# **📚 Problem Statement: Library Management System**

Write a Python program to simulate a **library book issue/return system**.

## **✅ Requirements**

1. **Display Available Books**
   1. Show a list of all books available in the library with details:
      1. Book Name
      2. Author (optional)
      3. Availability Status (Available / Issued)
2. **Student Actions**
   1. A student can:
      1. **Issue a book**
      2. **Return a book**
      3. **View borrowed books**
3. **Track Borrowed Books**
   1. Maintain a record of which books a student has borrowed.
   2. Track the **issue date** to calculate fines for late returns.

## **💰 Library Rules**

1. A student can **issue a maximum of 3 books** at a time.
2. Books returned after **7 days** incur a fine of **₹10 per day**.
3. When returning, calculate any fines if the book is overdue.
4. Display a summary for the student including:
   1. Borrowed books
   2. Issue dates
   3. Return dates (if returned)
   4. Fines (if any)

## **🔹 Concepts to Use**

* **Functions**
  + show\_books() → Display all books with availability.
  + issue\_book(student\_name, book\_name) → Issue a book, check limits, update records.
  + return\_book(student\_name, book\_name, return\_date) → Return a book, calculate fines.
  + view\_borrowed\_books(student\_name) → Show all books borrowed by a student.
  + library\_system() → Main function controlling the flow.
* **Loops**
  + Allow students to perform multiple actions until they choose to exit.
* **Conditions**
  + Check for maximum books issued.
  + Check if the book is available.
  + Calculate fines based on return date and issue date.
* **Data Structures**
  + Use **list of dictionaries** for books:

books = [  
 {"name": "Python Basics", "author": "John Doe", "available": True},  
 {"name": "Data Science 101", "author": "Jane Smith", "available": True},  
 {"name": "Machine Learning", "author": "Alice Brown", "available": True}  
]

* + Use a **dictionary** for students’ borrowed books:

borrowed\_books = {  
 "Student1": [  
 {"book\_name": "Python Basics", "issue\_date": "2025-09-25"}  
 ]  
}

* **Date Handling**
  + Use Python’s datetime module to calculate overdue days and fines.
* **String Formatting**
  + Display a **clear summary** of borrowed books and fines.

## **🚀 Implementation Strategy**

1. **Step 1: Setup Data**
   1. Define books with name, author, and availability.
   2. Initialize empty record for students’ borrowed books.
2. **Step 2: Display Menu**
   1. Options: Issue a book, Return a book, View borrowed books, Exit.
3. **Step 3: Issue Book**
   1. Check if student has issued less than 3 books.
   2. Check if selected book is available.
   3. Record issue date and mark book as unavailable.
4. **Step 4: Return Book**
   1. Check if student has borrowed that book.
   2. Ask for return date.
   3. Calculate days elapsed since issue.
   4. Apply fine if days > 7 (₹10/day).
   5. Mark book as available again.
5. **Step 5: View Borrowed Books**
   1. Display all books borrowed by the student with issue dates and fines (if any).
6. **Step 6: Loop Option**
   1. After one action, allow student to select another action until exit.