```
 1
 2     DEMOS : Try out the Helloworld/AOP/Course-api Demos
 3
 4     ----------------------------------------------------------
 5
 6     Ex-1.
 7     - Create a RESTful CRUD Application for
 8     - Product (id->int,name->string,description->string)
 9     - Create the model/daos/services/controllers as discussed during theory sessions
10     - Simulate CRUD Operations on the List of Products.
11     ----------------------------------------------------------------------
12
13     DEMOS : Try out the course-api-jdbc
14
15     Ex-2.
16     - Create a RESTful CRUD Application for
17     - Project (id->int,name->string,duration->int, client->string)
18     - Create the model/daos/services/controllers as discussed during theory sessions
19     - Auto create the schema and sample data
20     ----------------------------------------------------------
21
22     Ex-3.
23     - Create a RESTful CRUD Application for
24     - Product (id->int,name->string,description->string,price->double)
25     - ShopCart(id->int,orderAmount->double ,products->List<Product>)
26     - Create the model/daos/services/controllers as discussed during theory sessions
27     - Auto create the schema and sample data
28     -------------------------------------------------------
29
30
31     Ex-4.
32     - Add the AOP support for before and after advices for the ProjectServiceImpl from Ex. 1
33     - before and after : Must log on console the method that is invoked along with the time
34     ----------------------------------------------------------
35
36     DEMOS :  Try out the EmpCrudService/Actuator/Swagger Demos
37
38     Ex-5.
39     - Create a RESTful CRUD Application using CrudRepository for
40     - Person(id->int auto generated, Country->String, age->int, bloodGroup->String)
41     - Create the model/daos/services/controllers as discussed during theory sessions
42     - Enable the actuator support for the Application monitoring
43     - Create swagger2 documentation for the PersonController
44     ----------------------------------------------------------------------
45
46     Ex-6.
47     - Create a RESTful CRUD Application for
48     - Customer (id->int,name->string,ssn->string,city->string)
49     - Create the model/daos/services/controllers as discussed during theory sessions
50     - Use the Hibernate ORM for the DB Operations
51
52     -------------------------------------------------------
53
54     DEMOS :  Try out the
       MvcJspApp/SpringSecurityMvcBasic/DemoMvcSecurityJdbc/SpringCloudConfigServer/SpringCloudC
       onfigClient Demos
55
56     Ex 7.
57     - Create a traditional MVC application with spring boot with following fuctionality
58     - index page shows links for add, view , update , delete , view all
59     - click on each of the link should perform the corresponding operation
60     for the BankAccount(accNo->int,accType->string,accHolderName->string,balance->double)
61     - Create the appropriate folder hierarchy and required models/daos/services/controllers
       as discussed in theory sessions.
62
63     -------------------------------------------------------
64
65     Ex-8.
66     - Implement the Custom form based JDBC authentication using spring boot for the MVC app
```

from Ex 6.
67 - Create two roles ADMIN , Customer
68 - add, update, delete functionalities should be allowed only to ADMIN role
69 - view, view all functionalities must be allowed to both the roles
70
71 ---
72
73 Ex-9.
74 - Standup the config server for picking up the hibernate properties for Ex-6. from git repo.
75 - Create a new Project for config server and only modify the Ex-6 project wherever required.