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

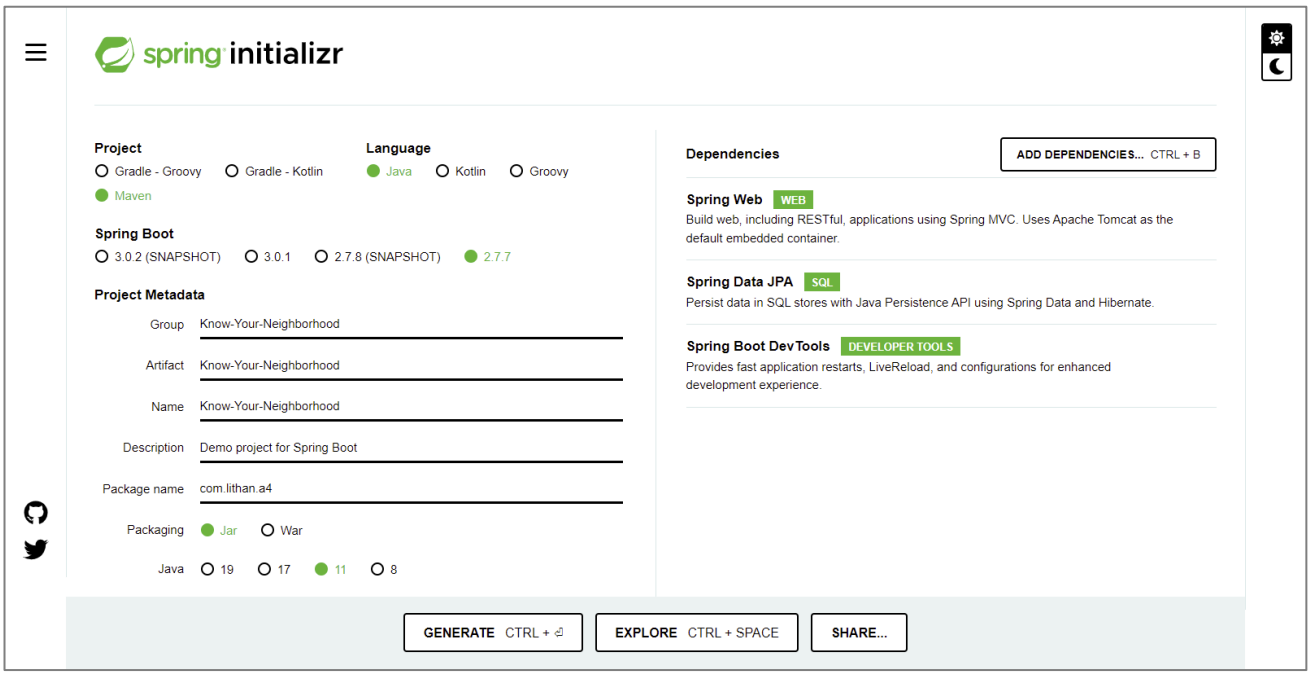
Learner declaration
I certify that the work submitted for this assignment is my own and research sources are fully acknowledged. Student signature: **Chathushi** Date: 02/01/2023

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

Lab Assessment Statement	**Assignment 4 - Spring Boot + JPA**
	
	Let's build on the previous assignment.
	
	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**
	   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

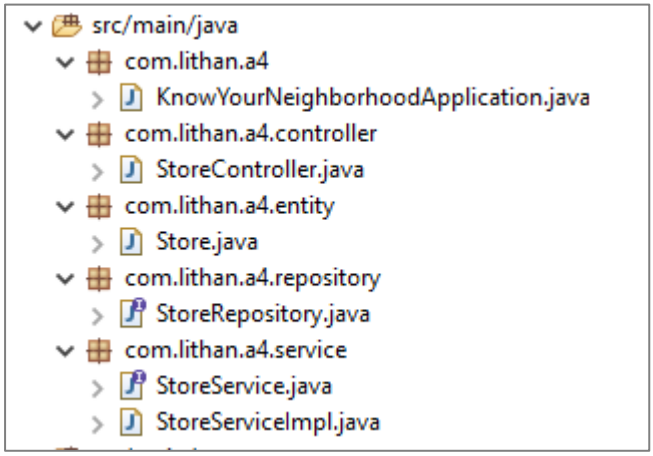
1. Create a Spring Boot application for “Know-Your-Neighborhood”.



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

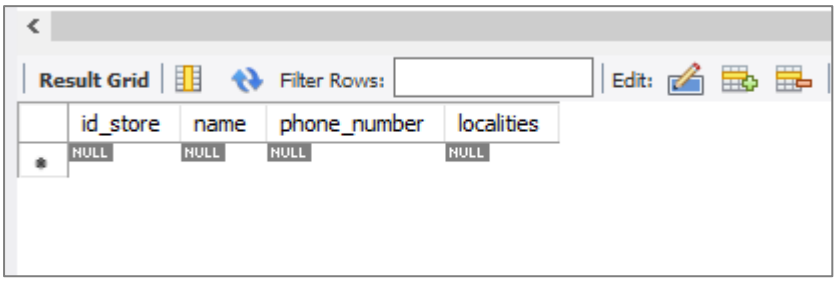
```
spring.mvc.view.prefix=/WEB-INF/
spring.mvc.view.suffix=.jsp
```

3. Move already developed classes into this project.



4. Create store table in the database

```
CREATE TABLE `test`.`tb_store` (
  `id_store` INT NOT NULL AUTO_INCREMENT,
  `name` VARCHAR(45) NOT NULL,
  `phone_number` INT NOT NULL,
  `localities` TEXT NOT NULL,
  PRIMARY KEY (`id_store`)
  ) ENGINE = InnoDB;
```



5. Annotate all the fields of Store object appropriately.

Store.java

```
package com.lithan.a4.entity;

import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.Table;

@Entity
@Table(name = "tb_store")
public class Store {

  @Id
  @GeneratedValue(strategy = GenerationType.IDENTITY)
  @Column(name = "id_store")
  private int id_store;

  @Column(name = "name")
  private String name;

  @Column(name = "phone_number")
  private String phone_number;

  @Column(name = "localities")
  private String localities;
```

```
  public Store() {

  }

  public Store(String name, String phone_number, String localities) {
    this.name = name;
    this.phone_number = phone_number;
    this.localities = localities;
  }

  public String getName() {
    return name;
  }

  public void setName(String name) {
    this.name = name;
  }

  public String getPhone_number() {
    return phone_number;
  }

  public void setPhone_number(String phone_number) {
    this.phone_number = phone_number;
  }

  public String getLocalities() {
    return localities;
  }

  public void setLocalities(String localities) {
    this.localities = localities;
  }

  public int getId_store() {
    return id_store;
  }

  public void setId_store(int id_store) {
    this.id_store = id_store;
  }

}
```

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);
}
```

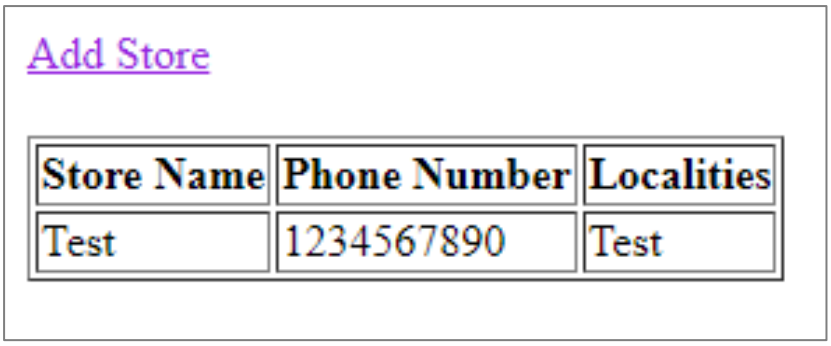
storeserviceImpl.java

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

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

import com.lithan.a4.entity.Store;
import com.lithan.a4.repository.StoreRepository;

@Service
public class StoreServiceImpl implements StoreService {

  @Autowired
  private StoreRepository storeRepo;

  @Override
  public void addStore(Store store) {

    storeRepo.save(store);
  }

  @Override
  public List<Store> listStore() {

    List<Store> listStore = storeRepo.findAll();

    return listStore;
  }

  @Override
  public Store editStore(Store store) {
    Store editedStore = storeRepo.findById(store.getId_store()).get();

    editedStore.setName(store.getName());
    editedStore.setPhone_number(store.getPhone_number());
    editedStore.setLocalities(store.getLocalities());

    storeRepo.save(editedStore);

    return editedStore;
  }

  @Override
  public void deleteStore(int id) {

    storeRepo.deleteById(id);
  }

  @Override
  public Store findById(int id) {
    Store store = storeRepo.findById(id).get();

    return store;
  }
}

```

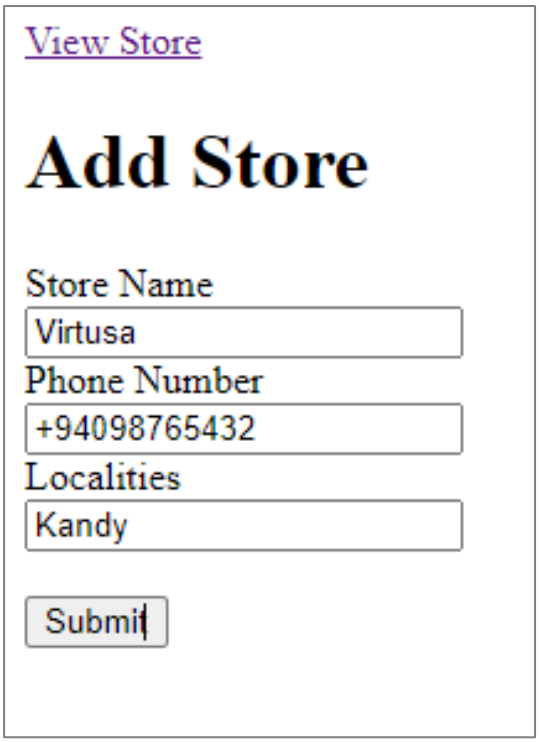
storeController.java

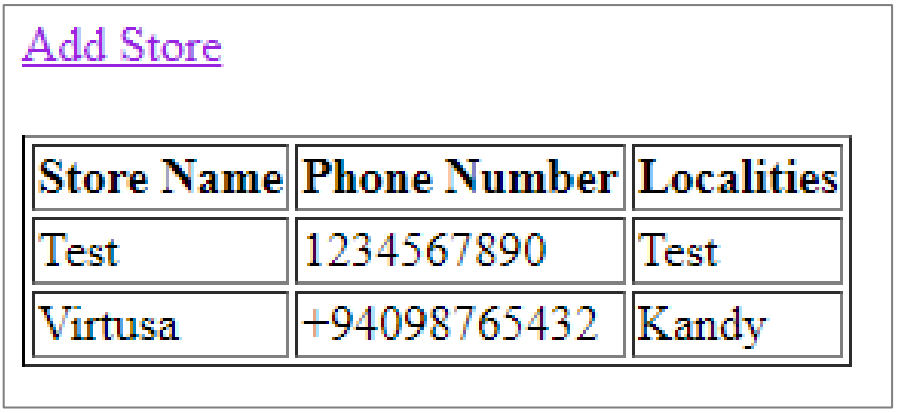
```java
@Controller
public class StoreController {

  @Autowired
  StoreService storeService;

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

  @GetMapping("/store")
  public ModelAndView viewStore() {
    ModelAndView mv = new ModelAndView("store");

    List<Store> stores = storeService.listStore();

    mv.addObject("stores", stores);

    return mv;
  }

  // Add Store
  @GetMapping("/addStore")
  public String addStore(Model model) {
    Store store = new Store();

    model.addAttribute("store", store);

    return "add-store";
  }

  @PostMapping("/saveAddStore")
  public String saveAddStore(@ModelAttribute("store") Store store) {

    storeService.addStore(store);

    return "redirect:/store";
  }

  // Edit Store
  @GetMapping("/editStore")
  public String editStore(@RequestParam("id") int id, Model model) {

    Store store = storeService.findById(id);

    model.addAttribute("store", store);

    return "edit-store";
  }

  @PostMapping("/saveEditStore")
  public String saveEditStore(@ModelAttribute("store") Store store) {

    storeService.editStore(store);

    return "redirect:/store";
  }

  // Delete Store
  @GetMapping("/deleteStore")
  public String saveEditStore(@RequestParam("id") int id) {

    storeService.deleteStore(id);

    return "redirect:/store";
  }

}
```

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

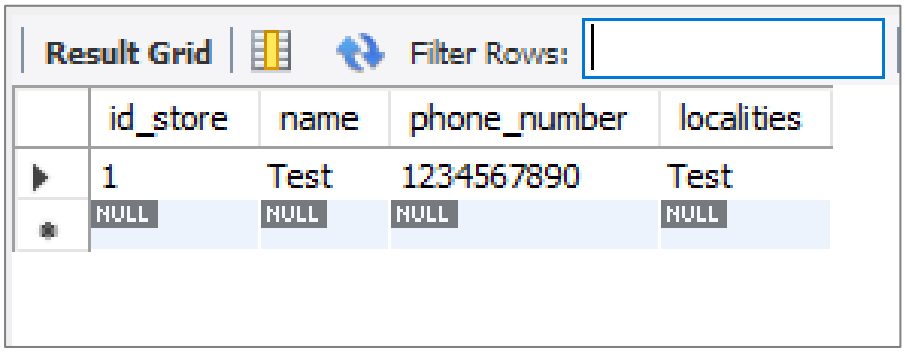


e. **Ensure that view stores request works end-to-end.**

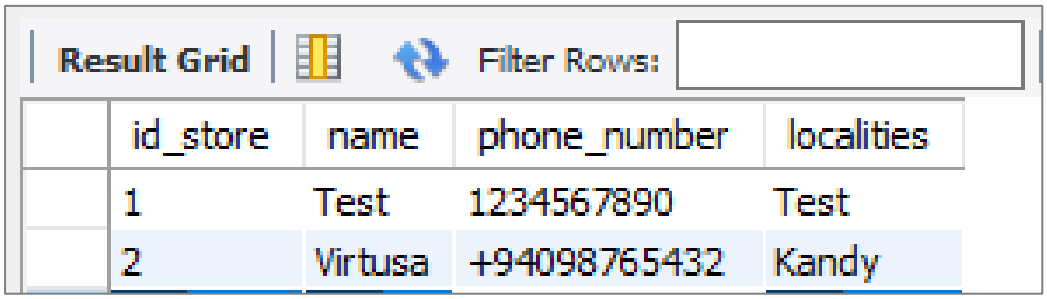




Database – Before



Database - After



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

✓ **Add Store**

```
<%@ 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

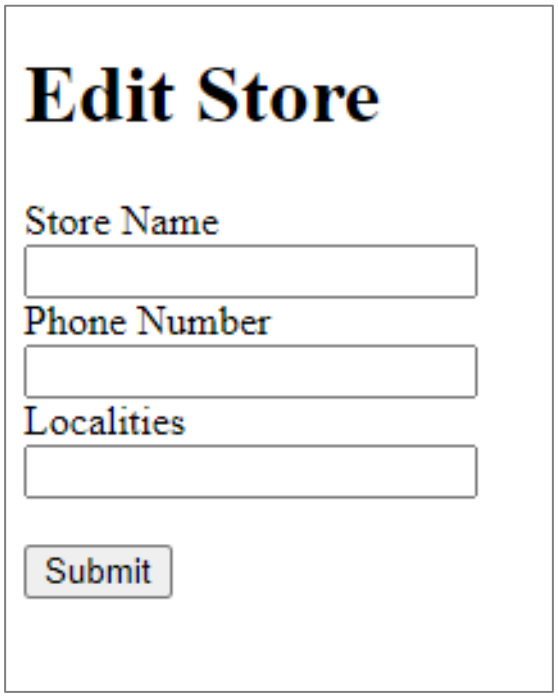
```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="ISO-8859-1" />
    <title>Know Your Neighborhood</title>
  </head>
  <body>
    <a href="<%= request.getContextPath() %>/addStore">Add Store</a>

    <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

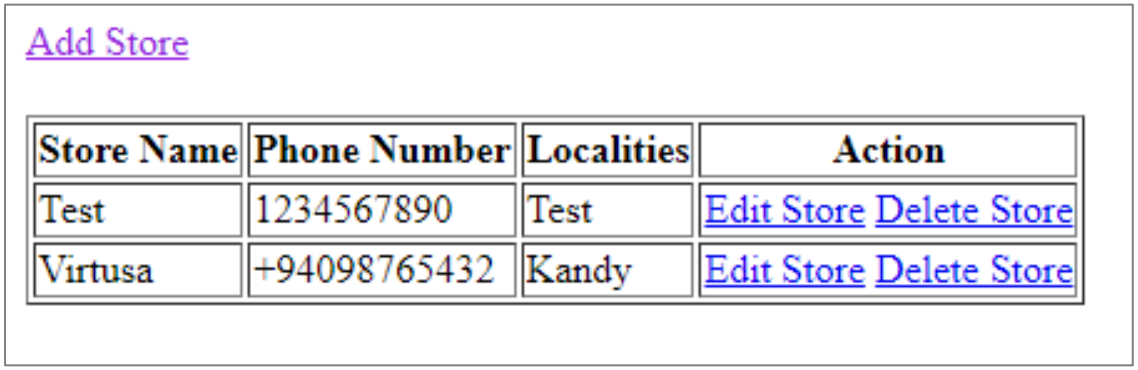
Store Name	Phone Number	Localities
Test	1234567890	Test
Virtusa	+94098765432	Kandy

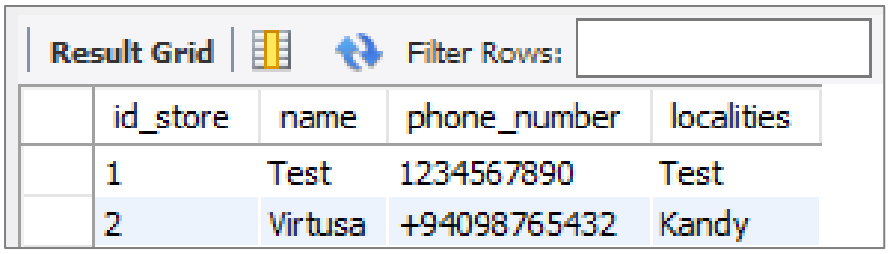
- ✓ **Update Store**

```
<!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 – Before Edit





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