# Library Management System – Assignment Question for Students

## Problem Statement

You are required to develop a **Library Management System** in Python.
The system should allow the librarian to:

1. Add new students
2. Add new books
3. Issue books to students for a limited number of days
4. Return the books
5. Automatically calculate a **fine** if the book is returned after the due date

This program should run in the **command-line/terminal** and should use **file handling** to permanently store data.

---

# Part 1: Requirements

### You must create and use the following files:

- `students.json` → stores all students
- `books.json` → stores all books
- `issued_books.json` → stores issued book details

  These files must be created and updated automatically by your program using Python file handling.

---

# Part 2: Functional Requirements

### 1. Add Student

- Input: Student ID and Student Name
- Store them in `students.json`
- If a student already exists, program should not duplicate entry

---

### 2. Add Book

- Input: Book ID and Book Title
- Every new book must have:

    - `title`
    - `available` status = True
- Save data in `books.json`

---

## 3. Issue Book to Student

- Ask Student ID
- Ask Book ID
- Check:
    - If student exists
    - If book exists
    - If book is already issued
- Ask for number of days to issue (e.g., 5 days)
- Store the following in `issued_books.json`:
    - `student_id`
    - `issue_date`
    - `due_date`
    - Use `datetime` module for date handling
- When a book is issued → mark its `available = False` in `books.json`

---

## 4. Return Book

- Ask Book ID
- Check if this book is issued
- Read issue information:
    - issued date
    - due date
    - student details
- Calculate fine:

```
If book is returned after due date:
      Fine = Rs. 10 per day
Else:
      No fine
```

- After return:
    - Mark book as `available = True`
    - Remove that book entry from `issued_books.json`

---

### 5. Exit Program

When user selects exit, the program should stop safely.

---

# Part 3: Technical Requirements

### Your program MUST use:

✓ Functions
✓ File Handling (JSON read/write)
✓ Exception Handling
✓ Loops
✓ Conditional Statements
✓ `datetime` module for date calculation

---

# Part 4: Menu System

Your program must display the following menu repeatedly:

```
========= LIBRARY MANAGEMENT SYSTEM =========
1. Add Student
2. Add Book
3. Issue Book
4. Return Book
5. Exit
============================================
```

The user must be able to select any option until they choose Exit.

---

# Part 5: Expected Output Examples

### ✓ When issuing a book:

```
Enter Student ID: 101
Enter Book ID: B1
Enter number of days: 5
Book issued successfully!
Due date: 2025-12-15
```

### ✓ When returning a book:

```
Enter Book ID: B1
Book Title: Python Basics
Issued to: Akshat
Due Date: 2025-12-15
Return Date: 2025-12-18

! Book returned late by 3 days
   Fine to be paid: Rs. 30
Book returned successfully!
```

---

# Part 6: BONUS TASK (Optional)

For extra marks (5 marks each):

✓ Add feature to display all books
✓ Add feature to display all issued books
✓ Add feature to search book by title
✓ Add feature to list all students
✓ Store fine history in a new file

---

# Part 7: Submission Requirements

Students must submit:

1. Python File → `library_system.py`
2. Screenshot of program output
3. All generated `.json` files
4. A short explanation (in 6–8 lines) about how the program works

Just tell me!