

# Assignment -3

<b>Student Name/ID Number:</b>	Chathushi Jayarathna
<b>Academic Year:</b>	2022/23
<b>Unit Assessor:</b>	MS. Aravinder Kaur
<b>Project Title:</b>	Assignment 3 - ADP
<b>Issue Date:</b>	24/11/2022
<b>Submission Date:</b>	02/01/2023
<b>Internal Verifier Name:</b>	
<b>Date:</b>	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: 02/01/2022

<b>Module No:</b>	1	<b>IU No:</b>	3	<b>Exercise No.</b>	1
-------------------	---	---------------	---	---------------------	---

<b>Lab Assessment Statement</b>	<p><b><u>Assignment 4 - Spring Boot</u></b></p> <p>Let's build on the previous assignment.</p> <ol style="list-style-type: none"> <li>1. Create a Spring Boot application for "Know-Your-Neighborhood".</li> <li>2. Add support for JSP views and create required folder structure.</li> <li>3. Move already developed classes into this project.</li> <li>4. Develop all components required to view the stores <ol style="list-style-type: none"> <li>a. Add method to existing Controller class to receive the request to fetch stores</li> <li>b. Add method to existing Service class to process the request (or add method to existing service class)</li> <li>c. Add method existing repository class to return all available stores.</li> <li>d. Create HTML to view the stores. Show name, phone number and localities it serves for each store.</li> <li>e. 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).</li> </ol> </li> <li>5. Create an HTML page to add a store and link it to the view stores page.</li> <li>6. Let the response of add request submission be the view page that shows all stores including recently added store.</li> </ol>
<b>Technical Environment</b>	-
<b>Guidelines</b>	-
<b>Duration</b>	120 mins

Let's build on the previous assignment.

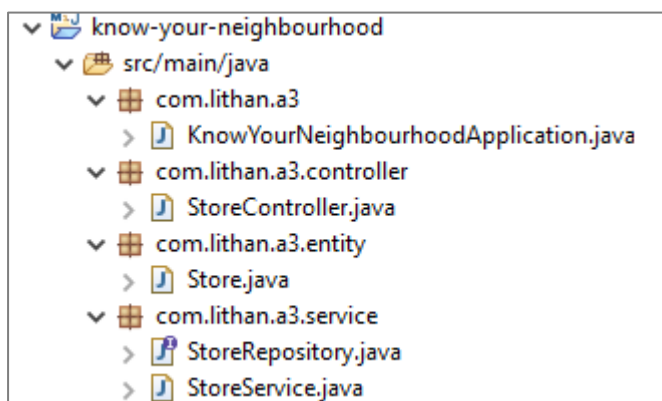
## 1. Create a Spring Boot application for "Know-Your-Neighbourhood"

The screenshot shows the Spring Initializr web application. The 'Project' section has 'Maven' selected. The 'Language' section has 'Java' selected. The 'Spring Boot' section has '2.7.7' selected. The 'Project Metadata' section shows the following fields: Group (know-your-neighborhood), Artifact (know-your-neighborhood), Name (know-your-neighborhood), Description (Demo project for Spring Boot), and Package name (com.lithan.a3). The 'Packaging' section has 'Jar' selected. The 'Dependencies' section has 'Spring Web' selected. At the bottom, there are buttons for 'GENERATE', 'EXPLORE', and 'SHARE...'. The 'GENERATE' button has a tooltip that says 'CTRL + G'. The 'EXPLORE' button has a tooltip that says 'CTRL + SPACE'. The 'SHARE...' button has a tooltip that says 'CTRL + S'.

## 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



## 4. Develop all components required to view the stores

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

```
1 package com.lithan.a3.controller;
2
3 import java.util.List;
4
5 import org.springframework.beans.factory.annotation.Autowired;
6 import org.springframework.stereotype.Controller;
7 import org.springframework.ui.Model;
8 import org.springframework.web.bind.annotation.GetMapping;
9 import org.springframework.web.bind.annotation.ModelAttribute;
10 import org.springframework.web.bind.annotation.PostMapping;
11 import org.springframework.web.servlet.ModelAndView;
12
13 import com.lithan.a3.entity.Store;
14 import com.lithan.a3.service.StoreRepository;
15
16 @Controller
17 public class StoreController {
18
19     @Autowired
20     StoreRepository storeService;
21
22     @GetMapping("/store")
23     public ModelAndView viewStore() {
24         ModelAndView mv = new ModelAndView("store");
25
26         List<Store> stores = storeService.listAll();
27
28         mv.addObject("stores", stores);
29
30         return mv;
31     }
32
33     @GetMapping("/addStore")
34     public String addStore(Model model) {
35         Store store = new Store();
36
37         model.addAttribute("store", store);
38
39         return "add-store";
40     }
41
42     @PostMapping("/saveAddStore")
43     public String saveAddStore(@ModelAttribute("store") Store store) {
44
45         storeService.save(store);
46
47         return "redirect:/store";
48     }
49 }
50
```

**b. Add method to existing Service class to process the request (or add method to existing service class)**

```
package com.lithan.a3.service;

import java.util.ArrayList;
import java.util.List;

import org.springframework.stereotype.Service;

import com.lithan.a3.entity.Store;

@Service
public class StoreService implements StoreRepository {

    private List<Store> stores = new ArrayList<Store>();

    @Override
    public void save(Store store) {
        stores.add(store);
    }

    @Override
    public List<Store> listAll() {
        return stores;
    }

}
```

**c. Add method existing repository class to return all available stores.**

```
package com.lithan.a3.service;

import java.util.List;

import com.lithan.a3.entity.Store;

public interface StoreRepository {

    public void save(Store store);

    public List<Store> listAll();

}
```

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

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1" isELIgnored="false"%>
<%@ taglib prefix="form" uri="http://www.springframework.org/tags/form" %>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<!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>
```

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

[Add Store](#)

Store Name	Phone Number	Localities
Test	1234567890	Test

## 5. Create an HTML page to add a store and link it to the view stores page.

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1" isELIgnored="false"%>
<%@ taglib prefix="form" uri="http://www.springframework.org/tags/form" %>
<!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](#)

## Add Store

Store Name

Phone Number

Localities

6. Let the response of add request submission be the view page that shows all stores including recently added store.

[Add Store](#)

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

### Source Code



Assignment 3.rar



