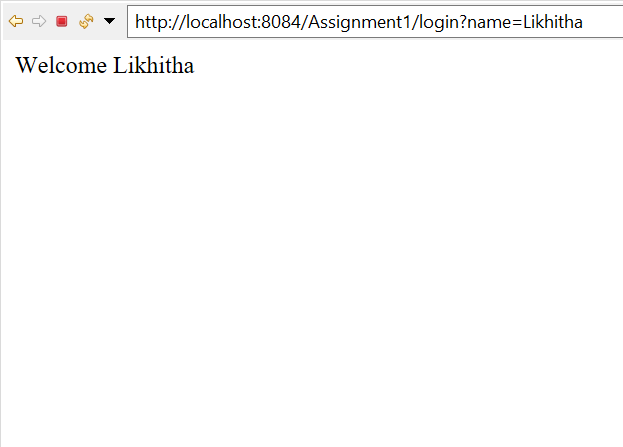
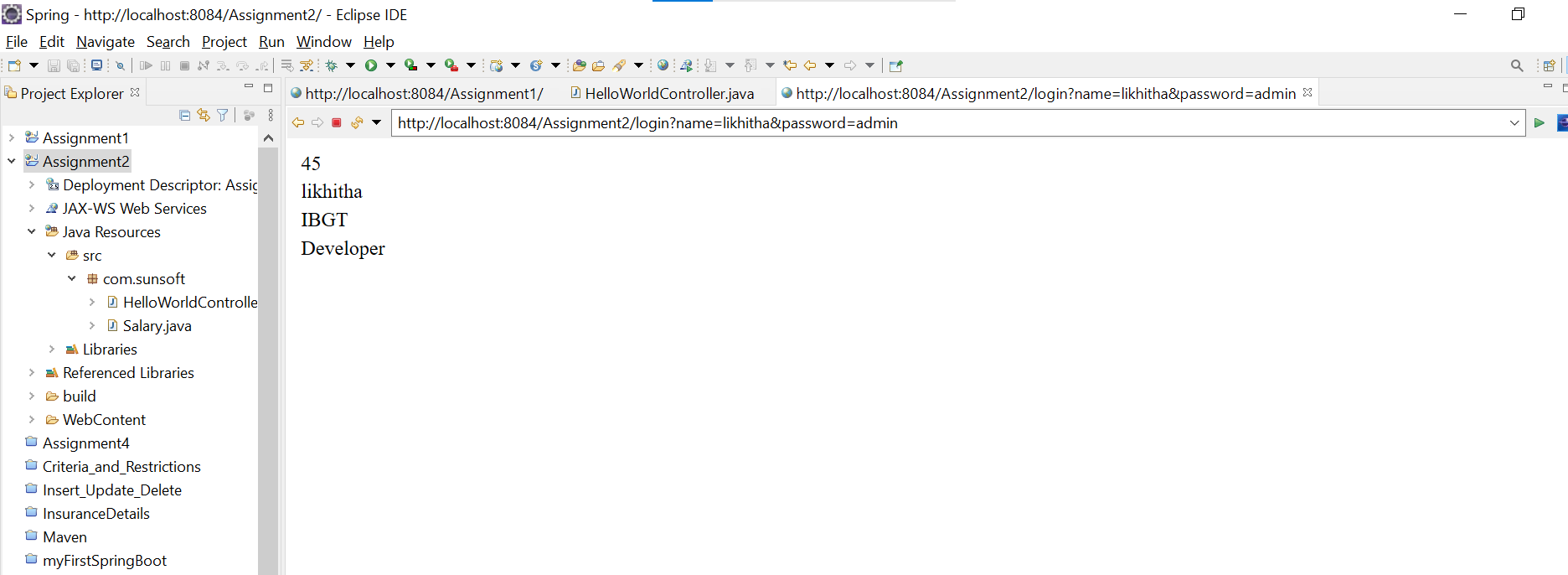
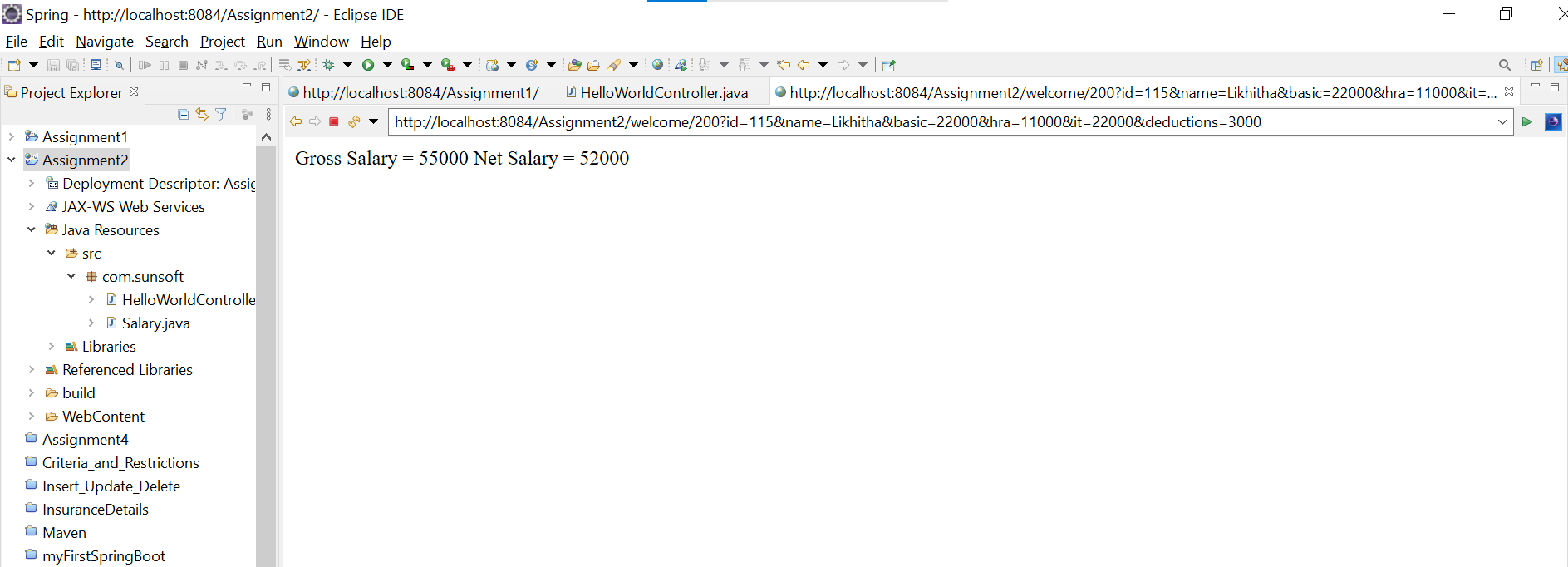
**Spring MVC**

1. Write Spring MVC code to accept your name in the edit box and display Welcome, given user name.****

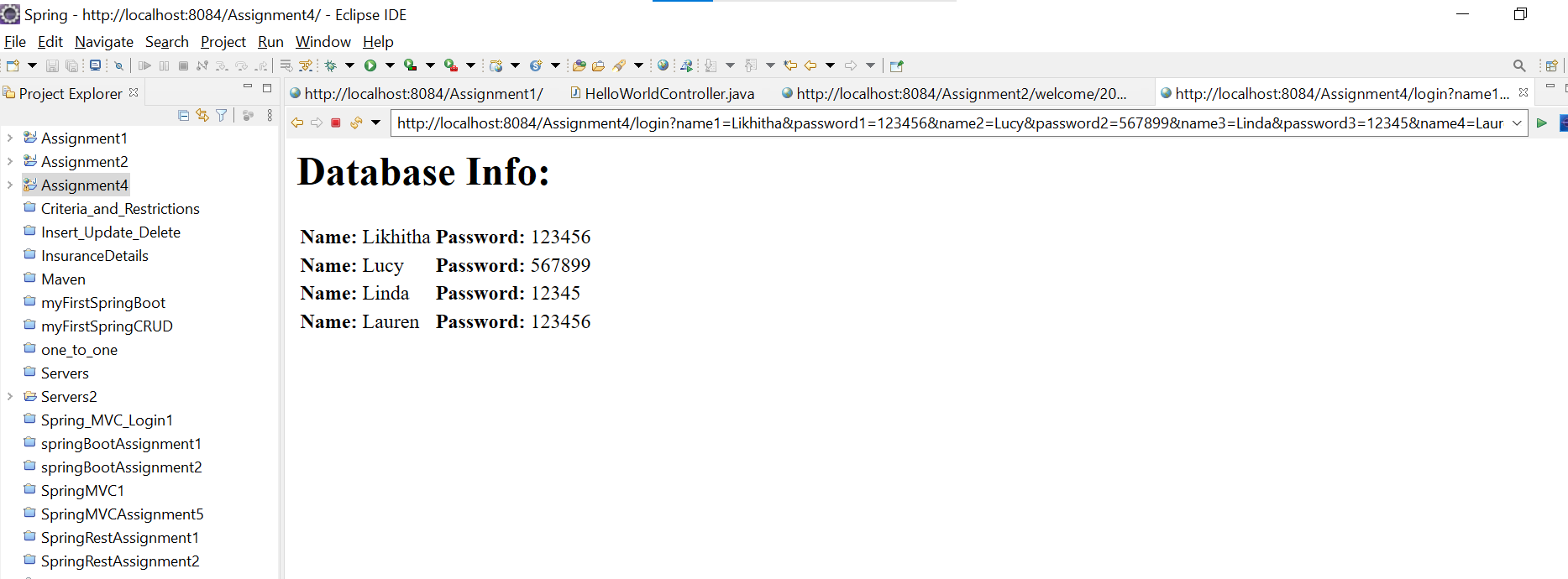
2. Write Spring MVC code, once the valid userid and password is entered then display the employee details in a table. Employee details will have Empid, name, dept and designation.



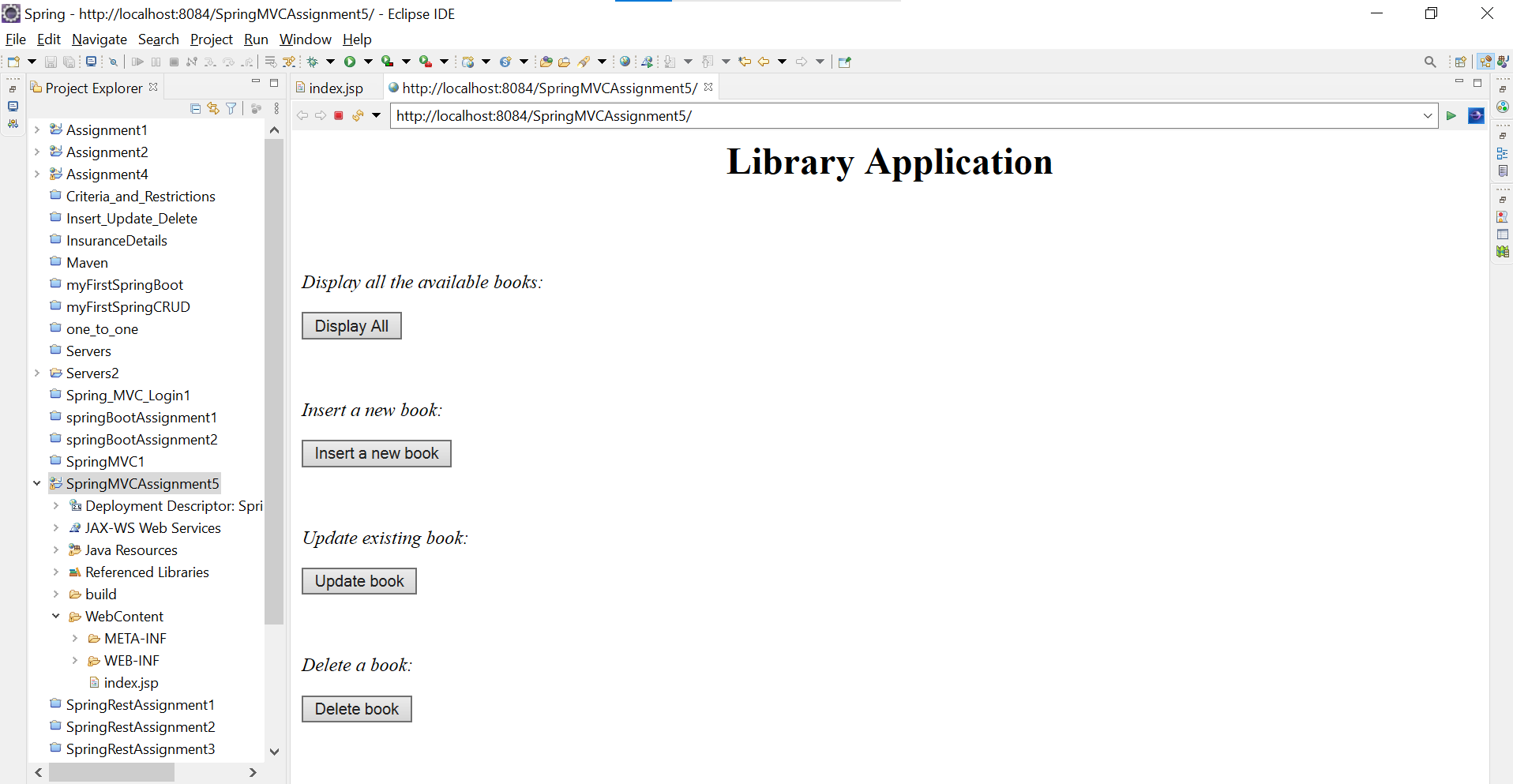
3. Pass employee salary details as part of URL. Details are EmpID, Name, basic, hra, da, IT and deducations. Calculate the gross salary and net salary and display the details only if the valid Employee id is given. If it is invalid Employee id, display the details in another page as invalid employee id is passed.



4. Write Spring MVC code to accept Insurance domain details and add them in the data structure. User can enter multiple records and can be inserted into this data structure. This data structure should be static, so that it can support multiple threads. Also have the option to display all the records of policy holders and their details.



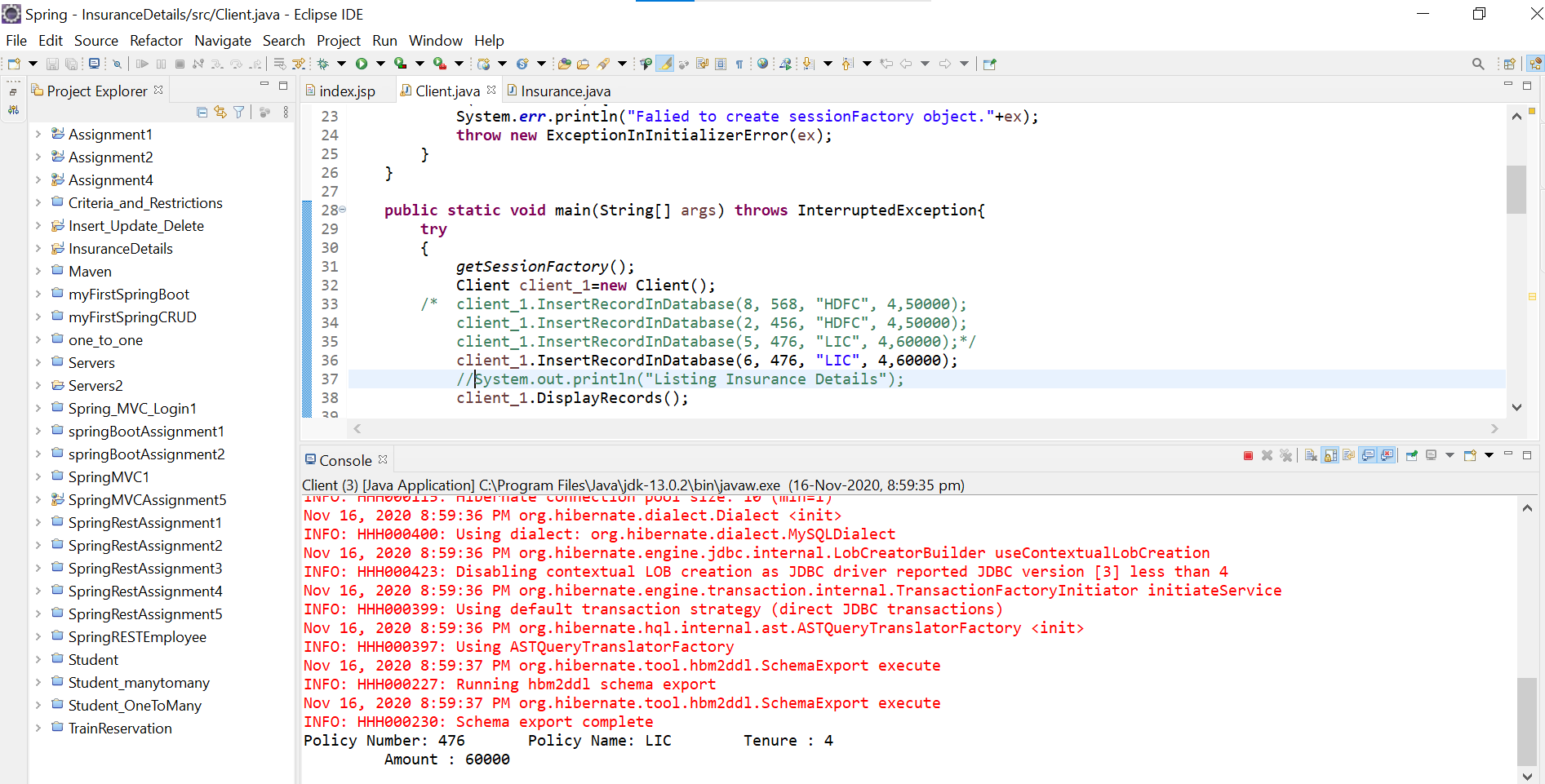
5. Write Spring MVC code to perform CRUD operations on any domain. You can either use the text file for this operation or you can use any data struture as static variable. As in static ArrayList<>.



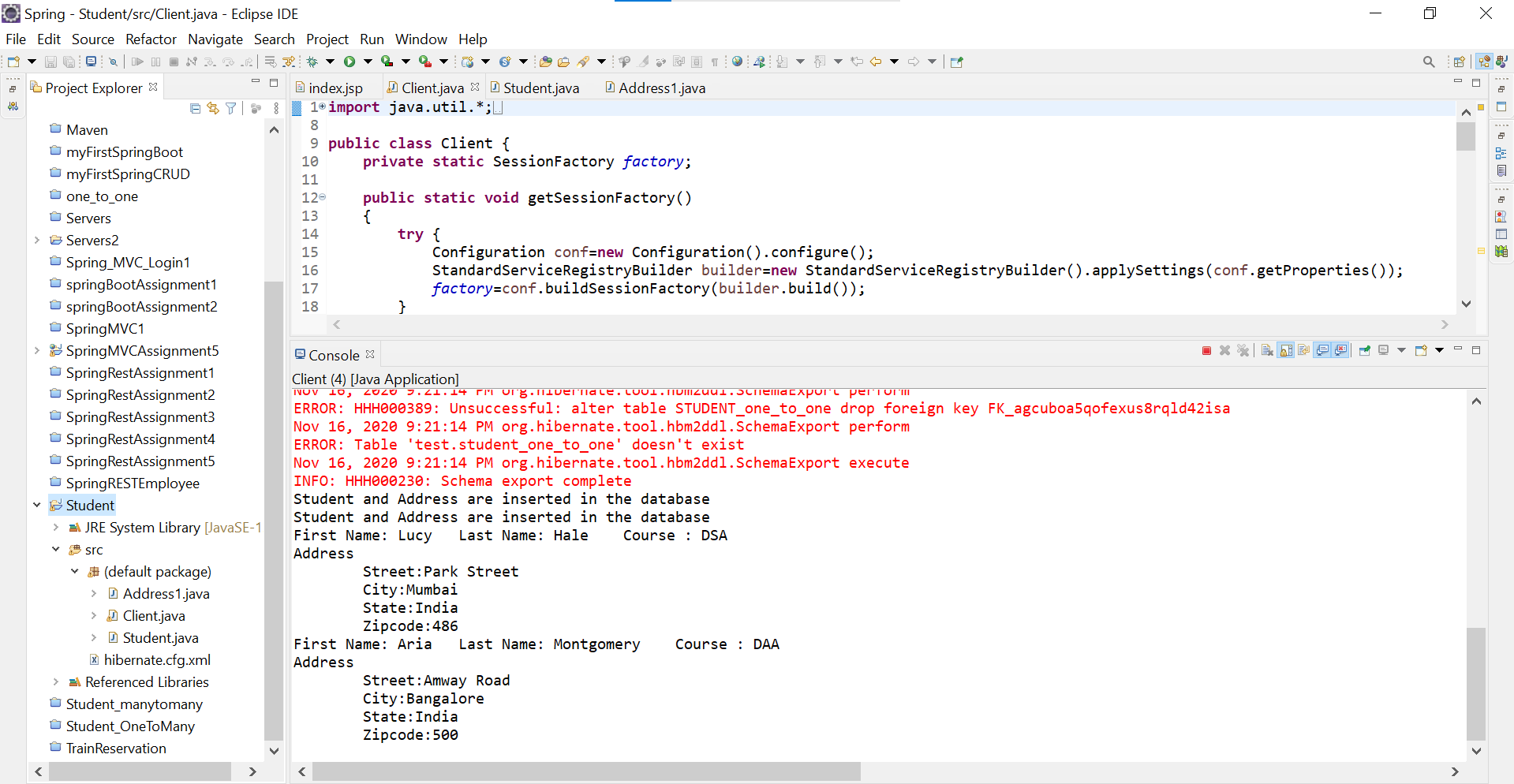


**Hibernate**

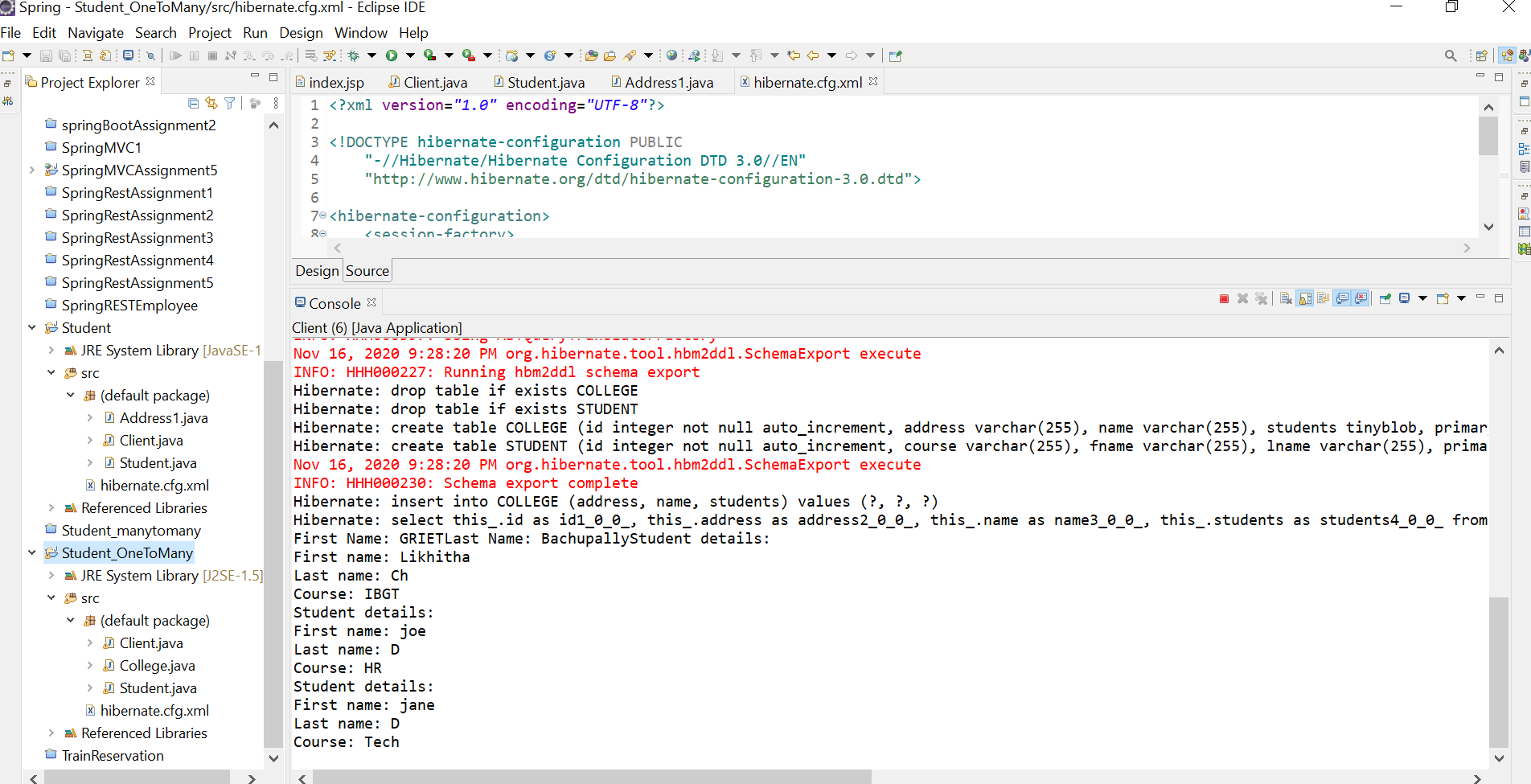
1. Write hibernate code to perform CRUD operations on Insurance Domain. Table and class will have the fields Policy No, Name, Tenure, amount to be paid. Initially write this in xml file and convert it into annotation using @Table, @id and @Column.



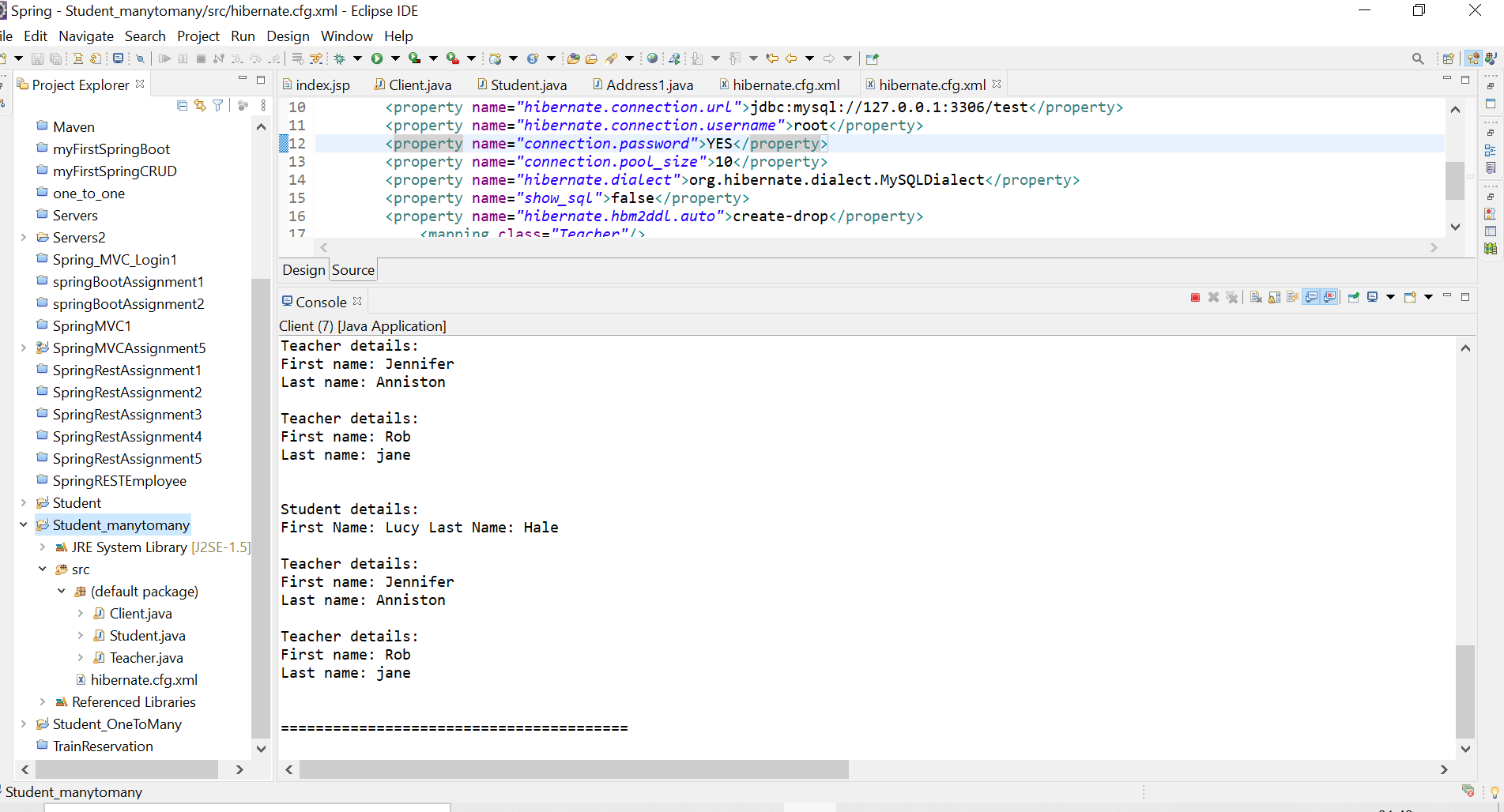
2. Write hibernate code to perform one to one relationship of Student domain. One student will have one address. Choose the student and address table fields appropriately. Use annotation @OneToOne for this.



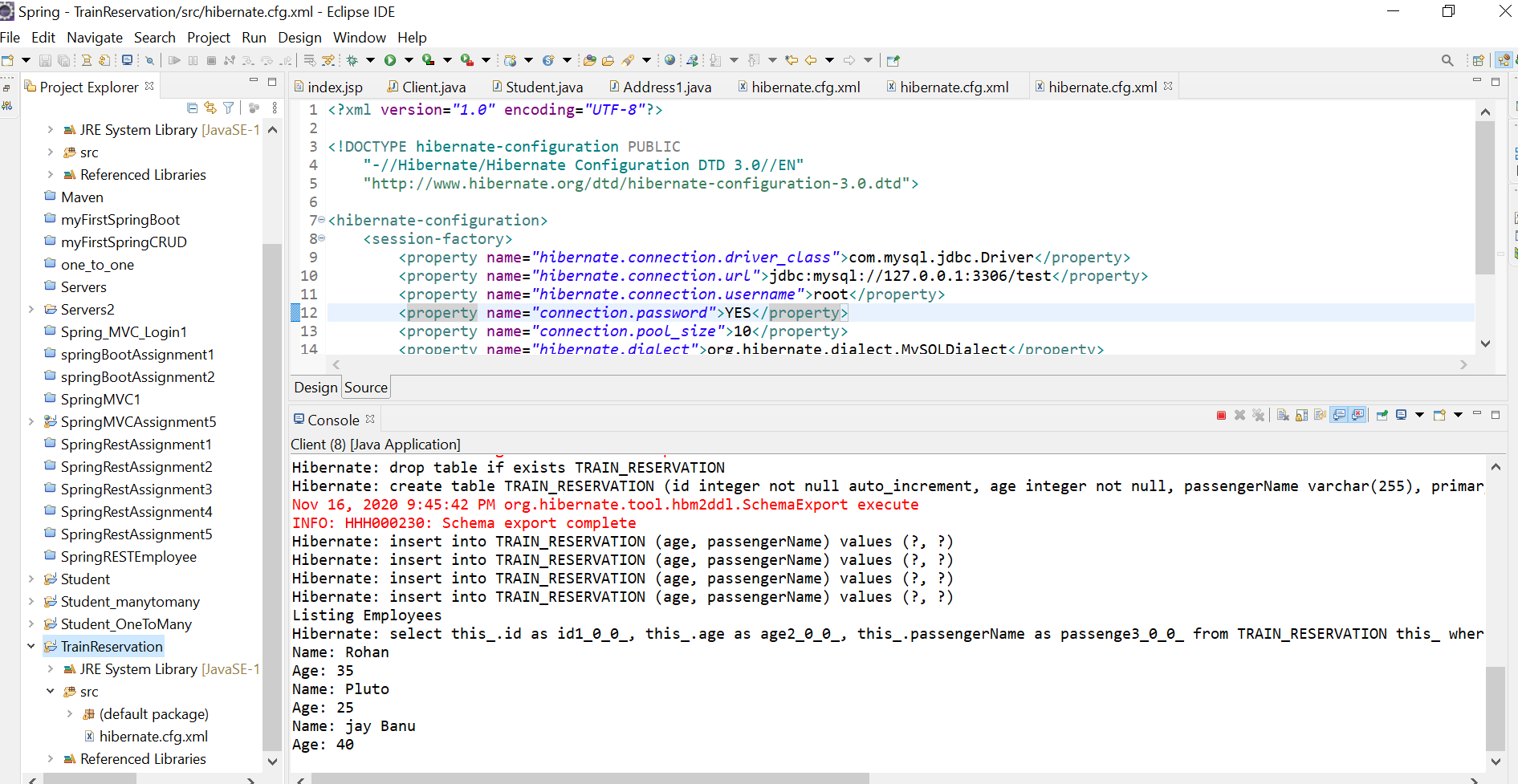
3. Write hibernate code to demonstrate one to many relationship on Student domain. One college can have many students. Use annotation for this.



4. Write hibernate code to demonstrate many to many relationships on student and teacher domain. One student can have many teachers and one teacher can teach many students.

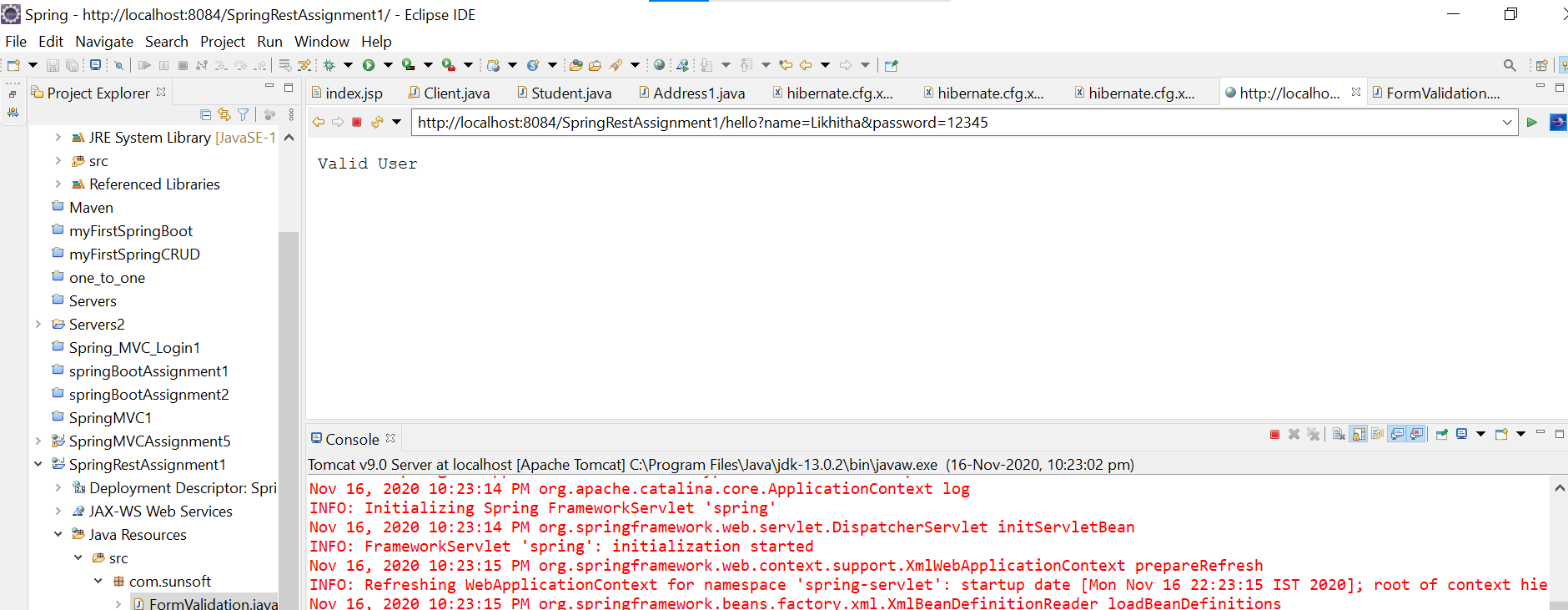


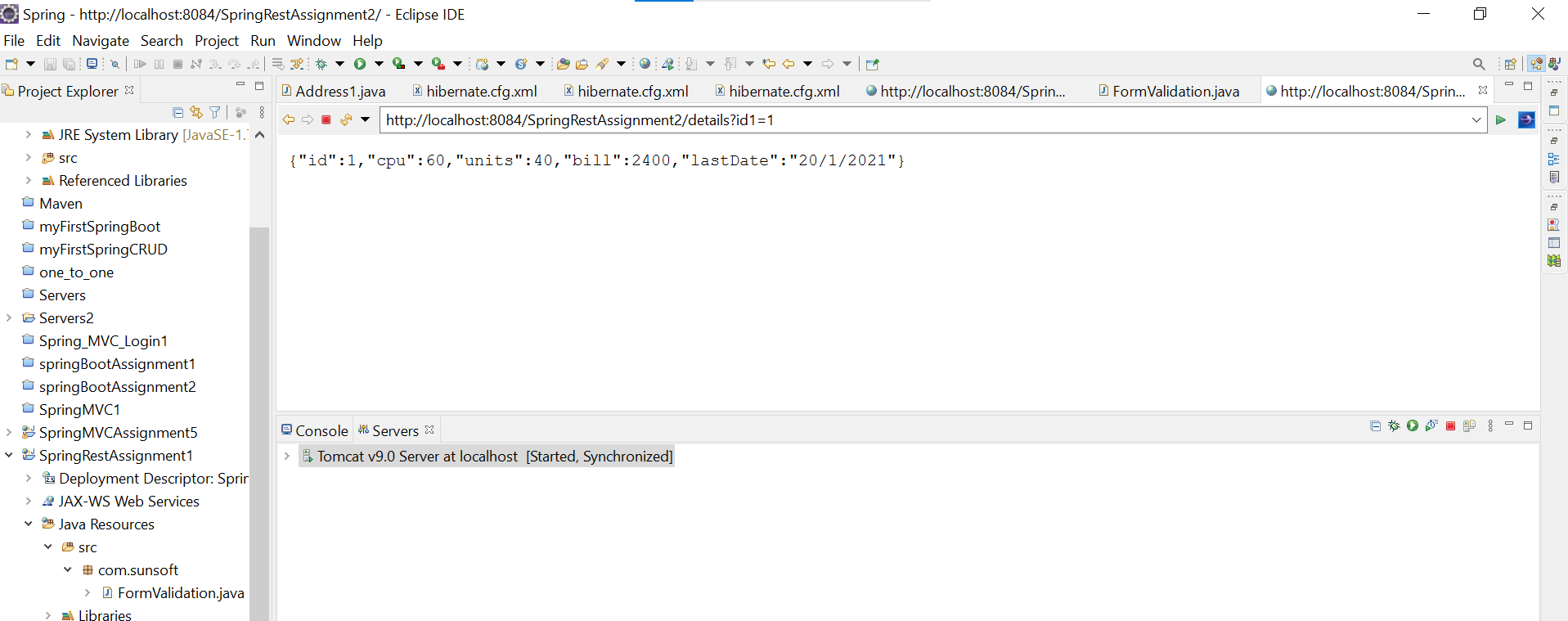
5. Write hibernate code to demonstrate criteria and restrictionson Train reservation domain. Display all the passengers whose age is between 25 to 45. Choose the passenger fields appropriately



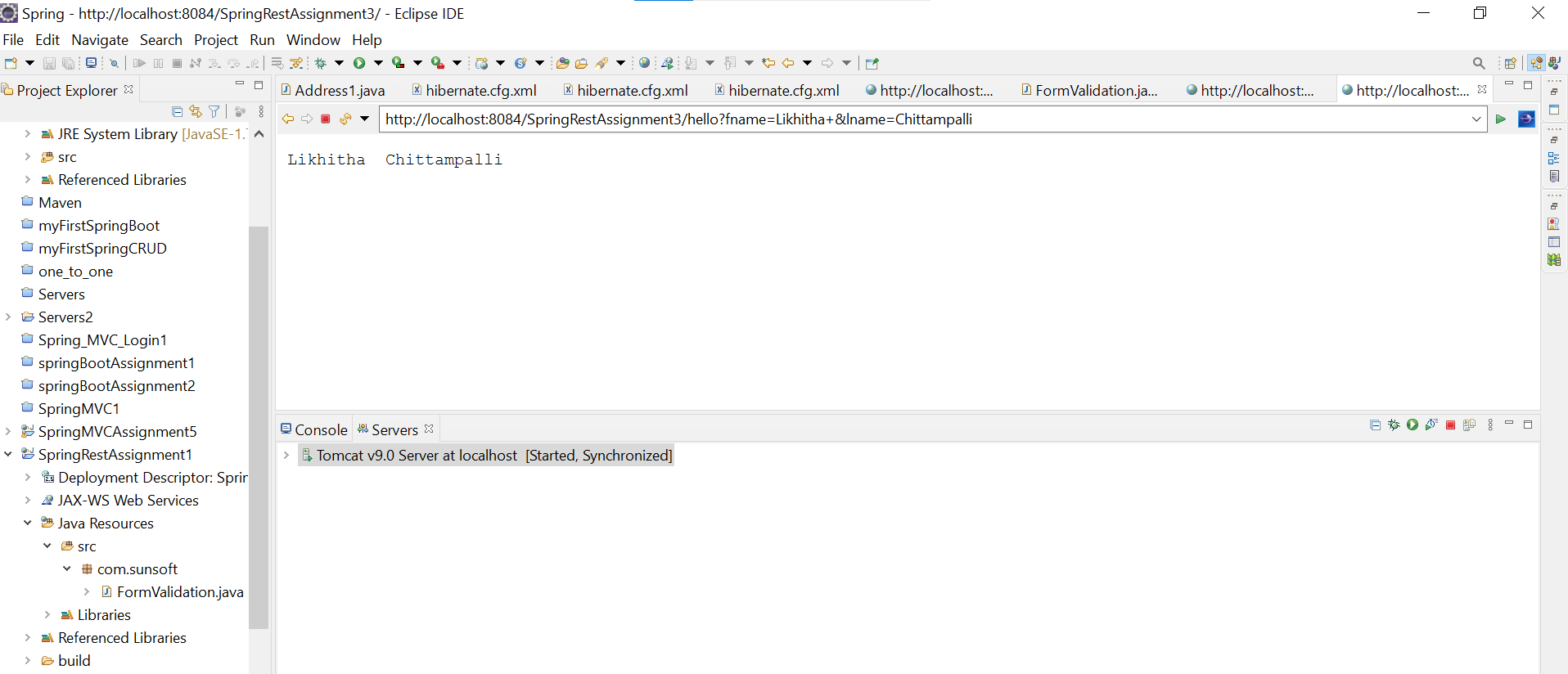
**Spring REST**

1. Develop REST application which takes the user id and password and returns back whether it is a valid user or not.

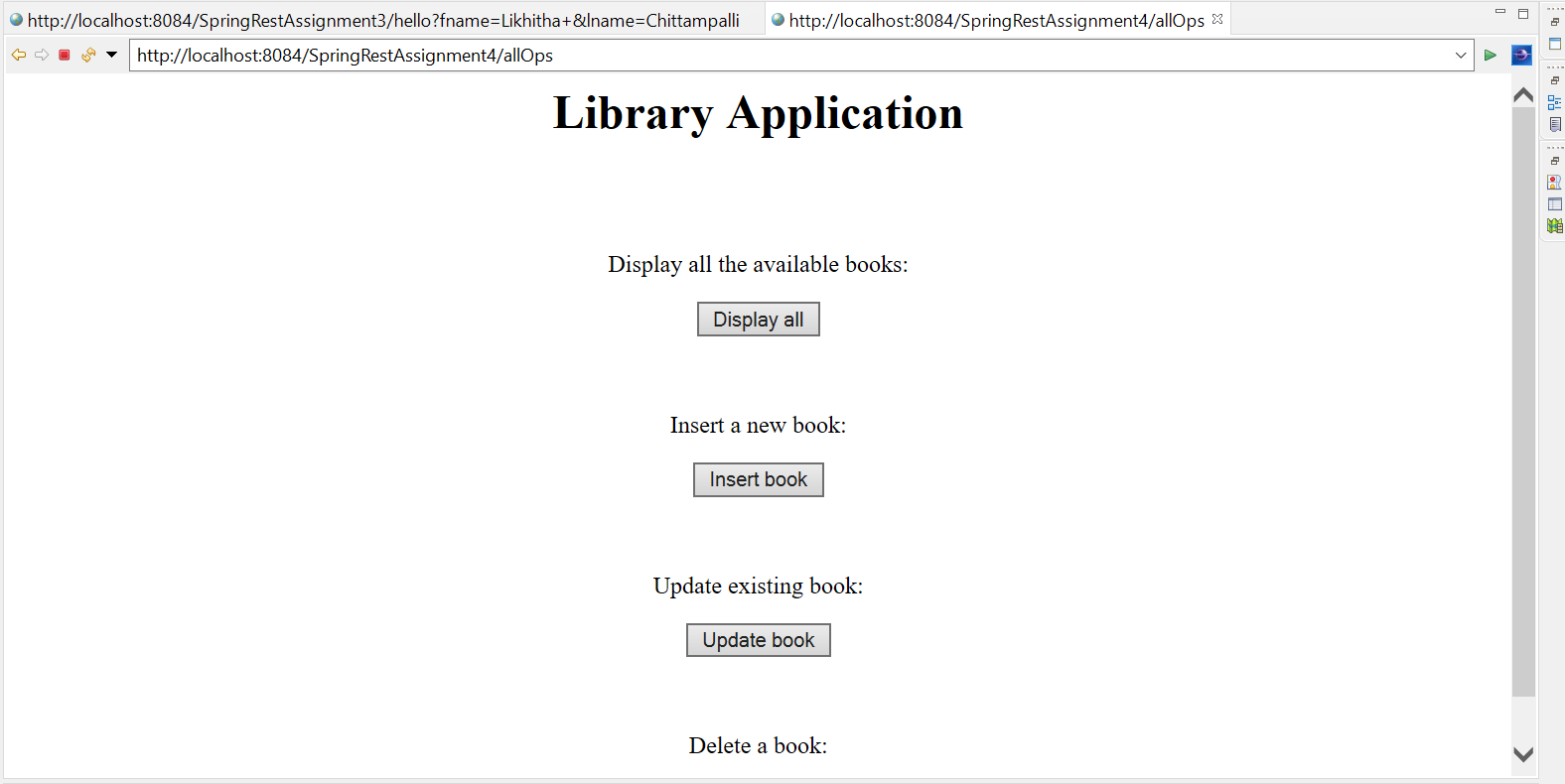
 2. Develop REST application which takes the Electricity meter number and will display the details about the electricity consumed in that month. Decide on the fields to be returned back to the user. A static datastructure can be maintained with the electricity meter no. Create a class with electricity meter no as the id and details based on it. Also include the fields monthly consumed units, cost per unit and how much customer has to pay and when is the laste date pay. Return this data in json format



3. Write a REST application to take the first name and last name and return full name.



4. Write the REST application for Library. When /display\_all is given then display all the books in the library. Decide the attributes for the library books. When /display and the book no is given then display only the book asked for. Also perform Insert, update and delete operation on books of the library. Initially static data structure can be used for database. Once all the assigments are done, this program should be converted to perform CRUD operations on the database. Connect REST with hibernate to perform these operations.



5. Write the REST application to validate the Login credentials (UserId and Password) using POST method. REST Controller will have 3 users. Client data is validated with this data and will respond as a valid user or invalid user to the client. Test this application using Postman

