. Search Book by Title
4. Issue Book
5. Return Book
6. List All Books
7. Delete Book
8. Exit
Enter your choice: 2
Enter Book ID: 8996
Book found: rich investments by sai chetan
```

```
Library Management System
1. Add New Book
2. Search Book by ID
3. Search Book by Title
4. Issue Book
5. Return Book
6. List All Books
7. Delete Book
8. Exit
Enter your choice: 3
Enter Book Title: rich investments
```

```
Library Management System
1. Add New Book
2. Search Book by ID
3. Search Book by Title
4. Issue Book
5. Return Book
6. List All Books
7. Delete Book
8. Exit
Enter your choice: 4
Enter Book ID: 8996
Enter Student Name: sai chetan
Book issued to sai chetan
```

```
Library Management System
1. Add New Book
2. Search Book by ID
3. Search Book by Title
4. Issue Book
5. Return Book
6. List All Books
7. Delete Book
8. Exit
Enter your choice: 5
Enter Book ID: 8996
Book returned.
```

```
Library Management System
1. Add New Book
2. Search Book by ID
3. Search Book by Title
4. Issue Book
5. Return Book
6. List All Books
7. Delete Book
8. Exit
Enter your choice: 6
8996: rich investments by sai chetan
```