

# VACCINATION CENTER

## SOURCE CODE

### com.ecommerce.SpringBootMvcDemoApplication

```
package com.ecommerce;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.context.annotation.ComponentScan;

@ComponentScan({"com.ecommerce.controllers", "com.ecommerce.entity", "com.ecommerce.repositories"})
@SpringBootApplication

public class SpringBootMvcDemoApplication {

    public static void main(String[] args) {
        SpringApplication.run(SpringBootMvcDemoApplication.class,
args);
    }

}
```

### com.ecommerce.controllers.CenterController.java

```
package com.ecommerce.controllers;

import java.util.List;
import java.util.Optional;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.http.ResponseEntity;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.*;

import com.ecommerce.entity.Center;
import com.ecommerce.entity.Citizen;
import com.ecommerce.repositories.CenterRepository;
import com.ecommerce.repositories.CitizenRepository;

@Controller
public class CenterController {

    @Autowired
    CenterRepository centerRepository;

    @Autowired
```

```

CitizenRepository citizenRepository;

@GetMapping("/add-center")
public String ShowAddCenterForm(Model model) {
    model.addAttribute("center", new Center());
    return "add-center";
}

@PostMapping("/add-center")
public String SaveCenter(@ModelAttribute("center") Center center, Model
model) {
    // Save the center to the database
    centerRepository.save(center);
    String alertMessage = "Center added to List";
    model.addAttribute("alertMessage", alertMessage);
    // Redirect to a success page or dashboard
    return "dashboard";
}

@GetMapping("/centers")
public ResponseEntity<List<Center>> getCenters() {
    List<Center> centerList = centerRepository.findAll(); // Assuming
you have a CenterRepository for fetching the data from the database

    // Check if centers exist
    if (centerList.isEmpty()) {
        return ResponseEntity.noContent().build();
    }

    return ResponseEntity.ok(centerList);
}

// Get ID from dashboard to edit
@GetMapping("/edit-center")
public String showEditProductForm(@RequestParam("centerID") int id,
Model model) {
    Optional<Center> productFromRepo = centerRepository.findById(id);

    Center center = productFromRepo.get();

    model.addAttribute("center", center);
    return "edit-center"; // go to edit-product.jsp
}

@PostMapping("/edit-center-save")
public String saveEditedProduct(@ModelAttribute("product") Center
center, Model model) {

    centerRepository.save(center);
    String alertMessage = "Center Details Updated";
    model.addAttribute("alertMessage", alertMessage);

    return "dashboard"; // go to edit-product-saved-success.jsp
}

// Delete a Center
@GetMapping("/delete-center")
public String deleteProduct(@RequestParam int centerID, Model model) {
    Optional<Center> productFromRepo =
centerRepository.findById(centerID);

```

```

        centerRepository.deleteById(centerID);
        model.addAttribute("id", centerID);
        String alertMessage = "Center Deleted From the list";
        model.addAttribute("alertMessage", alertMessage);
        // redirectAttributes.addFlashAttribute("alertMessage", "Center
deleted successfully.");

        return "dashboard"; // go to delete-product-success.jsp
    }

    // View a particular record
    @GetMapping("/view-center")
    public String showEditProductForm1(@RequestParam("centerID") int id,
Model model) {
        Optional<Center> centerFromRepo = centerRepository.findById(id);

        if (centerFromRepo.isPresent()) {
            Center center = centerFromRepo.get();
            List<Citizen> citizens =
citizenRepository.findByCenterCenterName(center.getCenterName());

            model.addAttribute("center", center);
            model.addAttribute("citizens", citizens);
        }

        return "view-center"; // go to view-center.jsp
    }

    // View center by name
    @GetMapping("/view-center-details")
    public String viewCenterDetails(@RequestParam("centerName") String
centerName, Model model) {
        List<Center> centers =
centerRepository.findAllByCenterName(centerName);
        model.addAttribute("centers", centers);
        return "view-center-details"; // go to view-center-details.jsp
    }
}

```

## com.ecommerce.controllers.CitizenController.java

```

package com.ecommerce.controllers;

import java.util.List;
import java.util.Optional;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.http.ResponseEntity;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.validation.BindingResult;
import org.springframework.web.bind.annotation.*;
import com.ecommerce.entity.Center;

```

```

import com.ecommerce.entity.Citizen;
import com.ecommerce.repositories.CenterRepository;
import com.ecommerce.repositories.CitizenRepository;

@Controller
public class CitizenController {

    @Autowired
    CitizenRepository citizenRepository;

    @Autowired
    CenterRepository centerRepository;

    @GetMapping("/citizens")
    public ResponseEntity<List<Citizen>> listCitizens(Model model) {
        List<Citizen> citizenList = citizenRepository.findAll();

        // Fetch center details for each citizen
        for (Citizen citizen : citizenList) {
            citizen.getCenter().getCenterName(); // Access the center's
name or other details
        }

        return ResponseEntity.ok().body(citizenList);
    }

    @GetMapping("/add-citizen")
    public String register(Model model) {
        List<Center> centerNames = centerRepository.findAll();
        model.addAttribute("citizen", new Citizen());
        model.addAttribute("centers", centerNames);
        return "add-citizen";
    }

    @PostMapping("/add-citizen")
    public String addCitizen(@ModelAttribute("citizen") Citizen citizen,
BindingResult result, Model model) {
        if (result.hasErrors()) {
            model.addAttribute("centers", centerRepository.findAll());
            return "add-citizen";
        }

        // Get the center name from the citizen's associated center
        String centerName = citizen.getCenter().getCenterName();

        citizen.setCitizenDose("0");
        citizen.setCitizenVaccinationStatus("Not Vaccinated");
        // Set the center name in the citizen entity
        citizen.setCenterName(centerName);

        // Save the citizen
        citizenRepository.save(citizen);
        String alertMessage = "Citizen Data Added To the List";
        model.addAttribute("alertMessage", alertMessage);

        return "dashboard";
    }

    // View citizen
    @GetMapping("/view-citizen")

```

```

    public String showEditProductForm(@RequestParam("citizenID") int id,
Model model) {
        Optional<Citizen> productFromRepo = citizenRepository.findById(id);

        Citizen citizen = productFromRepo.get();

        model.addAttribute("citizen", citizen);
        return "view-citizen"; // go to edit-product.jsp
    }

    // Delete a Citizen
    @GetMapping("/delete-citizen")
    public String deleteProduct(@RequestParam int citizenID, Model model) {
        Optional<Citizen> productFromRepo =
citizenRepository.findById(citizenID);

        citizenRepository.deleteById(citizenID);
        //model.addAttribute("id", citizenID);
        String alertMessage = "Citizen Data Deleted from List";
        model.addAttribute("alertMessage", alertMessage);
        return "dashboard"; // go to delete-product-success.jsp
    }

    // Get ID from dashboard to edit
    @GetMapping("/edit-citizen")
    public String showEditProductForm1(@RequestParam("citizenID") int id,
Model model) {
        Optional<Citizen> productFromRepo = citizenRepository.findById(id);

        Citizen citizen = productFromRepo.get();
        List<Center> centerNames = centerRepository.findAll();
        model.addAttribute("centers", centerNames);
        model.addAttribute("citizen", citizen);
        return "edit-citizen"; // go to edit-product.jsp
    }

    @PostMapping("/edit-citizen-save")
    public String saveEditedProduct(@ModelAttribute("citizen1") Citizen
citizen, Model model) {
        if ((citizen.getCitizenDose()).equals("1")) {
            citizen.setCitizenVaccinationStatus("Partially Vaccinated");
        } else if ((citizen.getCitizenDose()).equals("2")) {
            citizen.setCitizenVaccinationStatus("Fully Vaccinated");
        } else {
            citizen.setCitizenVaccinationStatus("Not Vaccinated");
        }
        citizenRepository.save(citizen);
        String alertMessage = "Citizen Data is updated";
        model.addAttribute("alertMessage", alertMessage);
        return "dashboard"; // go to edit-product-saved-success.jsp
    }
}

```

## com.ecommerce.controllers.MainController.java

```
package com.ecommerce.controllers;

import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.GetMapping;

@Controller
public class MainController {

    @GetMapping("/")
    public String index(Model model) {
        return "login";
    }

}
```

## com.ecommerce.controllers.UserController.java

```
package com.ecommerce.controllers;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.http.ResponseEntity;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.*;
import com.ecommerce.entity.User;
import com.ecommerce.repositories.CenterRepository;
import com.ecommerce.repositories.UserRepository;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpSession;

@Controller
public class UserController {

    @Autowired
    UserRepository userRepository;

    @Autowired
    CenterRepository centerRepository;

    @GetMapping("/register")
    public String register(Model model) {
        model.addAttribute("user", new User());
        return "register";
    }

    @PostMapping("/register")
    public String processRegistrationForm(@ModelAttribute("user") User
user, Model model) {
        // Save the user to the database
        userRepository.save(user);
    }
}
```

```

        String alertMessage = "Registered Successfully";
        model.addAttribute("alertMessage", alertMessage);
        // Redirect to a success page or login page
        return "login";
    }

    @PostMapping("/login")
    public ResponseEntity<String> loginUser(@RequestBody User user,
        HttpServletRequest request) {
        User existingUser =
        userRepository.findByUserEmail(user.getUserEmail());
        HttpSession session = request.getSession();
        session.setAttribute("userEmail", user.getUserEmail());
        if (existingUser == null) {
            return ResponseEntity.badRequest().body("Invalid User");
        }

        // Validate the user's password
        if (!existingUser.getUserPassword().equals(user.getUserPassword()))
        {
            return ResponseEntity.badRequest().body("Invalid password");
        }

        return ResponseEntity.ok("User logged in successfully");
    }

    @GetMapping("/logout")
    public String logoutUser(HttpServletRequest request) {
        HttpSession session = request.getSession(false); // Get the current
        session, if it exists
        if (session != null) {
            session.invalidate(); // Invalidate the session, effectively
        logging out the user
        }

        return "login";
    }

    @GetMapping("/dashboard")
    public String dashboard(Model model, HttpServletRequest request) {
        // Retrieve the user email from the session
        HttpSession session = request.getSession();
        String userEmail = (String) session.getAttribute("userEmail");

        // Use the userEmail to fetch the user details from the database or
        any other source
        User user = userRepository.findByUserEmail(userEmail);

        // Add the user object to the model
        model.addAttribute("user", user);

        return "dashboard";
    }
}

```

## com.ecommerce.entity.Center.java

```
package com.ecommerce.entity;

import java.util.List;

import jakarta.persistence.CascadeType;
import jakarta.persistence.Column;
import jakarta.persistence.Entity;
import jakarta.persistence.GeneratedValue;
import jakarta.persistence.GenerationType;
import jakarta.persistence.Id;
import jakarta.persistence.OneToMany;
import jakarta.persistence.Table;

@Entity
@Table(name = "center")
public class Center {

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

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

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

    @OneToMany(mappedBy = "center", cascade = CascadeType.REMOVE)
    private List<Citizen> citizens;

    public int getCenterID() {
        return centerID;
    }

    public void setCenterID(int centerID) {
        this.centerID = centerID;
    }

    public String getCenterName() {
        return centerName;
    }

    public void setCenterName(String centerName) {
        this.centerName = centerName;
    }

    public String getCenterCity() {
        return centerCity;
    }

    public void setCenterCity(String centerCity) {
        this.centerCity = centerCity;
    }

    public List<Citizen> getCitizens() {
```



```

        return citizens;
    }

    public void setCitizens(List<Citizen> citizens) {
        this.citizens = citizens;
    }
}

```

## com.ecommerce.entity.Citizen.java

```

package com.ecommerce.entity;

import jakarta.persistence.Entity;
import jakarta.persistence.GeneratedValue;
import jakarta.persistence.GenerationType;
import jakarta.persistence.Id;
import jakarta.persistence.JoinColumn;
import jakarta.persistence.ManyToOne;
import jakarta.persistence.Table;
import jakarta.persistence.Transient;

@Entity
@Table(name = "citizen")
public class Citizen {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private int citizenID;

    private String citizenName;

    private String citizenCity;

    private String citizenDose;

    private String citizenVaccinationStatus;

    @Transient
    private String centerName;

    @ManyToOne
    @JoinColumn(name = "center_id", referencedColumnName = "centerID")
    private Center center;

    public Citizen() {
    }

    public int getCitizenID() {
        return citizenID;
    }

    public void setCitizenID(int citizenID) {
        this.citizenID = citizenID;
    }

    public String getCitizenName() {

```

```

        return citizenName;
    }

    public void setCitizenName(String citizenName) {
        this.citizenName = citizenName;
    }

    public String getCitizenCity() {
        return citizenCity;
    }

    public void setCitizenCity(String citizenCity) {
        this.citizenCity = citizenCity;
    }

    public String getCitizenDose() {
        return citizenDose;
    }

    public void setCitizenDose(String citizenDose) {
        this.citizenDose = citizenDose;
    }

    public String getCitizenVaccinationStatus() {
        return citizenVaccinationStatus;
    }

    public void setCitizenVaccinationStatus(String
citizenVaccinationStatus) {
        this.citizenVaccinationStatus = citizenVaccinationStatus;
    }

    public Center getCenter() {
        return center;
    }

    public void setCenter(Center center) {
        this.center = center;
    }

    public String getCenterName() {
        if (center != null) {
            return center.getCenterName();
        }
        return null;
    }

    public void setCenterName(String centerName) {
        this.centerName = centerName;
    }
}

```

## com.ecommerce.entity.User.java

```
package com.ecommerce.entity;

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

@Entity
@Table(name = "user")
public class User {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    @Column(name = "userId")
    private int ID;

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

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

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

    public User() {
    }

    public int getID() {
        return ID;
    }

    public void setID(int ID) {
        this.ID = ID;
    }

    public String getUserName() {
        return userName;
    }

    public void setUserName(String userName) {
        this.userName = userName;
    }

    public String getUserEmail() {
        return userEmail;
    }

    public void setUserEmail(String userEmail) {
        this.userEmail = userEmail;
    }

    public String getUserPassword() {
        return userPassword;
    }
}
```

```

        public void setUserPassword(String userPassword) {
            this.userPassword = userPassword;
        }
    }
}

```

## **com.ecommerce.repositories.CenterRepository.java**

```

package com.ecommerce.repositories;
import com.ecommerce.entity.Center;
import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.data.jpa.repository.Query;
import org.springframework.stereotype.Repository;

import java.util.List;

@Repository
public interface CenterRepository extends JpaRepository<Center, Integer> {
    @Query("SELECT c.centerName FROM Center c")

    List<String> findAllCenterNames();

    List<Center> findAll();
    List<Center> findAllByCenterName(String centerName);
}

```

## **com.ecommerce.repositories.CitizenRepository.java**

```

package com.ecommerce.repositories;

import com.ecommerce.entity.Citizen;
import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.data.jpa.repository.Query;
import org.springframework.data.repository.query.Param;
import org.springframework.stereotype.Repository;

import java.util.List;

@Repository
public interface CitizenRepository extends JpaRepository<Citizen, Integer>
{
    List<Citizen> findByCenterCenterName(String centerName);

    @Query("SELECT c.center.centerName FROM Citizen c WHERE c.center.centerID = :centerId")
    String findCenterNameById(@Param("centerId") int centerId);
}

```

## com.ecommerce.repositories.UserRepository.java

```
package com.ecommerce.repositories;

import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.stereotype.Repository;

import com.ecommerce.entity.User;

@Repository
public interface UserRepository extends JpaRepository<User, Integer> {
    User findByUserEmail(String userEmail);
}
```

## application.properties

```
#server.port=10000

#JSP files configuration
spring.mvc.view.prefix=/WEB-INF/views/
spring.mvc.view.suffix=.jsp

#Database
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
spring.datasource.url=jdbc:mysql://localhost:3306/springboot2
spring.datasource.username=root
spring.datasource.password=root123
#spring.jpa.show-sql=true
spring.jpa.hibernate.ddl-auto=update
spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQL8Dialect

spring.profiles.active=development
```

## add-center.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>Center Registration</title>
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/materialize/1.0.0/css/material
ize.min.css">
    <style>
        .small-container {
            max-width: 400px;
```

```

        margin: 0 auto;
    }
</style>
</head>
<body>
    <div class="container small-container">
        <div class="card-panel">
            <h5>ADD VACCINATION CENTER</h5>
            <form action="add-center" method="post"
modelAttribute="center">
                <div class="input-field">
                    <input id="centerName" type="text" name="centerName"
class="validate" required>
                        <label for="centerName">Center Name</label>
                    </div>
                    <div class="input-field">
                        <select id="centerCity" name="centerCity" required>
                            <option value="" disabled selected>Select Center
City</option>
                                <option value="Mumbai">Mumbai</option>
                                <option value="Bangalore">Bangalore</option>
                                <option value="Chennai">Chennai</option>
                                <option value="Kolkata">Kolkata</option>
                                <option value="Hyderabad">Hyderabad</option>
                                <option value="Jaipur">Jaipur</option>
                                <option value="Pune">Pune</option>
                                <option value="Kochi">Kochi</option>
                                <option value="Mysore">Mysore</option>
                                <option value="Thane">Thane</option>
                                <option value="Ranchi">Ranchi</option>
                            </select>
                            <label for="centerCity">Center City</label>
                        </div>
                        <button class="btn waves-effect waves-light"
type="submit">ADD VACCINATION CENTER</button>
                    </form>
                </div>
            </div>

            <script
src="https://cdnjs.cloudflare.com/ajax/libs/materialize/1.0.0/js/materializ
e.min.js"></script>
            <script>
                // Initialize Materialize dropdown
                document.addEventListener('DOMContentLoaded', function() {
                    var elems = document.querySelectorAll('select');
                    var instances = M.FormSelect.init(elems);
                });
            </script>
        </body>
</html>

```

## add-citizen.jsp

```

<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>

```

```

<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>

<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>Citizen Registration</title>
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/materialize/1.0.0/css/material
ize.min.css">
    <style>
        .small-container {
            max-width: 400px;
            margin: 0 auto;
        }
    </style>
</head>
<body>
    <div class="container small-container">
        <div class="card-panel">
            <h5>ADD CITIZEN</h5>
            <form action="add-citizen" method="post"
modelAttribute="citizen">
                <!-- Citizen name field -->
                <div class="input-field">
                    <label for="citizenName">Citizen Name</label>
                    <input id="citizenName" type="text" name="citizenName"
class="validate" required>
                </div>

                <!-- Citizen city field -->
                <div class="input-field">
                    <select id="citizenCity" name="citizenCity" required>
                        <option value="" disabled selected>Select Citizen
City</option>
                        <option value="Mumbai">Mumbai</option>
                        <option value="Bangalore">Bangalore</option>
                        <option value="Chennai">Chennai</option>
                        <option value="Kolkata">Kolkata</option>
                        <option value="Hyderabad">Hyderabad</option>
                        <option value="Jaipur">Jaipur</option>
                        <option value="Pune">Pune</option>
                        <option value="Kochi">Kochi</option>
                        <option value="Mysore">Mysore</option>
                        <option value="Thane">Thane</option>
                        <option value="Ranchi">Ranchi</option>
                    </select>
                </div>

                <!-- Center name field -->
                <div class="input-field">
                    <label for="centerName">Center Name</label>
                    <select id="center" name="center" required>
                        <option value="">Select a center</option>
                        <!-- Iterate over the center list and populate the
dropdown options -->
                        <c:forEach items="${centers}" var="center">
                            <option
value="${center.centerID}">${center.centerName}</option>
                        </c:forEach>
                    </select>
                </div>
            </form>
        </div>
    </div>

```

```

        </div>

        <!-- Submit button -->
        <button class="btn waves-effect waves-light"
type="submit">ADD CITIZEN</button>
        </form>
    </div>
</div>

<script
src="https://cdnjs.cloudflare.com/ajax/libs/materialize/1.0.0/js/materializ
e.min.js"></script>
<script>
    // Initialize Materialize dropdown
    document.addEventListener('DOMContentLoaded', function() {
        var elems = document.querySelectorAll('select');
        var instances = M.FormSelect.init(elems);
    });
</script>
</body>
</html>

```

## dashboard.jsp

```

<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1" %>
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>

<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Admin Dashboard</title>
    <link href="https://fonts.googleapis.com/icon?family=Material+Icons"
rel="stylesheet">
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/materialize/1.0.0/css/material
ize.min.css">
    <style>
        /* Add your custom styles here */
    </style>
</head>
<body>
    <div class="row" style="margin-bottom: 0;">
        <div class="col s12">
            <div class="row" style="margin-bottom: 0;">
                <div class="col s12 center" style="margin-top: .1rem;">
                    <ul class="tabs">
                        <li class="tab col s3"><a href="#citizen-add"
class="indigo-text text-darken-4">LIST CITIZENS</a></li>
                        <li class="tab col s3"><a href="#center-add"
class="indigo-text text-darken-4">VACCINATION CENTERS</a></li>
                        <li class="tab col s3"><a href="#" id="logout-link"
class="indigo-text text-darken-4">LOGOUT</a></li>

```



```

        <li class="tab col s3" style="color: red; font-
weight: bold;">WELCOME ${user.userName}</li>
    </ul>
</div>
</div>
</div>
</div>

<div id="citizen-add" class="col s8 tab-content">
    <div>
        WELCOME TO DASHBOARD
    </div>
</div>

<div id="center-add" class="col s8 tab-content">
    <div>
    </div>
</div>

<script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
<script
src="https://cdnjs.cloudflare.com/ajax/libs/materialize/1.0.0/js/materializ
e.min.js"></script>
<script>
    $(document).ready(function () {
        $(".tabs").tabs();

        $("a[href='#citizen-add']").click(function () {
            $.ajax({
                url: "/citizens",
                method: "GET",
                success: function (citizenList) {
                    // Clear the existing content
                    $("#citizen-add").empty();

                    // Create the button element
                    var addButton = $("

```

```

        $("<th>").text("CITIZEN DOSE"),
        $("<th>").text("VACCINATION STATUS"),
        $("<th>").text("CENTER NAME"),
        $("<th>").text(""),
        $("<th>").text("ACTION")
        // Add more columns as needed
    )
    );

    // Create the table body
    var tableBody = $("<tbody>");
    // Loop through the data and create table rows
    citizenList.forEach(function (citizen) {
        var row = $("<tr>").append(
            $("<td>").text(""),
            $("<td>").text(citizen.citizenID),
            $("<td>").text(citizen.citizenName),
            $("<td>").text(citizen.citizenCity),
            $("<td>").text(citizen.citizenDose),

            $("<td>").text(citizen.citizenVaccinationStatus),
            $("<td>").text(citizen.centerName),
            $("<td>").append(
                $("<a>").attr("href", "view-
citizen?citizenID=" + citizen.citizenID).text("VIEW")
            ),
            $("<td>").append(
                $("<a>").attr("href", "edit-
citizen?citizenID=" + citizen.citizenID).text("EDIT")
            ),
            $("<td>").append(
                $("<a>").attr("href", "delete-
citizen?citizenID=" + citizen.citizenID).text("DELETE")
            )
            // Add more columns as needed
        );
        tableBody.append(row);
    });

    // Append the table header and body to the table
    table.append(tableHeader).append(tableBody);

    // Append the table to the #center-add div
    $("#citizen-add").append(table);

    // Display the count of rows
    var rowCount = citizenList.length;
    var countRow = $("<tr>")
        .append(
            $("<td>")
                .attr("colspan", "6")
                .css({ "text-align": "center", "font-
weight": "bold" })
                .text("Total " + rowCount + "
Vaccination Centers Found")
        );
    tableBody.append(countRow);
},
error: function (xhr, status, error) {
    console.log("Error:", error);
}
}

```

```

        });
    });
</script>

<script>
$(document).ready(function () {
    $(".tabs").tabs();

    $("a[href='#center-add']").click(function () {
        $.ajax({
            url: "/centers",
            method: "GET",
            success: function (centerList) {
                // Clear the existing content
                $("#center-add").empty();

                // Create the button element
                var addButton = $("

```

```

        ),
        $(" ").append(             $("").attr("href", "edit- center?centerID=" + center.centerID).text("EDIT")         ),         $(" ").append(             $("").attr("href", "delete- center?centerID=" + center.centerID).text("DELETE")         )         // Add more columns as needed     );     tableBody.append(row); });  // Append the table header and body to the table table.append(tableHeader).append(tableBody);  // Append the table to the #center-add div $("#center-add").append(table);  // Display the count of rows var rowCount = centerList.length; var countRow = $("|  | | --- | |")     .append(         $(" ")             .attr("colspan", "6")             .css({ "text-align": "center", "font- weight": "bold" })             .text("Total " + rowCount + " Vaccination Centers Found")     );     tableBody.append(countRow); }, error: function (xhr, status, error) {     console.log("Error:", error); } }); }); }); </script>  <script>     document.getElementById("logout-link").addEventListener("click", function (event) {     event.preventDefault(); // Prevent the default behavior of the link      // Display the confirmation dialog     if (confirm("Are you sure you want to logout?")) {         // If the user confirms, redirect to the logout endpoint         window.location.href = "/logout";     } }); </script>  <script>     var alertMessage = "${alertMessage}";     if (alertMessage) {         alert(alertMessage);     } </script> | | |
```

```
</body>
```

```
</html>
```

## **edit-center.jsp**

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>Center Registration</title>
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/materialize/1.0.0/css/material
ize.min.css">
    <style>
        .small-container {
            max-width: 400px;
            margin: 0 auto;
        }
    </style>
</head>
<body>
    <div class="container small-container">
        <div class="card-panel">
            <h5>EDIT VACCINATION CENTER</h5>
            <form action="edit-center-save" method="post"
modelAttribute="center">
                <div class="input-field">
                    <input id="centerID" type="text" name="centerID"
class="validate" value="{center.centerID}" readonly="">
                    <label for="centerID">Center ID</label>
                </div>
                <div class="input-field">
                    <input id="centerName" type="text" name="centerName"
class="validate" value="{center.centerName}" required>
                    <label for="centerName">Center Name</label>
                </div>
                <div class="input-field">
                    <select id="centerCity" name="centerCity" required>
                        <option value="" disabled selected>Select Center
City</option>
                        <option value="Mumbai">Mumbai</option>
                        <option value="Bangalore">Bangalore</option>
                        <option value="Chennai">Chennai</option>
                        <option value="Kolkata">Kolkata</option>
                        <option value="Hyderabad">Hyderabad</option>
                        <option value="Jaipur">Jaipur</option>
                        <option value="Pune">Pune</option>
                        <option value="Kochi">Kochi</option>
                        <option value="Mysore">Mysore</option>
                        <option value="Thane">Thane</option>
                        <option value="Ranchi">Ranchi</option>
                    </select>
```

```

        <label for="centerCity">Center City</label>
    </div>
    <button class="btn waves-effect waves-light"
type="submit">EDIT</button>
    </form>
</div>
</div>

<script
src="https://cdnjs.cloudflare.com/ajax/libs/materialize/1.0.0/js/materializ
e.min.js"></script>
<script>
    // Initialize Materialize dropdown
    document.addEventListener('DOMContentLoaded', function() {
        var elems = document.querySelectorAll('select');
        var instances = M.FormSelect.init(elems);
    });
</script>
<script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
<script>
    // Get the value of the centerCity from the model
    var centerCityValue = "${center.centerCity}";

    // Set the selected option based on the centerCity value
    $("#centerCity").val(centerCityValue);
</script>
</body>
</html>

```

## edit-citizen.jsp

```

<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>

<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>Center Registration</title>
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/materialize/1.0.0/css/material
ize.min.css">
    <style>
        .small-container {
            max-width: 400px;
            margin: 0 auto;
        }
    </style>
</head>
<body>
    <div class="container small-container">
        <div class="card-panel">
            <h5>EDIT CITIZEN</h5>
            <form action="edit-citizen-save" method="post"
modelAttribute="citizen1">

```

```

        <div class="input-field">
            <input id="citizenID" type="text" name="citizenID"
class="validate" value="{citizen.citizenID}" readonly="">
            <label for="citizenID">Citizen ID</label>
        </div>
        <div class="input-field">
            <input id="citizenName" type="text" name="citizenName"
class="validate" value="{citizen.citizenName}">
            <label for="citizenName">Citizen Name</label>
        </div>
        <div class="input-field">
            <select id="citizenCity" name="citizenCity" required>
                <option value="" disabled selected>Select Center
City</option>
                <option value="Mumbai">Mumbai</option>
                <option value="Bangalore">Bangalore</option>
                <option value="Chennai">Chennai</option>
                <option value="Kolkata">Kolkata</option>
                <option value="Hyderabad">Hyderabad</option>
                <option value="Jaipur">Jaipur</option>
                <option value="Pune">Pune</option>
                <option value="Kochi">Kochi</option>
                <option value="Mysore">Mysore</option>
                <option value="Thane">Thane</option>
                <option value="Ranchi">Ranchi</option>
            </select>
            <label for="citizenCity">Center City</label>
        </div>
        <div class="input-field">
            <select id="citizenDose" name="citizenDose" required>
                <option value="" disabled selected>Select
Dose</option>
                <option value="0">None</option>
                <option value="1">1</option>
                <option value="2">2</option>
            </select>
            <label for="citizenDose">Center Dose</label>
        </div>
        <label for="centerName">Center Name</label>
        <select id="center" name="center" required>
            <option value="" disabled selected>Select a
center</option>
            <!-- Iterate over the center list and populate the
dropdown options -->
            <c:forEach items="{centers}" var="center">
                <option
value="{center.centerID}" "{center.centerName}"></option>
            </c:forEach>
        </select>
        <button class="btn waves-effect waves-light"
type="submit">EDIT</button>
    </form>
</div>
</div>

<script
src="https://cdnjs.cloudflare.com/ajax/libs/materialize/1.0.0/js/materializ
e.min.js"></script>
<script>
    // Initialize Materialize dropdown
    document.addEventListener('DOMContentLoaded', function() {

```

```

        var elems = document.querySelectorAll('select');
        var instances = M.FormSelect.init(elems);
    });
</script>
<script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
<script>
    // Get the value of the citizenCity from the model
    var citizenCityValue = "${citizen.citizenCity}";

    // Set the selected option based on the citizenCity value
    $("#citizenCity").val(citizenCityValue);

    // Get the value of the citizenDose from the model
    var citizenDoseValue = "${citizen.citizenDose}";

    // Set the selected option based on the citizenDose value
    $("#citizenDose").val(citizenDoseValue);

    // Get the value of the center from the model
    var centerValue = "${citizen.center.centerID}";

    // Set the selected option based on the center value
    $("#center").val(centerValue);
</script>
</body>
</html>

```

## login.jsp

```

<!DOCTYPE html>
<html>
<head>
    <title>Login</title>
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/materialize/1.0.0/css/material
ize.min.css">
    <script>
        function showAlert(message) {
            alert(message);
        }
    </script>
    <script>
        var alertMessage = "${alertMessage}";
        if (alertMessage) {
            alert(alertMessage);
        }
    </script>
    <style>
        .small-container {
            max-width: 400px;
            margin: 0 auto;
        }
        h4 {
            text-align: center;

```



```

        text-shadow: 2px 2px 4px rgba(0, 0, 0, 0.5);
        color: #333;
        margin-top: 1em;
    }
</style>
</head>
<body>
    <h4 align="center">VACCINATION CENTER</h4>
    <div class="container small-container" style="margin-top: 3em;">
        <div class="card-panel">
            <h5>USER LOGIN</h5>
            <form id="loginForm">
                <div class="input-field">
                    <input id="userEmail" type="text" name="userEmail"
class="validate" required>
                    <label for="userEmail">User Email</label>
                </div>
                <div class="input-field">
                    <input id="userPassword" type="password"
name="userPassword" class="validate" required>
                    <label for="userPassword">User Password</label>
                </div>
                <button class="btn waves-effect waves-light"
type="submit">LOGIN</button>
                <button class="btn waves-effect waves-light red"
onclick="window.location.href = '/register';" type="submit" style="margin-
left: 2em;">REGISTER</button>
            </form>
        </div>
    </div>
    <div id="message"></div>
    <script>
        document.getElementById("loginForm").addEventListener("submit",
function(event) {
            event.preventDefault(); // Prevent form submission
            // Get form data
            var userEmail = document.getElementById("userEmail").value;
            var userPassword =
document.getElementById("userPassword").value;
            // Create an object with the form data
            var user = {
                userEmail: userEmail,
                userPassword: userPassword
            };
            // Send the request as JSON
            fetch('/login', {
                method: 'POST',
                headers: {
                    'Content-Type': 'application/json'
                },
                body: JSON.stringify(user)
            })
            .then(function(response) {
                if (response.ok) {
                    return response.text();
                } else if (response.status === 400) {
                    return response.text().then(function(errorMessage) {
                        throw new Error(errorMessage);
                    });
                } else {
                    throw new Error(response.status);
                }
            });
        });
    </script>

```

```

        }
    })
    .then(function(data) {
        if (data === "User logged in successfully") {
            // Redirect to the desired page after successful login
            window.location.href = "/dashboard";
        }
    })
    .catch(function(error) {
        showAlert(error.message); // Display the error message as
an alert
        console.log(error); // Handle any errors here
    });
});
</script>
<script>
src="https://cdnjs.cloudflare.com/ajax/libs/materialize/1.0.0/js/materializ
e.min.js"></script>
<script>
    // Initialize Materialize dropdown
    document.addEventListener('DOMContentLoaded', function() {
        var elems = document.querySelectorAll('select');
        var instances = M.FormSelect.init(elems);
    });
</script>
</body>
</html>

```

## register.jsp

```

<!DOCTYPE html>
<html>
<head>
    <title>Login</title>
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/materialize/1.0.0/css/material
ize.min.css">
    <script>
        function showAlert(message) {
            alert(message);
        }
    </script>
    <style>
        .small-container {
            max-width: 400px;
            margin: 0 auto;
        }
    </style>
</head>
<body>
    <div class="container small-container" style="margin-top: 5em;">
        <div class="card-panel">
            <h5>USER REGISTRATION</h5>
            <form action="register" method="post" modelAttribute="user">
                <div class="input-field">

```

```

        <input id="userName" type="text" name="userName"
class="validate" required>
        <label for="userName">User Name</label>
    </div>
    <div class="input-field">
        <input id="userEmail" type="text" name="userEmail"
class="validate" required>
        <label for="userEmail">User Email</label>
    </div>
    <div class="input-field">
        <input id="userPassword" type="password"
name="userPassword" class="validate" required>
        <label for="userPassword">User Password</label>
    </div>
    <button class="btn waves-effect waves-light"
type="submit">REGISTER</button>
    </form>
</div>
</div>
<script
src="https://cdnjs.cloudflare.com/ajax/libs/materialize/1.0.0/js/materializ
e.min.js"></script>
<script>
    // Initialize Materialize dropdown
    document.addEventListener('DOMContentLoaded', function() {
        var elems = document.querySelectorAll('select');
        var instances = M.FormSelect.init(elems);
    });
</script>
</body>
</html>

```

## view-center.jsp

```

<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>

<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>Center Registration</title>
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/materialize/1.0.0/css/material
ize.min.css">
    <style>
        .small-container {
            max-width: 400px;
            margin: 0 auto;
        }
    </style>
</head>
<body>
    <h6 align="center">CENTER INFORMATION</h6>
    <div class="container small-container">

```

```

        <div class="card-panel">
            <h4 align="center" style="color:
green;">${center.centerName}</h4>
            <table>
                <tr>
                    <th>Center ID</th>
                    <td>${center.centerID}</td>
                </tr>
                <tr>
                    <th>Center City</th>
                    <td>${center.centerCity}</td>
                </tr>
            </table>
        </div>
    </div>
    <br><br>
    <h6 align="center">CITIZENS IN ${center.centerName}</h6>
    <div class="container small-container">
        <div class="card-panel">
            <table>
                <thead>
                    <tr>
                        <th>Citizen ID</th>
                        <th>Citizen Name</th>
                        <th>ACTION</th>
                    </tr>
                </thead>
                <tbody>
                    <c:forEach var="citizen" items="${citizens}">
                        <tr>
                            <td>${citizen.citizenID}</td>
                            <td>${citizen.citizenName}</td>
                            <td><a href="view-
citizen?citizenID=${citizen.citizenID}">VIEW</a></td>
                        </tr>
                    </c:forEach>
                </tbody>
            </table>
        </div>
    </div>
    <script
src="https://cdnjs.cloudflare.com/ajax/libs/materialize/1.0.0/js/materializ
e.min.js"></script>
    <script>
        // Initialize Materialize dropdown
        document.addEventListener('DOMContentLoaded', function() {
            var elems = document.querySelectorAll('select');
            var instances = M.FormSelect.init(elems);
        });
    </script>
    <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
    <script>
        // Get the value of the centerCity from the model
        var centerCityValue = "${center.centerCity}";

        // Set the selected option based on the centerCity value
        $("#centerCity").val(centerCityValue);
    </script>
</body>
</html>

```

## view-center-details.jsp

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

<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>Center Registration</title>
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/materialize/1.0.0/css/material
ize.min.css">
    <style>
        .small-container {
            max-width: 400px;
            margin: 0 auto;
        }
    </style>
</head>
<body>
    <h6 align="center">CENTER INFORMATION</h6>
    <div class="container small-container">
        <div class="card-panel">
            <table>
                <c:forEach var="center" items="${centers}">
                    <tr>
                        <th>Center ID</th>
                        <td>${center.centerID}</td>
                    </tr>
                    <tr>
                        <th>Center Name</th>
                        <td>${center.centerName}</td>
                    </tr>
                    <tr>
                        <th>Center City</th>
                        <td>${center.centerCity}</td>
                    </tr>
                </c:forEach>
            </table>
        </div>
    </div>
    <br><br>
</body>
</html>
```

## view-citizen.jsp

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

```

<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Center Registration</title>
  <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/materialize/1.0.0/css/material
ize.min.css">
  <style>
    .small-container {
      max-width: 400px;
      margin: 0 auto;
    }
  </style>
</head>
<body>
  <h6 align="center">CITIZEN INFORMATION</h6>
  <div class="container small-container">
    <div class="card-panel">
      <table>
        <tr>
          <th>Citizen Name</th>
          <th>${citizen.citizenName}</th>
        </tr>
        <tr>
          <th>Citizen ID</th>
          <td>${citizen.citizenID}</td>
        </tr>
        <tr>
          <th>Citizen City</th>
          <td>${citizen.citizenCity}</td>
        </tr>
        <tr>
          <th>Number of vaccines</th>
          <td>${citizen.citizenDose}</td>
        </tr>
        <tr>
          <th>Vaccination Status</th>
          <td>${citizen.citizenVaccinationStatus}</td>
        </tr>
        <tr>
          <th>Alloted Vaccination Center</th>
          <td>
            <a href="view-center-
details?centerName=${citizen.center.centerName}">
              ${citizen.center != null ?
citizen.center.centerName : 'N/A'}
            </a>
          </td>
        </tr>
      </table>
    </div>
  </div>
  <br><br>

  <script
src="https://cdnjs.cloudflare.com/ajax/libs/materialize/1.0.0/js/materializ
e.min.js"></script>
  <script>
    // Initialize Materialize dropdown
    document.addEventListener('DOMContentLoaded', function() {

```

```

        var elems = document.querySelectorAll('select');
        var instances = M.FormSelect.init(elems);
    });
</script>
<script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
</body>
</html>

```

## pom.xml

```

<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
https://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>
    <parent>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-parent</artifactId>
        <version>3.1.0</version>
        <relativePath/> <!-- lookup parent from repository -->
    </parent>
    <groupId>com.ecommerce</groupId>
    <artifactId>spring-boot-mvc-demo</artifactId>
    <version>0.0.1-SNAPSHOT</version>
    <name>spring-boot-mvc-demo</name>
    <description>Demo project for Spring Boot</description>
    <properties>
        <java.version>17</java.version>
    </properties>
    <dependencies>
        <!-- JSP support in Spring -->
        <dependency>
            <groupId>org.apache.tomcat.embed</groupId>
            <artifactId>tomcat-embed-jasper</artifactId>
            <scope>provided</scope>
        </dependency>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-web</artifactId>
        </dependency>

        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-test</artifactId>
            <scope>test</scope>
        </dependency>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-data-jpa</artifactId>
        </dependency>
        <dependency>
            <groupId>com.h2database</groupId>
            <artifactId>h2</artifactId>
            <version>1.4.200</version> <!-- Replace with the desired version --
>

```

```
</dependency>
<dependency>
    <groupId>jakarta.servlet.jsp.jstl</groupId>
    <artifactId>jakarta.servlet.jsp.jstl-api</artifactId>
    <version>3.0.0</version>
</dependency>
<dependency>
    <groupId>com.mysql</groupId>
    <artifactId>mysql-connector-j</artifactId>
    <scope>runtime</scope>
</dependency>
</dependencies>

<build>
    <plugins>
        <plugin>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-maven-plugin</artifactId>
        </plugin>
    </plugins>
</build>

</project>
```