**LAB 6**

This lab you will continue your previous lab that contains the Post entity. You will be adding another entity with similar functionalities based on the following:

public class User {  
  
 long id;  
 String name;  
 List<Post> posts;  
  
}

1. Make a domain model using the domain model above.
2. Make a repository and database that will hold the Users.
3. Make a service layer that will inject the repository.
4. Make an ORM relation between the User and Post entities, where a User holds a collection of Posts. Create it using uni-directional and JoinTable or JoinColumn
5. Make a RestController for the User that will inject the service and respond to the following requests.
   1. GET localhost:8080/users 🡪 This should retrieve all the users in the database.
   2. GET localhost:8080/users/1 🡪 This should retrieve the user with id = 1.
   3. POST localhost:8080/users 🡪 This should create and save a new user.
   4. GET localhost:8080/users/1/posts 🡪 This should retrieve the posts of the user where id = 1.

*Note: 1 is just an example, the actual path should ask for ‘id’*

1. Make a query that will return all the users that have more than 1 post

**Optional**

1. In a real application, the user should be logged in (authenticated). Assuming that the user is authenticated, make another version of this project that will have the same functionality, but will not ask for the ID, it should be auto-generated. The postman should accept a post, and the post will be mapped with the corresponding user.

Make sure you execute all these requests using postman and follow best practices and conventions..   
*You may refer to the* ***demo*** *project in Sakai 🡪 Resources 🡪 Demos 🡪 Lessons 6 and 7.zip*