DESIGN RATIONALE FOR MINI LIBRARY MANAGEMENT SYSTEM

Introduction

The Mini Library Management System is designed to manage essential library operations, including adding, searching, updating, deleting, borrowing, and returning books. The system utilizes fundamental Python data structures, including dictionaries, lists, and tuples, as well as functions, to store, manipulate, and retrieve information efficiently.

The design approach focuses on simplicity, data integrity, and modularity. Each component of the system (books, members, and genres) is represented using the most suitable data structure based on how the data behaves and how it will be accessed. By combining these structures with well-defined functions, the system achieves both flexibility and clarity in its implementation.

Choice of Data Structures

1. Dictionaries for Books

Books are stored using a dictionary where the key is the ISBN (International Standard Book Number), and the value is another dictionary containing details like title, author, genre, total copies, and available copies.
This choice was made because:

- Dictionaries allow fast access, insertion, and deletion using unique keys.
- The ISBN serves as a unique identifier for each book, preventing duplication.
- Nested dictionaries make it easier to organize multiple attributes of a single book in a structured and readable way.

2. Lists of Dictionaries for Members

Members are stored in a list, where each member is represented as a dictionary containing the member's ID, name, email, and a list of borrowed books.

This structure was selected because:

- Lists are ordered and easy to iterate over when searching for a member.
- Dictionaries inside the list store each member's personal details and borrowed books clearly.
- The system can dynamically grow as more members are added.

3. Tuples for Genres

A tuple is used to store valid book genres, such as: (Fiction, Non-Fiction, Sci-Fi, Mystery, Romance)

Tuples are immutable, meaning their content cannot be accidentally changed.
This ensures that the list of valid genres remains constant throughout program execution, promoting data consistency.
If lists were used instead, accidental additions or deletions could compromise the validity of book data.

Conclusion

I chose these designs because they balance efficiency, clarity, and scalability.

- Dictionaries ensure fast and organized access to book data.
- Lists of dictionaries provide flexible and dynamic member management.
- Tuples guarantee data integrity for fixed categories.

Together with modular functions, these choices create a well-structured and reliable Mini Library Management System.
This project illustrates how fundamental Python structures can effectively solve real-world management problems while keeping the system lightweight and easy to expand.