

Assignment 4



Student Name/ID Number:	Chathushi Jayarathna
Academic Year:	2022/23
Unit Assessor:	MS. Aravinder Kaur
Project Title:	Assignment 4 - ADP
Issue Date:	25/11/2022
Submission Date:	02/01/2023
Internal Verifier Name:	
Date:	02/01/2023

Learner declaration

I certify that the work submitted for this assignment is my own and research sources are fully acknowledged.

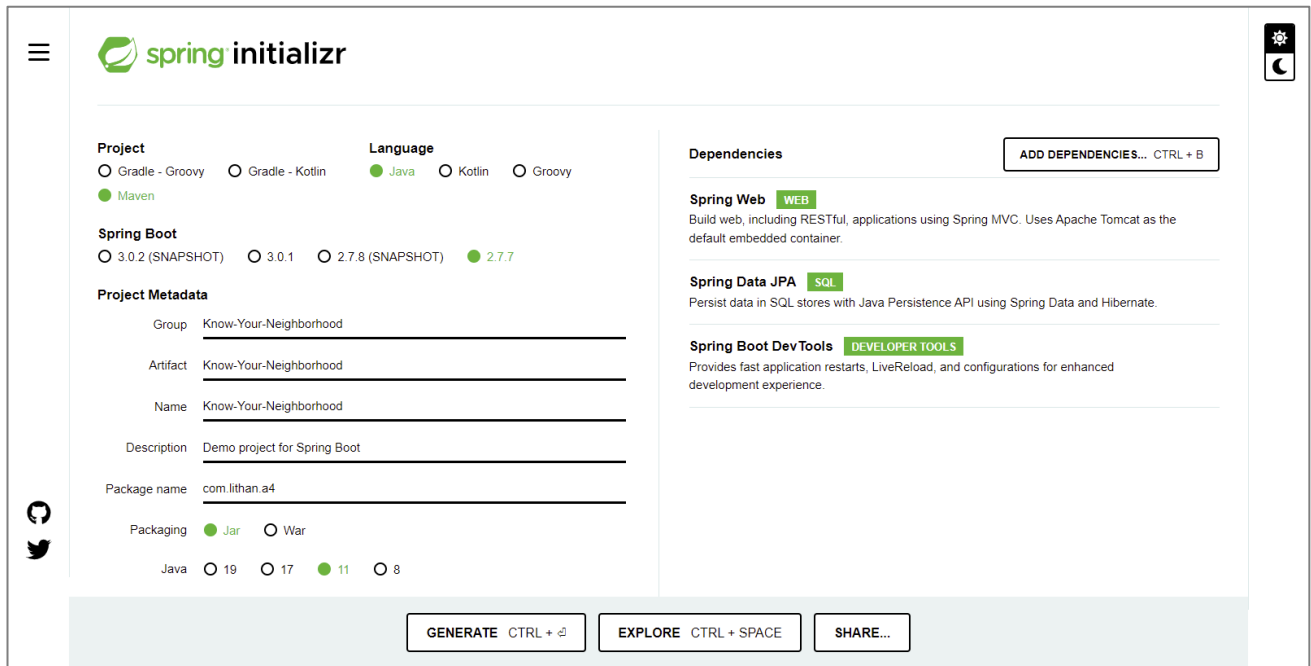
Student signature: **Chathushi**

Date: 02/01/2023

Module No:	1	IU No:	4	Exercise No.	1
-------------------	---	---------------	---	---------------------	---

Lab Assessment Statement	<p><u>Assignment 4 - Spring Boot + JPA</u></p> <p>Let's build on the previous assignment.</p> <ol style="list-style-type: none"> 1. Create a Spring Boot application for "Know-Your-Neighborhood". 2. Add support for JSP views and create required folder structure. 3. Move already developed classes into this project. 4. Create store table in the database 5. Annotate all the fields of Store object appropriately. 6. Develop all components required to view the stores <ol style="list-style-type: none"> a. Add method to existing Controller class to receive the request to fetch stores b. Enhance Store service and Store Controller objects to support CRUD operations on store object. c. d. Add method existing repository class to perform crud operations on Store object. e. Create HTML to view the stores. Show name, phone number and localities it serves for each store. f. Ensure that view stores request works end-to-end. (i.e., should be able to submit request to view the stores in the browser and get the page back with all stores). 7. Create an HTML page to perform Add store, Fetch stores, Update store, & Delete store operations from the UI.
Technical Environment	-
Guidelines	-
Duration	120 mins

1. Create a Spring Boot application for "Know-Your-Neighborhood".



The image shows the Spring Initializr web interface. On the left, there are navigation icons for a menu, a refresh, and a share. The main content area is divided into several sections: 'Project' with radio buttons for Gradle - Groovy, Gradle - Kotlin, Java (selected), Kotlin, and Groovy; 'Maven' (selected); 'Spring Boot' with radio buttons for 3.0.2 (SNAPSHOT), 3.0.1, 2.7.8 (SNAPSHOT), and 2.7.7 (selected); 'Project Metadata' with input fields for Group (Know-Your-Neighborhood), Artifact (Know-Your-Neighborhood), Name (Know-Your-Neighborhood), Description (Demo project for Spring Boot), and Package name (com.lithan.a4); 'Packaging' with radio buttons for Jar (selected) and War; and 'Java' with radio buttons for 19, 17, 11 (selected), and 8. On the right, there is a 'Dependencies' section with a button 'ADD DEPENDENCIES... CTRL + B'. It lists three dependencies: 'Spring Web' (WEB) with a description 'Build web, including RESTful, applications using Spring MVC. Uses Apache Tomcat as the default embedded container.'; 'Spring Data JPA' (SQL) with a description 'Persist data in SQL stores with Java Persistence API using Spring Data and Hibernate.'; and 'Spring Boot DevTools' (DEVELOPER TOOLS) with a description 'Provides fast application restarts, LiveReload, and configurations for enhanced development experience.' At the bottom, there are three buttons: 'GENERATE CTRL + G', 'EXPLORE CTRL + SPACE', and 'SHARE...'.

2. Add support for JSP views and create required folder structure.

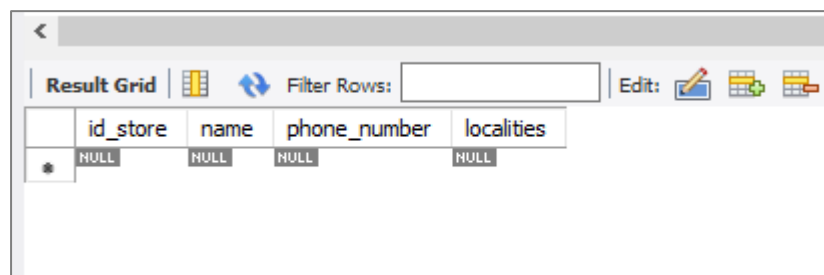
```
spring.mvc.view.prefix=/WEB-INF/  
spring.mvc.view.suffix=.jsp
```

3. Move already developed classes into this project.

```
src/main/java  
├── com.lithan.a4  
│   ├── KnowYourNeighborhoodApplication.java  
│   ├── com.lithan.a4.controller  
│   │   ├── StoreController.java  
│   ├── com.lithan.a4.entity  
│   │   ├── Store.java  
│   ├── com.lithan.a4.repository  
│   │   ├── StoreRepository.java  
│   ├── com.lithan.a4.service  
│   │   ├── StoreService.java  
│   │   └── StoreServiceImpl.java
```

4. Create store table in the database

```
1 • CREATE TABLE `test`.`tb_store` (  
2     `id_store` INT NOT NULL AUTO_INCREMENT,  
3     `name` VARCHAR(45) NOT NULL,  
4     `phone_number` INT NOT NULL,  
5     `localities` TEXT NOT NULL,  
6     PRIMARY KEY (`id_store`)  
7 ) ENGINE = InnoDB;
```



	id_store	name	phone_number	localities
*	NULL	NULL	NULL	NULL

5. Annotate all the fields of Store object appropriately.

Store.java

```
1 package com.lithan.a4.entity;  
2  
3 import javax.persistence.Column;  
4 import javax.persistence.Entity;  
5 import javax.persistence.GeneratedValue;  
6 import javax.persistence.GenerationType;  
7 import javax.persistence.Id;  
8 import javax.persistence.Table;  
9  
10 @Entity  
11 @Table(name = "tb_store")  
12 public class Store {  
13  
14     @Id  
15     @GeneratedValue(strategy = GenerationType.IDENTITY)  
16     @Column(name = "id_store")  
17     private int id_store;  
18  
19     @Column(name = "name")  
20     private String name;  
21  
22     @Column(name = "phone_number")  
23     private String phone_number;  
24  
25     @Column(name = "localities")  
26     private String localities;  
27
```

```
27
28 public Store() {
29
30 }
31
32 public Store(String name, String phone_number, String localities) {
33     this.name = name;
34     this.phone_number = phone_number;
35     this.localities = localities;
36 }
37
38 public String getName() {
39     return name;
40 }
41
42 public void setName(String name) {
43     this.name = name;
44 }
45
46 public String getPhone_number() {
47     return phone_number;
48 }
49
50 public void setPhone_number(String phone_number) {
51     this.phone_number = phone_number;
52 }
53
54 public String getLocalities() {
55     return localities;
56 }
57
58 public void setLocalities(String localities) {
59     this.localities = localities;
60 }
61
62 public int getId_store() {
63     return id_store;
64 }
65
66 public void setId_store(int id_store) {
67     this.id_store = id_store;
68 }
69
70 }
```

6. Develop all components required to view the stores

- a. Add method to existing Controller class to receive the request to fetch stores

storecontroller.java

```
@Controller
public class StoreController {

    @Autowired
    StoreService storeService;

    @GetMapping("/")
    public ModelAndView home() {
        ModelAndView mv = new ModelAndView();
        mv.setViewName("home");
        return mv;
    }
}
```

- b. Enhance Store service and Store Controller objects to support CRUD operations on store object.
- c. Add method existing repository class to perform crud operations on Store object.

storeservice.java

```
package com.lithan.a4.service;

import java.util.List;

import com.lithan.a4.entity.Store;

public interface StoreService {

    public void addStore(Store store);

    public Store findById(int id);

    public List<Store> listStore();

    public Store editStore(Store store);

    public void deleteStore(int id);
}
```

storeserviceImpl.java

```
1 package com.lithan.a4.service;
2
3 import java.util.List;
4
5 import org.springframework.beans.factory.annotation.Autowired;
6 import org.springframework.stereotype.Service;
7
8 import com.lithan.a4.entity.Store;
9 import com.lithan.a4.repository.StoreRepository;
10
11 @Service
12 public class StoreServiceImpl implements StoreService {
13
14     @Autowired
15     private StoreRepository storeRepo;
16
17     @Override
18     public void addStore(Store store) {
19
20         storeRepo.save(store);
21     }
22
23     @Override
24     public List<Store> listStore() {
25
26         List<Store> listStore = storeRepo.findAll();
27
28         return listStore;
29     }
30
31     @Override
32     public Store editStore(Store store) {
33         Store editedStore = storeRepo.findById(store.getId_store()).get();
34
35         editedStore.setName(store.getName());
36         editedStore.setPhone_number(store.getPhone_number());
37         editedStore.setLocalities(store.getLocalities());
38
39         storeRepo.save(editedStore);
40
41         return editedStore;
42     }
43
44     @Override
45     public void deleteStore(int id) {
46
47         storeRepo.deleteById(id);
48     }
49
50     @Override
51     public Store findById(int id) {
52         Store store = storeRepo.findById(id).get();
53
54         return store;
55     }
56 }
57
```


storeController.java

```
17 @Controller
18 public class StoreController {
19
20     @Autowired
21     StoreService storeService;
22
23     @GetMapping("/")
24     public ModelAndView home() {
25         ModelAndView mv = new ModelAndView();
26         mv.setViewName("home");
27         return mv;
28     }
29
30     @GetMapping("/store")
31     public ModelAndView viewStore() {
32         ModelAndView mv = new ModelAndView("store");
33
34         List<Store> stores = storeService.listStore();
35
36         mv.addObject("stores", stores);
37
38         return mv;
39     }
40
41     // Add Store
42     @GetMapping("/addStore")
43     public String addStore(Model model) {
44         Store store = new Store();
45
46         model.addAttribute("store", store);
47
48         return "add-store";
49     }
50
51     @PostMapping("/saveAddStore")
52     public String saveAddStore(@ModelAttribute("store") Store store) {
53
54         storeService.addStore(store);
55
56         return "redirect:/store";
57     }
58
59     // Edit Store
60     @GetMapping("/editStore")
61     public String editStore(@RequestParam("id") int id, Model model) {
62
63         Store store = storeService.findById(id);
64
65         model.addAttribute("store", store);
66
67         return "edit-store";
68     }
69
70     @PostMapping("/saveEditStore")
71     public String saveEditStore(@ModelAttribute("store") Store store) {
72
73         storeService.editStore(store);
74
75         return "redirect:/store";
76     }
77
78     // Delete Store
79     @GetMapping("/deleteStore")
80     public String saveEditStore(@RequestParam("id") int id) {
81
82         storeService.deleteStore(id);
83
84         return "redirect:/store";
85     }
86
87 }
88
```

- d. Create HTML /JSP to view the stores. Show name, phone number and localities it serves for each store.

[Add Store](#)

Store Name	Phone Number	Localities
Test	1234567890	Test

- e. Ensure that view stores request works end-to-end.

[View Store](#)

Add Store

Store Name

Phone Number



Localities

[Add Store](#)

Store Name	Phone Number	Localities
Test	1234567890	Test
Virtusa	+94098765432	Kandy

Database – Before

Result Grid






Filter Rows:

	id_store	name	phone_number	localities
▶	1	Test	1234567890	Test
•	NULL	NULL	NULL	NULL

Database - After

Result Grid

Filter Rows:

	id_store	name	phone_number	localities
	1	Test	1234567890	Test
	2	Virtusa	+94098765432	Kandy

7. Create an HTML page to perform Add store, Fetch stores, Update store, & Delete store operations from the UI.

✓ **Add Store**

Add Store

Store Name

Phone Number

Localities

Submit

```

<%@ include file="jsp-tags.jsp" %>
<!DOCTYPE html>
<html>
  <head>
    <meta charset="ISO-8859-1" />
    <title>Know Your Neighborhood</title>
    <style type="text/css">
      label,
      input {
        display: block;
      }
    </style>
  </head>
  <body>
    <a href="<%= request.getContextPath() %>/store">View Store</a>

    <form:form modelAttribute="store" action="saveAddStore" method="POST">
      <h1>Add Store</h1>
      <label>Store Name</label>
      <form:input type="text" path="name" />

      <label>Phone Number</label>
      <form:input type="text" path="phone_number" />

      <label>Localities</label>
      <form:input type="text" path="localities" />

      <button type="submit">Submit</button>
    </form:form>
  </body>
</html>

```

✓ View Store

```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="ISO-8859-1" />
    <title>Know Your Neighborhood</title>
  </head>
  <body>
    <a href="<%= request.getContextPath() %>/addStore">Add Store</a>

    <table border="1" style="margin-top: 20px">
      <tr>
        <th>Store Name</th>
        <th>Phone Number</th>
        <th>Localities</th>
      </tr>
      <c:forEach items="${stores}" var="store">
        <tr>
          <td>${store.name}</td>
          <td>${store.phone_number}</td>
          <td>${store.localities}</td>
        </tr>
      </c:forEach>
    </table>
  </body>
</html>

```

Add Store

Store Name	Phone Number	Localities
Test	1234567890	Test
Virtusa	+94098765432	Kandy

✓ Update Store

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="ISO-8859-1" />
    <title>Know Your Neighborhood</title>
    <style type="text/css">
      label,
      input {
        display: block;
      }
    </style>
  </head>
  <body>
    <a href="<%= request.getContextPath() %>/store">View Store</a>

    <form:form modelAttribute="store" action="saveEditStore" method="POST">
      <h1>Edit Store</h1>
      <label>Store Name</label>
      <form:input type="text" path="name" />

      <label>Phone Number</label>
      <form:input type="text" path="phone_number" />

      <label>Localities</label>
      <form:input type="text" path="localities" />

      <form:hidden path="id_store" />

      <button type="submit">Submit</button>
    </form:form>
  </body>
</html>
```

Edit Store

Store Name

Phone Number

Localities

Submit

EDIT STORE – Before Edit

[Add Store](#)

Store Name	Phone Number	Localities	Action
Test	1234567890	Test	Edit Store Delete Store
Virtusa	+94098765432	Kandy	Edit Store Delete Store

Result Grid




Filter Rows:


	id_store	name	phone_number	localities
	1	Test	1234567890	Test
	2	Virtusa	+94098765432	Kandy

EDIT STORE – After Edit

Store Name	Phone Number	Localities	Action
Edit	1234567890	Edit	Edit Store Delete Store
Virtusa	+94098765432	Kandy	Edit Store Delete Store

Result Grid





Filter Rows:


	id_store	name	phone_number	localities
	1	Edit	1234567890	Edit
	2	Virtusa	+94098765432	Kandy


DELETE STORE – Before Delete

[Add Store](#)

Store Name	Phone Number	Localities	Action
Test	1234567890	Test	Edit Store Delete Store
Virtusa	+94098765432	Kandy	Edit Store Delete Store

Result Grid





Filter Rows:



	id_store	name	phone_number	localities
	1	Test	1234567890	Test
	2	Virtusa	+94098765432	Kandy

DELETE STORE – After Delete



[Add Store](#)

Store Name	Phone Number	Localities	Action
Test	1234567890	Test	Edit Store Delete Store

Result Grid

Filter Rows:

	id_store	name	phone_number	localities
	1	Test	1234567890	Test
	NULL	NULL	NULL	NULL

Source Code



Assignment 4.rar

