***Test details & explanation***

1.**test\_get\_books**

Test the book number is correct-positive functionality

3.**test\_new\_book**

Test new book creation-positive functionality

4.**test\_book\_details**

Test all book details-positive functionality

3.**test\_add\_book**

Test that adding a book is a success(uses fixture to add the book)

Test it’s title & author-positive functionality

4.**test\_update\_book**

Test updating a book succeeds-positive functionality

Test that the title is updated successfully

5.**test\_update\_non\_exist\_book**

Negative test to see we get 404(resource not found) when book not found

6.**test\_delete\_book**

Test that a book is deleted successful-positive functionality

**7.test\_borrow\_book**

Test that borrowing book is a success and we get correct borrowing book

Confirmation.

positive functionality

8.test\_borow\_not-exist\_book

Test we get 404 when trying to borrow book with non exist id number

Negative functionality.

9.**test\_return\_book**

Test that returning a book is a success & return confirmation message is

Valid-positive functionality.

10.**test\_return\_non\_exist\_book**

Test that trying to return a non exist book will return 404

Negative functionality.

11.**test\_add\_and\_update\_book**

Integration of 2 operations verify that after adding a book it

Can be updated successfully.

positive functionality

12.**test\_add\_and\_\_delete\_new\_book**

Another integration of two operations

positive functionality

13.**test\_and\_delete\_not\_existant\_book**

Negative integration of two operations functionality

14 .test\_add\_and\_double\_delete\_book

Test adding one time a book and trying to delete it twice fails

With 404 .

Negative integration of two operations functionality

**15.test\_get\_user\_number**

Test the number of users

Positive functionality.

**16.test\_users\_details**

Test for each user its details:

1.Id

2. name

Positive functionality

17.Test\_sanity\_check

Test we can get all the books and all the users

Sanity for the system.

USERS TESTS:

1.test\_get\_users

Test get all users-positive sanity-positive sanity

2.test\_get\_user

Test get specific users by it’s id-positive sanity

3.test\_get\_users\_with\_wrong\_url\_suffix

Test getting user with wrong url-negative sanity

**Integration tests**

Test Add and Borrow New Book**:**

This test involves adding a new book to the system and then borrowing it. The test would verify that the book can be successfully added and then borrowed, ensuring that the system correctly handles the addition and borrowing processes.

Test Borrow and Return Book Flow**:**

This test checks the flow of borrowing a book and then returning it. It ensures that the book's status is updated correctly when borrowed and returned, verifying the system's ability to manage book availability.

**Test Return Book Not Borrowed:**

This test attempts to return a book that hasn't been borrowed. It checks that the system handles this scenario gracefully, likely expecting an error or specific response indicating that the book cannot be returned because it wasn't borrowed.

**Test Delete Borrowed Book:**

This test tries to delete a book that is currently borrowed. It verifies whether the system prevents this action or handles it appropriately, ensuring that borrowed books cannot be deleted without being returned first.

**Test Update Book Information:**

This test updates the information of an existing book and verifies that the changes are reflected in the system. It checks that the book's details, such as title and author, are updated correctly.

**Positive tests** are important to test system happy flows

These are the sanity tests.

**Negative testing** is also important so that the system behave as expected

When it fails.

**Integration testing** is important because we must verify that few operations

Work as expected together