Library Management System - Explanation

1. Introduction

The Library Management System (LMS) is a beginner-friendly, console-based Python project that simulates basic operations of a library. This project helps new programmers understand object-oriented programming (OOP), file handling using JSON, and simple user interactions using terminal I/O.

2. Objectives

- Apply Object-Oriented Programming (OOP) concepts.
- Use classes and encapsulation to manage data.
- Persist data using JSON file storage.
- Provide a text-based menu interface for the user.

3. Class Overview

3.1 Book Class

Attributes: title, author, ISBN, is_borrowed

Methods: borrow(), return_book(), to_dict(), __str__()

Purpose: Represents a single book and its current state.

3.2 User Class

Attributes: name, user_id, list of borrowed book ISBNs

Methods: add_borrowed_book_isbn(), remove_borrowed_book_isbn(), to_dict(), __str__()

Purpose: Represents a library user and tracks borrowed books.

3.3 Library Class

Manages the collection of books and users.

Handles adding/removing/searching books, borrowing/returning, and JSON file persistence.

4. File Handling with JSON

Two JSON files (books.json, users.json) are used to store and retrieve library data.

_save_data() saves all book and user objects to files.

_load_data() reads from files and recreates objects.

5. Terminal-Based Menu Interface

The program includes a while loop that displays menu options to the user.

Operations include: Add Book, Register User, Borrow, Return, Search, Display.

The interface uses simple text input/output, making it beginner-friendly.

6. Output & Testing

- JSON storage works correctly.
- Prevents borrowing the same book twice.
- Prevents deletion of borrowed books.
- Displays appropriate feedback for each action.

7. Conclusion

The LMS project is ideal for beginners to practice OOP, data handling, and menu-driven programs.

It encourages modular thinking and clear class responsibilities.