Design a library system where the books can be added to the repository, existing books records can be updated or even removed. New users can be added, updated or removed from the system and they can allocate book(s).

Task 1: Add 20 books to the system. Take the hardcoded data as there is no UI/Web page involved. A book can have a Book ID, category, name, shelf number, price and any other etc.

Task 2: Add 10 users to the system. Take the hardcoded data as there is no UI/Web page involved. A user can have a user ID, Name and list of books issued (**bookIssuedList**).

Task 3: Write an API that fetches the list of all books (only list of books names).

Task 4: Write an endpoint to add a new user to the users List.

Task 5: Write an API that fetches a detail of a book. (Fetch book by name). This should also indicate weather this book is available or already issues by a user.

Task 6: Write an API that fetches a detail of a user. This should bring the books issued by that user.

Task 7: Write an endpoint to add a new book to the repository.

Task 8: Write an endpoint to add a new user to the system.

Task 9: Another endpoint is required to update the book record. (Book can be identified by its name)

Task 10: Write an endpoint is required to update the user record.

Task 11: If a user issues a new book, we need to add it to the **bookIssuedList.** Write an API endpoint for it (The book that is already issued should not be issued again until it is returned).

Task 12: Finally write endpoint to remove a book from the repository by its name (if the book is issued by any user then delete all the book record from the **bookIssuedList** too)

Task 13: Write an API that remove the user from the system (user must have returned all the books back to the library otherwise the user will not get deleted.)

Task 14: Write an API that remove the book from the **bookIssuedList** when the issued book is returned back to the library.