Final Project

Project: Library Management System

Source Code:

```
import java.util.ArrayList;
import java.util.Scanner;

class Book {
  String title;
  String author;
  boolean isAvailable;

  public Book(String title, String author) {
    this.title = title;
    this.author = author;
    this.isAvailable = true;
  }

  public void borrowBook() {
    if (isAvailable) {
      isAvailable = false;
      System.out.println(title + " has been borrowed.");
    } else {
      System.out.println(title + " is currently not available.");
    }
  }

  public void returnBook() {
    isAvailable = true;
    System.out.println(title + " has been returned.");
  }

  public void displayInfo() {
    System.out.println("Title: " + title + ", Author: " + author + ", Available: " + isAvailable);
  }
}

public class LibrarySystem {
  static ArrayList<Book> library = new ArrayList<>();
  static Scanner scanner = new Scanner(System.in);

  public static void addBook() {
    System.out.print("Enter title: ");
    String title = scanner.nextLine();
    System.out.print("Enter author: ");
    String author = scanner.nextLine();

    library.add(new Book(title, author));
    System.out.println("Book added successfully.");
  }
```

```
public static void displayBooks() {
    if (library.isEmpty()) {
        System.out.println("Library is empty.");
    } else {
        for (Book book : library) {
            book.displayInfo();
        }
    }
}

public static void searchBook() {
    System.out.print("Enter title to search: ");
    String title = scanner.nextLine();
    boolean found = false;
    for (Book book : library) {
        if (book.title.equalsIgnoreCase(title)) {
            book.displayInfo();
            found = true;
            break;
        }
    }
    if (!found) {
        System.out.println("Book not found.");
    }
}

public static void borrowBook() {
    System.out.print("Enter title to borrow: ");
    String title = scanner.nextLine();
    for (Book book : library) {
        if (book.title.equalsIgnoreCase(title)) {
            book.borrowBook();
            return;
        }
    }
    System.out.println("Book not found.");
}

public static void returnBook() {
    System.out.print("Enter title to return: ");
    String title = scanner.nextLine();
    for (Book book : library) {
        if (book.title.equalsIgnoreCase(title)) {
            book.returnBook();
            return;
        }
    }
    System.out.println("Book not found.");
}

public static void main(String[] args) {
```

```
        int choice;
        do {
            System.out.println("\n--- Library Menu ---");
            System.out.println("1. Add Book");
            System.out.println("2. Display All Books");
            System.out.println("3. Search Book");
            System.out.println("4. Borrow Book");
            System.out.println("5. Return Book");
            System.out.println("0. Exit");
            System.out.print("Enter your choice: ");
            choice = Integer.parseInt(scanner.nextLine());

            switch (choice) {
                case 1: addBook(); break;
                case 2: displayBooks(); break;
                case 3: searchBook(); break;
                case 4: borrowBook(); break;
                case 5: returnBook(); break;
                case 0: System.out.println("Exiting..."); break;
                default: System.out.println("Invalid choice.");
            }
        } while (choice != 0);
    }
}
```

Output:

```
D:\5G1>javac LibrarySystem.java

D:\5G1>java LibrarySystem

--- Library Menu ---
1. Add Book
2. Display All Books
3. Search Book
4. Borrow Book
5. Return Book
0. Exit
Enter your choice: 1
Enter title: Java Programming
Enter author: James Gosling
Book added successfully.

--- Library Menu ---
1. Add Book
2. Display All Books
3. Search Book
4. Borrow Book
5. Return Book
0. Exit
Enter your choice: 1
Enter title: Data Structures
```

Enter author: Mark Allen Weiss
Book added successfully.

--- Library Menu ---
1. Add Book
2. Display All Books
3. Search Book
4. Borrow Book
5. Return Book
0. Exit
Enter your choice: 1
Enter title: Clean Code
Enter author: Robert C. Martin
Book added successfully.

--- Library Menu ---
1. Add Book
2. Display All Books
3. Search Book
4. Borrow Book
5. Return Book
0. Exit
Enter your choice: 1
Enter title: Introduction to Algorithms
Enter author: Thomas H. Cormen
Book added successfully.

--- Library Menu ---
1. Add Book
2. Display All Books
3. Search Book
4. Borrow Book
5. Return Book
0. Exit
Enter your choice: 2
Title: Java Programming, Author: James Gosling, Available: true
Title: Data Structures, Author: Mark Allen Weiss, Available: true
Title: Clean Code, Author: Robert C. Martin, Available: true
Title: Introduction to Algorithms, Author: Thomas H. Cormen, Available: true

--- Library Menu ---
1. Add Book
2. Display All Books
3. Search Book
4. Borrow Book
5. Return Book
0. Exit
Enter your choice: 3
Enter title to search: Clean Code
Title: Clean Code, Author: Robert C. Martin, Available: true

--- Library Menu ---
1. Add Book
2. Display All Books
3. Search Book
4. Borrow Book
5. Return Book
0. Exit
Enter your choice: 3
Enter title to search: Python Programming
Book not found.

--- Library Menu ---
1. Add Book
2. Display All Books
3. Search Book
4. Borrow Book
5. Return Book
0. Exit
Enter your choice: 4
Enter title to borrow: Introduction to Algorithms
Introduction to Algorithms has been borrowed.

--- Library Menu ---
1. Add Book
2. Display All Books
3. Search Book
4. Borrow Book
5. Return Book
0. Exit
Enter your choice: 4
Enter title to borrow: Introduction to Algorithms
Introduction to Algorithms is currently not available.

--- Library Menu ---
1. Add Book
2. Display All Books
3. Search Book
4. Borrow Book
5. Return Book
0. Exit
Enter your choice: 2
Title: Java Programming, Author: James Gosling, Available: true
Title: Data Structures, Author: Mark Allen Weiss, Available: true
Title: Clean Code, Author: Robert C. Martin, Available: true
Title: Introduction to Algorithms, Author: Thomas H. Cormen, Available: false

--- Library Menu ---
1. Add Book
2. Display All Books
3. Search Book
4. Borrow Book

5. Return Book
0. Exit
Enter your choice: 5
Enter title to return: Introduction to Algorithms
Introduction to Algorithms has been returned.

--- Library Menu ---
1. Add Book
2. Display All Books
3. Search Book
4. Borrow Book
5. Return Book
0. Exit
Enter your choice: 2
Title: Java Programming, Author: James Gosling, Available: true
Title: Data Structures, Author: Mark Allen Weiss, Available: true
Title: Clean Code, Author: Robert C. Martin, Available: true
Title: Introduction to Algorithms, Author: Thomas H. Cormen, Available: true

--- Library Menu ---
1. Add Book
2. Display All Books
3. Search Book
4. Borrow Book
5. Return Book
0. Exit
Enter your choice: 0
Exiting...

Library Management System

- Java Project
- **Presented by:**

Siva Naga Sai Deepthi Vulli

Introduction

A Library Management System is software to manage and track books in a library.

Users can add, search, borrow, return, and display books.

Built in Java using Object-Oriented Programming (OOP) concepts.

Working of the System

1. Menu-driven console interface.
2. User selects an option: Add, Display, Search, Borrow, or Return Book.
3. Each book has a Title, Author, and Availability status.
4. Books are stored using an ArrayList.

Main Features

- ADD NEW BOOKS

- VIEW ALL BOOKS

- SEARCH FOR A BOOK BY TITLE

- BORROW A BOOK (IF AVAILABLE)

- RETURN A BOOK

- CONSOLE-BASED INTERACTION

Uses

- Suitable for schools, colleges, and small libraries.
- Good practice for learning Java OOP concepts.
- Replaces manual register systems.
- Can be extended for real-world scenarios.

Advantages

- Easy to use and maintain.

- Clean, modular code structure.

- Covers key Java concepts: classes, objects, lists.

- Fast development and execution.

Disadvantages

- Console-based only, no GUI.

- Data not stored permanently (no file/database).

- Limited to one user at a time.

- Not suitable for large libraries without changes.

Future Scope

• Add file handling or a database (MySQL, SQLite).

• Add login system for librarians/users.

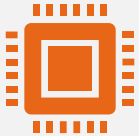
• Implement GUI using Java Swing/JavaFX.

• Include due dates, fine calculation, and categories.

Conclusion

- A simple yet effective mini project.

- Demonstrates Java fundamentals well.



- Serves as a base for building advanced systems.

Thank You