Task: Console-Based Library Management System

Objective:

Create a simple console-based library management system in Go to demonstrate the use of structs, interfaces, and other Go functionalities such as methods, slices, and maps.

Requirements

1. **Structs**:
    - Define a `Book` struct with the following fields:
        - `ID` (int)
        - `Title` (string)
        - `Author` (string)
        - `Status` (string) // can be "Available" or "Borrowed"

    - Define a `Member` struct with the following fields:
        - `ID` (int)
        - `Name` (string)
        - `BorrowedBooks` ([]Book) // a slice to hold borrowed books

2. **Interfaces**:
    - Define a `LibraryManager` interface with the following methods:
        - `AddBook(book Book)`
        - `RemoveBook(bookID int)`
        - `BorrowBook(bookID int, memberID int) error`
        - `ReturnBook(bookID int, memberID int) error`
        - `ListAvailableBooks() []Book`
        - `ListBorrowedBooks(memberID int) []Book`

3. **Implementation**:

    - Implement the `LibraryManager` interface in a `Library` struct. The `Library` struct should have a field to store all books (use a map with book ID as the key) and a field to store members (use a map with member ID as the key).

4. **Methods**:
    - Implement the methods defined in the `LibraryManager` interface:
        - `AddBook`: Adds a new book to the library.
        - `RemoveBook`: Removes a book from the library by its ID.
        - `BorrowBook`: Allows a member to borrow a book if it is available.
        - `ReturnBook`: Allows a member to return a borrowed book.
        - `ListAvailableBooks`: Lists all available books in the library.
        - `ListBorrowedBooks`: Lists all books borrowed by a specific member.

5. **Console Interaction**:
    - Create a simple console interface to interact with the library management system. Implement functions to:
        - Add a new book.
        - Remove an existing book.
        - Borrow a book.
        - Return a book.
        - List all available books.
        - List all borrowed books by a member.

Folder Structure

Follow the following folder structure for this task:

- `library_management/`
- `├── main.go`
- `├── controllers/`
- `│   └── library_controller.go`
- `├── models/`
- `│   └── book.go`
- `│   └── member.go`
- `├── services/`
- `│   └── library_service.go`
- `├── docs/`
- `│   └── documentation.md`
- `└── go.mod`

- **main.go**: Entry point of the application.
- **controllers/library_controller.go**: Handles console input and invokes the appropriate service methods.
- **models/book.go**: Defines the Book struct.
- **models/member.go**: Defines the Member struct.
- **services/library_service.go**: Contains business logic and data manipulation functions.
- **docs/documentation.md**: Contains system documentation and other related information.
- **go.mod**: Defines the module and its dependencies.

Evaluation Criteria

- **Correct Implementation**: Ensure that the structs, interfaces, and methods are correctly defined and implemented according to the requirements.
- **Functionality**: Verify that all required functionalities (adding, removing, borrowing, returning books, and listing books) are correctly implemented and working as expected.
- **Error Handling**: Ensure that appropriate error handling is implemented, particularly for scenarios where books or members are not found, or books are already borrowed.
- **Code Structure**: Verify that the code follows the provided folder structure and is organized in a clear and maintainable manner.
- **Documentation**