Week-3 Java Programming Basics

Library Management System

Description

A Java console application designed to manage essential library operations. Users can add books, register members, borrow and return books, view records, search for books, and calculate fines for late returns.The project demonstrates **Java fundamentals**, object-oriented programming, menu-driven logic, and user input handling.

Features

- Add new books to the library
- Register and manage library members
- Display all books and members
- Borrow and return books with due dates
- Calculate fines for delayed returns
- Search books by title, author, or genre
- Menu-driven, user-friendly console interface
- Input validation for all user actions
- (Optional) File-based data persistence

How to Run

```
# Compile Java files
javac -d bin src/library/*.java

# Run the application
java -cp bin library.Main
```

Sample Menu

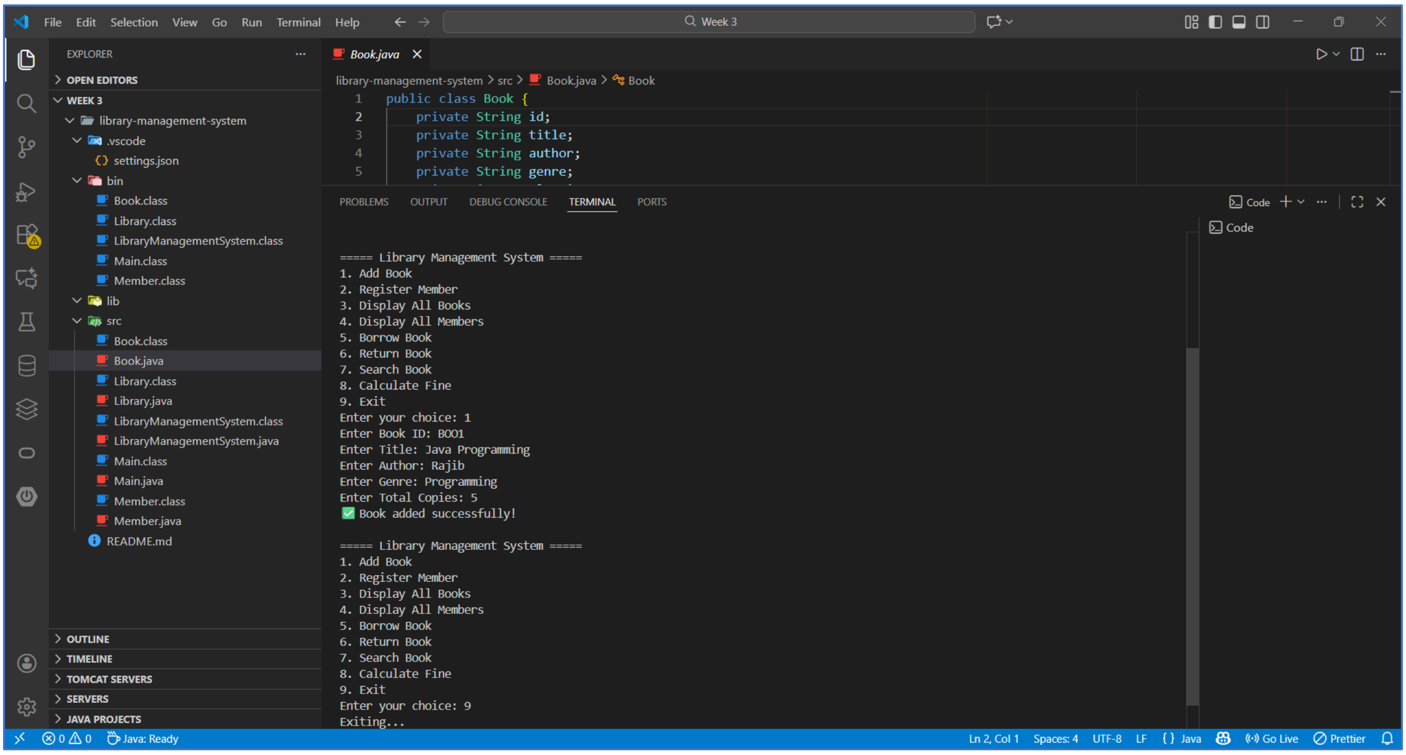
```
=== LIBRARY MANAGEMENT SYSTEM ===
1. Add New Book
2. Display All Books
3. Register Member
4. Display All Members
5. Borrow Book
6. Return Book
7. Search Book
8. Calculate Fine
9. Exit

Enter your choice:
```

Code Structure

```
library-management/
│── src/
│   └── library/
│       ├── Main.java
│       ├── Book.java
│       ├── Member.java
│
```

```
│       ├── LibraryService.java
│       └── Utils.java   (optional)
│
└── bin/   # Compiled .class files
```

Visual Documentation



```java
public class Book {
    private String id;
    private String title;
    private String author;
    private String genre;
```

```
===== Library Management System =====
1. Add Book
2. Register Member
3. Display All Books
4. Display All Members
5. Borrow Book
6. Return Book
7. Search Book
8. Calculate Fine
9. Exit
Enter your choice: 1
Enter Book ID: B001
Enter Title: Java Programming
Enter Author: Rajib
Enter Genre: Programming
Enter Total Copies: 5
✅ Book added successfully!

===== Library Management System =====
1. Add Book
2. Register Member
3. Display All Books
4. Display All Members
5. Borrow Book
6. Return Book
7. Search Book
8. Calculate Fine
9. Exit
Enter your choice: 9
Exiting...
```

Technical Details

- **Data Structures:**
  - ArrayList<Book> for book records
  - ArrayList<Member> for member records
  - Objects store attributes like ID, title, author, total copies, borrowed status
- **Algorithms:**
  - Linear search for book/member lookup
  - Condition-based fine calculation
  - Input validation loops for correct entries
- **Architecture:**
  - Modular design using classes: Book, Member, LibraryService, Main
  - Separation of concerns for better maintainability
    (Reference: Java SE Programming Guide)

Testing Evidence

- Tested adding/removing multiple books
- Verified borrow/return behavior
- Checked invalid inputs & error messages
- Tested search functions (title, author, genre)
- Verified fine calculation with different delays