**Steps to run the application**

1. Make schema in Mysql Workbench with name “library”.
   1. Make table with name “user”. Cols : username & password.
   2. Insert data for username and password in Mysql
2. Run AdvanceJavaAssignment5-2 as Java Application. Two tables will be created : Books & Author in Mysql database.
3. Run AdvanceJavaAssignment5-1 on server.
4. POST : <http://localhost:8082/Library/author> . Run this url in postman to add new authors in the database. (Run author payload)
5. GET : <http://localhost:8082/Library/authors> . Run this url in postman to see all entered authors.
6. PUT : <http://localhost:8082/Library/author/1> . . Run this url in postman to update the details of one author. (Run author payload)
7. DELETE : <http://localhost:8082/Library/author/1> . Run this url in postman to delete one author from database.
8. POST :. Run this url in postman to add new authors in the database. (Run author payload)

**Flow of program : AdvanceJavaAssignment5-2**

Controller > Service > Repository > Model

1. The url hits code written in Controller class.
2. The corresponding tasks are present in Service package, which contain below two interfaces and there corresponding implementation classes .
   1. Author Service
   2. AuthorServiceImp
   3. BookService
   4. BookServiceImp  
        
      For above two interfaces there are two implementation classes for loose coupling. So that only interface can be changed whenever new features are added.
3. Through the implementation classes task is directed towards Repository package. There are two classes:
   1. AuthorRepository ( Author model & Primary Key wrapper class is passed)
   2. BookRepository ( Book model & Primary Key wrapper class is passed)  
        
      Above two classes extends JpaRepository. JpaRepository is a**JPA (Java Persistence API)** specific extension of Repository. It contains the full API of **CrudRepository and PagingAndSortingRepository**. So it contains API for basic CRUD operations and also API for pagination and sorting.
4. In spring boot beans are created by component and consumed by autowired (called it dependecy injection(di) loose coupled)

Payload for create & Update.

Author Payload

{

"authorId" : "1",

"authorName" : "ABCDE"

}

Payload for book

{

"bookCode" : "1",

"bookTitle" : "ABCDE",

"addedOn" : "1",

"author" : {

"authorId" : "1"

}

}

**Flow of program : AdvanceJavaAssignment5-1**

login -> login controller -> listing.jsp -> 1) add controller(get,post(on submitting form -> listing.jsp))  
                              2) edit controller(get,post(on submitting form -> listing.jsp))  
                              3) delete controller(modal(listing) -> on delte -> delete controller-> listing.jsp)

Rest app connected through rest client service  
WebMvc -> web.xml  
Appconfig -> -servlet.xml (beans) those are consumed by autowired  
Hibernate config -> cfg.xml

RestClientService is the main class which connects Assignment5-1 to Assignment5-2.

In spring boot beans are created by component and consumed by autowired (called it dependecy injection(di) loose coupled)