**Design Rationale**

This Library Management System was built using simple Python data structures selected for their clarity, efficiency, and suitability to the system’s needs.

**Dictionary for Books**  
The use of the dictionary was because of its unique property to store a collection of key value pairs and there mutability allowing for unordered collection of information where each unique key maps to a specific value. This structure allows for efficient retrieval of values by referencing their corresponding keys. And there for it was a great fit for the library management app

**List for Members**  
Members are managed using a list of dictionaries. Each dictionary holds details like member\_id, name, email, and a list of borrowed books. Lists are simple, flexible, and efficient for iterating through a moderate number of records, which suits the system well. This makes it easy to add, search for, or update members as needed.

**Tuple for Genres**  
For the use of the tuple we took advantage of its immutability property which is very suitable for things like genre that we don’t want people to add too or change in anyway shape or form.

**Functions for Operations**  
All system operations—adding, searching, updating, deleting, borrowing, and returning books—are implemented as separate functions. Each function performs proper input validation and returns a Boolean value (True or False) to indicate success or failure. This design supports modular programming and makes automated testing easy with assert statements.

**Data Integrity**  
The system maintains accuracy and consistency of data through several rules:

* Every book must have a unique ISBN.
* Book genres must match the predefined tuple of valid genres.
* Books cannot be deleted if they are currently borrowed.
* Members cannot be deleted while they still have borrowed books.
* Each member can borrow a maximum of three books at once.

Overall, these design decisions create a system that is efficient, simple to understand, and easy to maintain. The use of dictionaries, lists, and tuples ensures fast data handling and reliability, while the function-based structure promotes reusability, clarity, and effective testing.

Conclusion

In conclusion, the Library Management System was designed to be simple, efficient, and easy to understand by using Python’s core data structures—dictionaries, lists, and tuples. Each structure was chosen for a specific purpose: dictionaries for fast and organized book management, lists for flexible handling of members, and tuples for secure and unchangeable genre validation. The use of independent functions for each operation ensures modularity, making the system easier to test, maintain, and expand in the future. Through input validation and data integrity rules, the system prevents errors and maintains reliable records of books and members. Overall, this design provides a solid foundation for a small-scale library management solution that balances performance, accuracy, and simplicity.