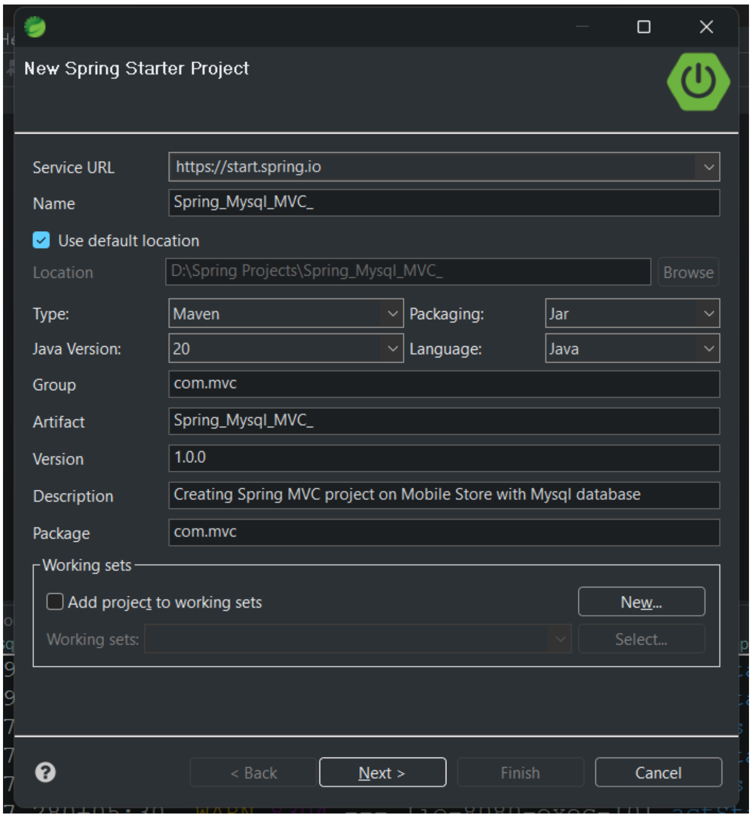
Spring MVC Project – Mobile Store

1. Create spring project and add following dependencies



New Spring Starter Project

Service URL: https://start.spring.io

Name: Spring_Mysql_MVC_

☑ Use default location

Location: D:\Spring Projects\Spring_Mysql_MVC_

Type: Maven

Packaging: Jar

Java Version: 20

Language: Java

Group: com.mvc

Artifact: Spring_Mysql_MVC_

Version: 1.0.0

Description: Creating Spring MVC project on Mobile Store with Mysql database

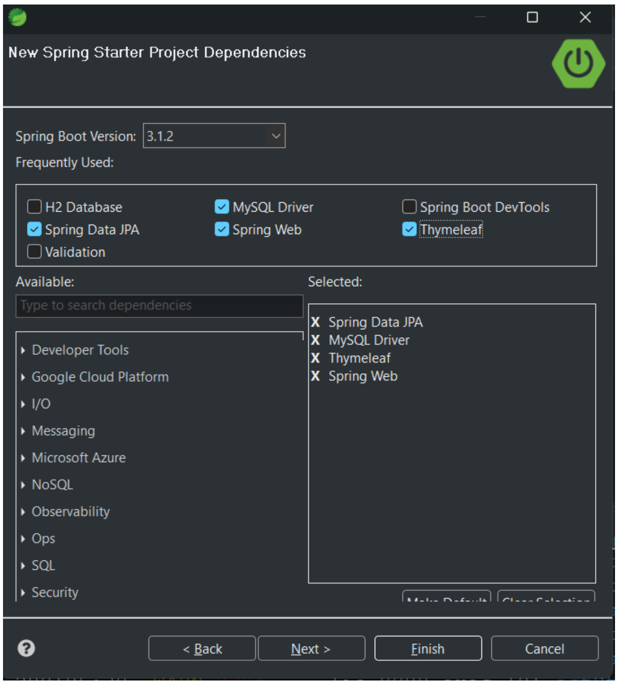
Package: com.mvc

Working sets

☐ Add project to working sets

Working sets:

2. Dependencies :



New Spring Starter Project Dependencies

Spring Boot Version: 3.1.2

Frequently Used:

☐ H2 Database
☑ MySQL Driver
☐ Spring Boot DevTools
☑ Spring Data JPA
☑ Spring Web
☑ Thymeleaf
☐ Validation

Available:

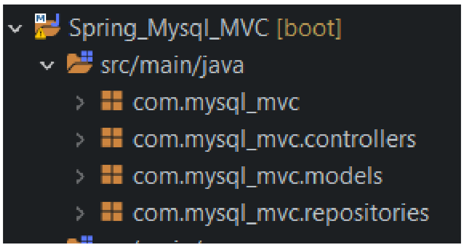
Type to search dependencies

- Developer Tools
- Google Cloud Platform
- I/O
- Messaging
- Microsoft Azure
- NoSQL
- Observability
- Ops
- SQL
- Security

Selected:

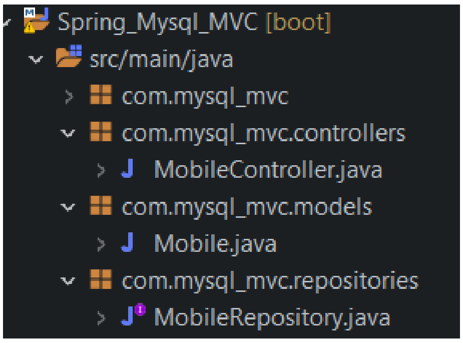
X Spring Data JPA
X MySQL Driver
X Thymeleaf
X Spring Web

3. Create 3 packages : models, repositories, controllers



```
Spring_Mysql_MVC [boot]
  src/main/java
    com.mysql_mvc
    com.mysql_mvc.controllers
    com.mysql_mvc.models
    com.mysql_mvc.repositories
```

4. Create 2 classes and 1 interface in respective packages



```
Spring_Mysql_MVC [boot]
  src/main/java
    com.mysql_mvc
    com.mysql_mvc.controllers
      MobileController.java
    com.mysql_mvc.models
      Mobile.java
    com.mysql_mvc.repositories
      MobileRepository.java
```

5. Code for Mobile.java :

```
package com.mysql_mvc.models;

import jakarta.persistence.Column;
import jakarta.persistence.Entity;
import jakarta.persistence.GeneratedValue;
import jakarta.persistence.GenerationType;
import jakarta.persistence.Id;

@Entity
public class Mobile {

        @Id // for primary key
        @GeneratedValue(strategy = GenerationType.AUTO) // to auto increment primary key
        int id;

        // @Column(name="mobile_name") used to change column name if you want
        @Column(nullable = false)
        String name;

        @Column(nullable = false, unique = true)
        String price;

        public Mobile() {
        }

        public Mobile(int id, String name, String price) {
                super();
                this.id = id;
                this.name = name;
                this.price = price;
```

```
    }

    public int getId() {
        return id;
    }

    public void setId(int id) {
        this.id = id;
    }

    public String getName() {
        return name;
    }

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

    public String getPrice() {
        return price;
    }

    public void setPrice(String price) {
        this.price = price;
    }
}
```

6. Code for MobileRepository.java :

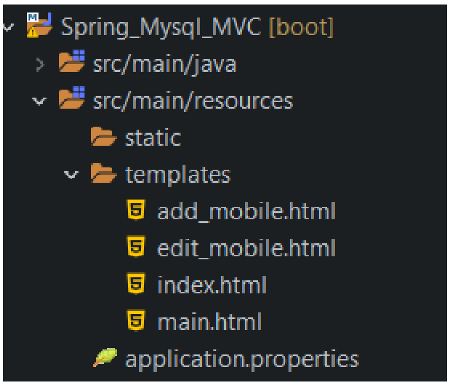
```
package com.mysql_mvc.repositories;

import org.springframework.data.repository.CrudRepository;
import com.mysql_mvc.models.Mobile;

public interface MobileRepository extends CrudRepository<Mobile, Integer>{
}
```

7. Code for MobileRepository.java :

```
package com.mysql_mvc.controllers;
import java.util.Iterator;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.ModelAttribute;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.RequestParam;
import com.mysql_mvc.models.Mobile;
import com.mysql_mvc.repositories.MobileRepository;

@Controller
public class MobileController {

    @Autowired
    MobileRepository mobileRepository;

    @GetMapping("/")
    public String index(Model model) {
        Iterable<Mobile> data = mobileRepository.findAll();
        Iterator<Mobile> all_mobiles = data.iterator();
```

```
            model.addAttribute("all_mobiles", all_mobiles);
            return "index";
    }

    @GetMapping("/add-mobile")
    public String add_mobile(Model model) {
            model.addAttribute("mobile_object", new Mobile());
            return "add_mobile";
    }

    @PostMapping("/add-mobile")
    public String add_mobile(@ModelAttribute Mobile mobile, Model model) {
            System.out.println(mobile.getId());
            System.out.println(mobile.getName());
            System.out.println(mobile.getPrice());

            mobileRepository.save(mobile);
            Iterable<Mobile> data = mobileRepository.findAll();
            Iterator<Mobile> all_mobiles = data.iterator();
            model.addAttribute("all_mobiles", all_mobiles);
            return "redirect:/";
    }

    @GetMapping("/edit-mobile")
    public String edit_mobile(@RequestParam("id") int mobileId, Model model) {
            Mobile mobile = mobileRepository.findById(mobileId).orElse(null);
            model.addAttribute("mobile_object", mobile);
            return "edit_mobile";
    }

    @PostMapping("/edit-mobile")
    public String update_mobile(@ModelAttribute("mobile_object") Mobile updatedMobile) {
            Mobile existingMobile = mobileRepository.findById(updatedMobile.getId()).orElse(null);
            if (existingMobile != null) {
                    existingMobile.setName(updatedMobile.getName());
                    existingMobile.setPrice(updatedMobile.getPrice());
                    mobileRepository.save(existingMobile);
            }
            return "redirect:/";
    }

    @GetMapping("/delete-mobile")
    public String deleteMobile(@RequestParam("id") int mobileId) {
            mobileRepository.deleteById(mobileId);
            return "redirect:/";
    }
}
```

8. Create 4 html files in templates folder.



```
Spring_Mysql_MVC [boot]
  src/main/java
  src/main/resources
    static
    templates
      add_mobile.html
      edit_mobile.html
      index.html
      main.html
    application.properties
```

9. Code for main.html :

```
<!doctype html>
<html xmlns:th="https://www.thymeleaf.org/">
  <head th:fragment="header_fragment">
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha3/dist/css/bootstrap.min.css" rel="stylesheet"
integrity="sha384-KK94CHFLLe+nY2dmCWGMq91rCGa5gtU4mk92HdvYe+M/SXH301p5ILy+dN9+nJOZ"
crossorigin="anonymous">
  </head>
  <body>
  <!-- code for navbar start  -->
  <div th:fragment="navbar_fragment">
    <nav class="navbar navbar-expand-lg bg-body-tertiary">
     <div class="container-fluid">
       <a class="navbar-brand" href="#">SPRING MVC PROJECT</a>
       <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarSupportedContent"
aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation">
         <span class="navbar-toggler-icon"></span>
       </button>
       <div class="collapse navbar-collapse" id="navbarSupportedContent">
         <ul class="navbar-nav me-auto mb-2 mb-lg-0">
           <li class="nav-item">
             <a class="nav-link active" aria-current="page" href="#">Home</a>
           </li>
         </ul>
       </div>
     </div>
   </nav>
  </div>

   <script th:fragment="js_fragment" src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-
alpha3/dist/js/bootstrap.bundle.min.js" integrity="sha384-
ENjdO4Dr2bkBIFxQpeoTz1HIcje39Wm4jDKdf19U8gI4ddQ3GYNS7NTKfAdVQSZe" crossorigin="anonymous"></script>

  </body>
</html>
```

10. Code for index.html :

```
<!DOCTYPE html>
<html  xmlns:th="https://www.thymeleaf.org/">
<head th:include="main.html :: header_fragment">
<meta charset="ISO-8859-1">
<title>MVC CRUD</title>
</head>
<body>
 <div th:include="main.html :: navbar_fragment"></div>

 <div class="container text-center">
   <a class="h1 btn btn-primary mt-3" th:href="@{/add-mobile}">Add New Mobile</a>
   <table class="table mt-3 table-bordered">
     <thead>
       <tr class="h4">
         <th>ID</th>
         <th>NAME</th>
         <th>Price</th>
         <th colspan="2">OPERATION</th>
       </tr>
     </thead>
     <tbody>
       <tr class="h5" th:each="mobile: ${all_mobiles}">
        <td th:text="${mobile.id}" />
        <td th:text="${mobile.name}" />
        <td th:text="${mobile.price}" />
        <td> <a class="btn btn-warning mt-3" th:href="@{/edit-mobile(id=${mobile.id})}">Edit</a> </td>
        <td> <a class="btn btn-danger mt-3" th:href="@{/delete-mobile(id=${mobile.id})}" onclick="return confirm('Are you
sure you want to delete this mobile?')">Delete</a> </td>
    </tr>
     </tbody>
  </table>
 </div>

 <script th:replace="main.html :: js_fragment"></script>

</body>
</html>
```

11. Code for add_mobile.html :

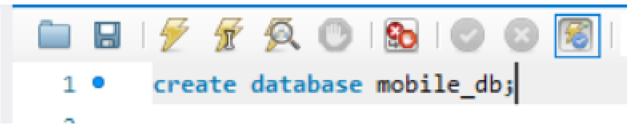
```
<!DOCTYPE html>
<html  xmlns:th="https://www.thymeleaf.org/">
<head th:include="main.html :: header_fragment">
<meta charset="ISO-8859-1">
<title>MVC CRUD</title>
</head>
<body>
 <div th:include="main.html :: navbar_fragment"></div>

 <div class="container text-center">
   <h1>Add Mobile</h1>
 </div>

 <!-- Form code -->
 <div class="container">
   <form class="col-6" th:object="${mobile_object}" method="post" th:action="@{/add-mobile}">
     <div class="mb-3">
       <label for="mobile_name" class="form-label">Name :</label>
       <input type="text" class="form-control" id="mobile_name" th:field="*{name}">
     </div>
     <div class="mb-3">
       <label for="mobile_price" class="form-label">Price :</label>
       <input type="text" class="form-control" id="mobile_price" th:field="*{price}">
     </div>
     <button type="submit" class="btn btn-primary">Submit</button>
   </form>
 </div>

 <script th:replace="main.html :: js_fragment"></script>

</body>
</html>
```

12. Code for edit_mobile.html :

```
<!DOCTYPE html>
<html xmlns:th="https://www.thymeleaf.org/">
<head th:include="main.html :: header_fragment">
    <meta charset="ISO-8859-1">
    <title>MVC CRUD - Edit Mobile</title>
</head>
<body>
<div th:include="main.html :: navbar_fragment"></div>
```

```
<div class="container text-center">
   <h1>Edit Mobile</h1>
</div>

<!-- Form code -->
<div class="container">
   <form class="col-6" th:object="${mobile_object}" method="post" th:action="@{/edit-mobile}">
      <input type="hidden" th:field="*{id}" />
      <div class="mb-3">
         <label for="mobile_name" class="form-label">Name:</label>
         <input type="text" class="form-control" id="mobile_name" th:field="*{name}"
th:placeholder="${mobile_object.name}" />
      </div>
      <div class="mb-3">
         <label for="mobile_price" class="form-label">Price:</label>
         <input type="text" class="form-control" id="mobile_price" th:field="*{price}"
th:placeholder="${mobile_object.price}" />
      </div>
      <button type="submit" class="btn btn-primary">Update</button>
   </form>
</div>

<script th:replace="main.html :: js_fragment"></script>

</body>
</html>
```

13. Creating database mobile_db in Mysql Workbench :

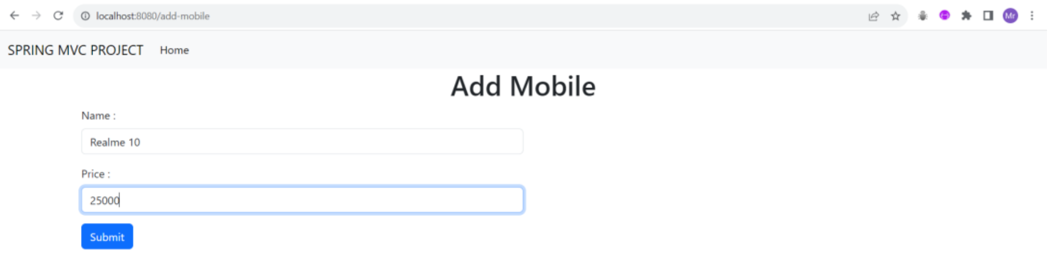


```
create database mobile_db;
```

14. Setting credentials in application.properties to use MYSQL database

```
spring.datasource.url=jdbc:mysql://localhost:3306/mobile_db
spring.datasource.username=root
spring.datasource.password=root
spring.jpa.hibernate.ddl-auto=update
```

15. Restart project and add some data

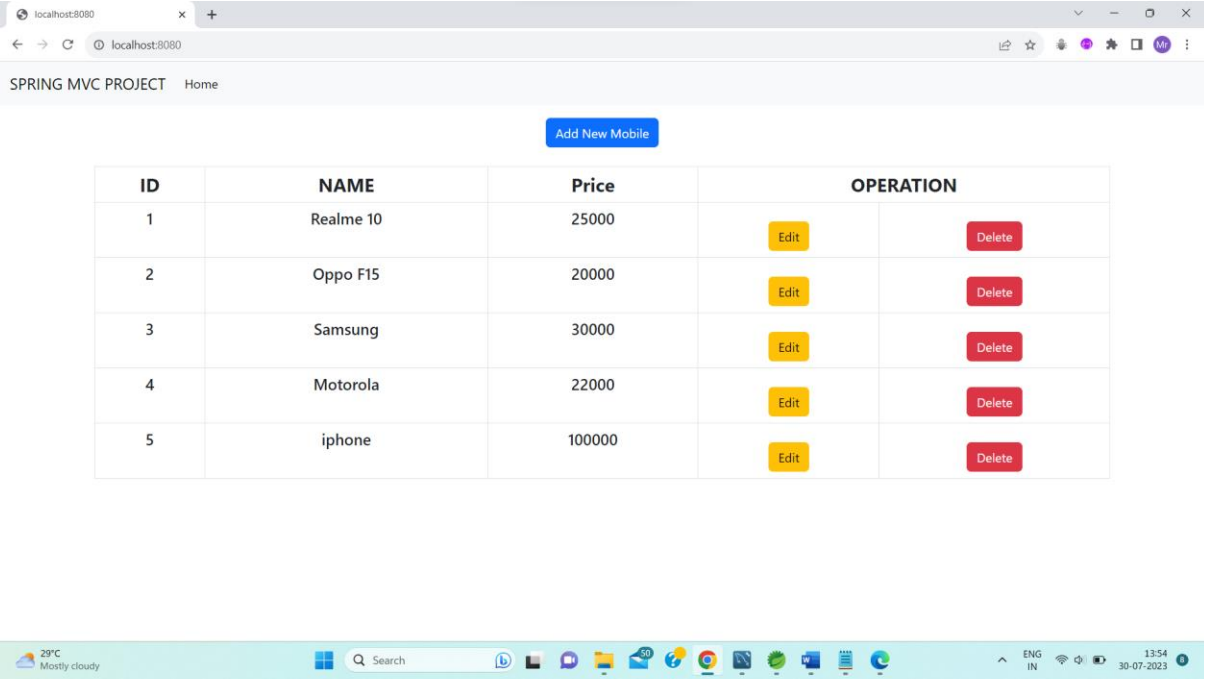


Add Mobile

Name : Realme 10

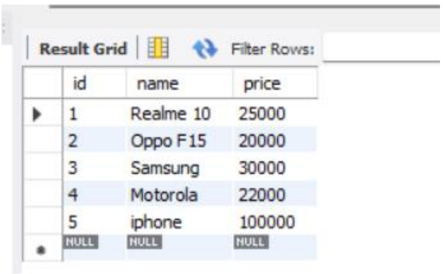
Price : 25000

OUTPUT :



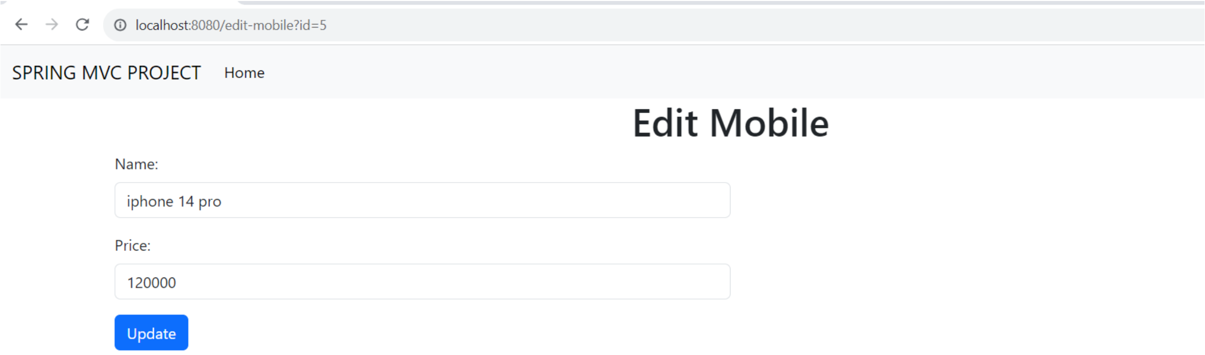
ID	NAME	Price	OPERATION	
1	Realme 10	25000	Edit	Delete
2	Oppo F15	20000	Edit	Delete
3	Samsung	30000	Edit	Delete
4	Motorola	22000	Edit	Delete
5	iphone	100000	Edit	Delete

Mobile_Store table in the database :



id	name	price
1	Realme 10	25000
2	Oppo F15	20000
3	Samsung	30000
4	Motorola	22000
5	iphone	100000
NULL	NULL	NULL

Editing data :



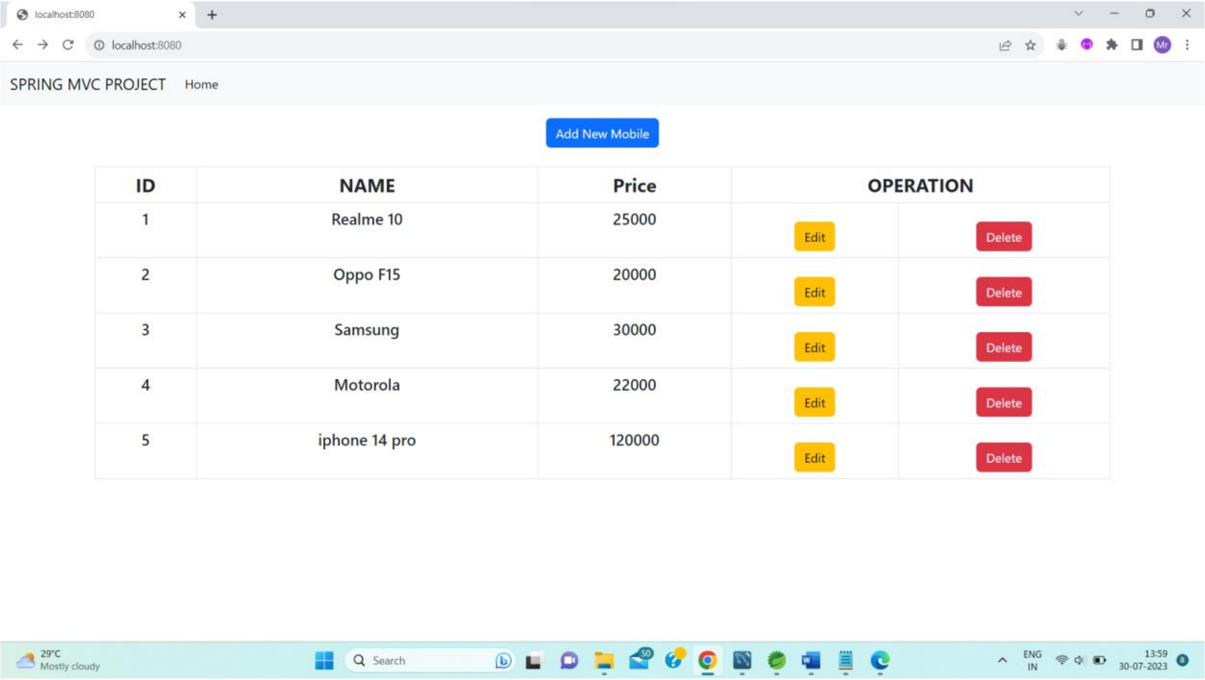
Edit Mobile

Name:

iphone 14 pro

Price:

120000

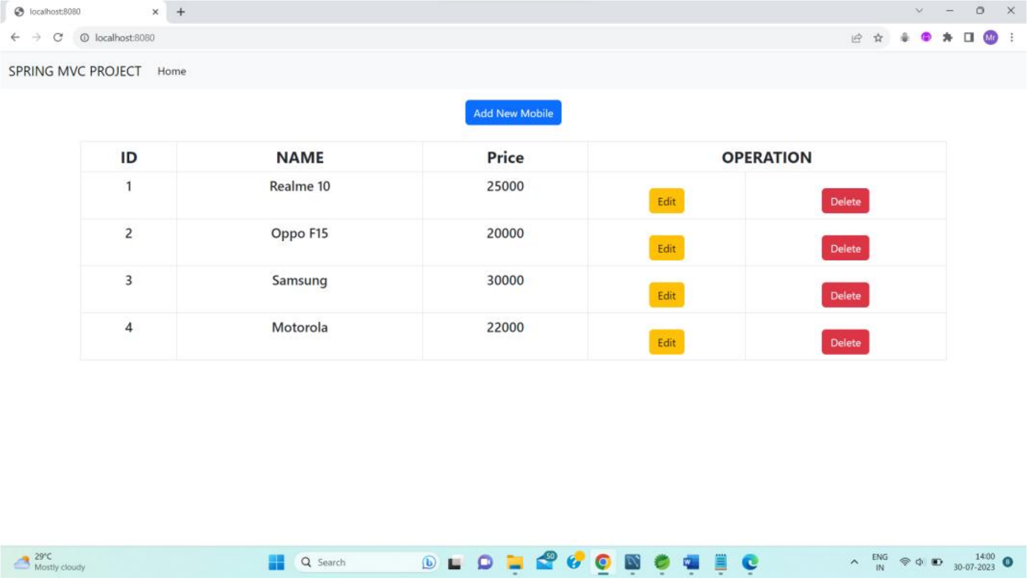


SPRING MVC PROJECT Home

Add New Mobile

ID	NAME	Price	OPERATION	
1	Realme 10	25000	Edit	Delete
2	Oppo F15	20000	Edit	Delete
3	Samsung	30000	Edit	Delete
4	Motorola	22000	Edit	Delete
5	iphone 14 pro	120000	Edit	Delete

Deleting record :



SPRING MVC PROJECT Home

Add New Mobile

ID	NAME	Price	OPERATION	
1	Realme 10	25000	Edit	Delete
2	Oppo F15	20000	Edit	Delete
3	Samsung	30000	Edit	Delete
4	Motorola	22000	Edit	Delete