Rationale

This project was developed to simulate a basic **Library Management System** that allows users to manage books and members using simple operations. The program focuses on essential **CRUD (Create, Read, Update, Delete)** functionalities, along with the ability to **borrow and return books**.

The system is built using **Python** because of its readability, flexibility, and suitability for data handling through built-in data structures like **dictionaries** and **lists**. The operations.py file serves as the logic core of the program, defining all necessary functions such as add_book, add_member, borrow_book, return_book, and delete_book. These functions interact with in-memory data to perform real-time operations without the need for an external database, keeping the system simple and efficient for beginners.

The demo.py file demonstrates how the system works by executing each function step-by-step and displaying the outputs. This allows users to observe how data changes with each operation. The tests.py file provides automated testing using assert statements to confirm that all functions perform correctly and produce expected results, ensuring program reliability and correctness.

Overall, this system provides a practical introduction to **data management, modular programming, and software testing** in Python. It mirrors how real-world library systems handle book records and member interactions, giving users both conceptual understanding and hands-on experience in building structured applications.