Vaccination Center

Course-end Project 1

DESCRIPTION

Build an application to manage vaccination centers working on distributing vaccines to all citizens of India

Sourcecode

Controllers

Citizencontroller

```
package com.ecommerce.controllers;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.DeleteMapping;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.ModelAttribute;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.PutMapping;
import org.springframework.web.bind.annotation.RequestBody;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.ResponseBody;
import org.springframework.web.bind.annotation.RestController;
import org.springframework.web.servlet.ModelAndView;

import com.ecommerce.entity.Citizen;
import com.ecommerce.entity.VaccinationCentre;
import com.ecommerce.repository.CitizenRepository;
import com.ecommerce.repository.VaccinationCentreRepository;
@RestController
@RequestMapping("/citizens")
//@Controller
public class CitizenController {

    @Autowired
     CitizenRepository citizenRepository;
```

```
@Autowired
        VaccinationCentreRepository vaccinationCentreRepository;

@GetMapping
public ModelAndView showCitizens() {
    ModelAndView modelAndView = new ModelAndView("citizens");
    List<Citizen> citizens = citizenRepository.findAll();
    int totalCitizens = citizens.size();
    modelAndView.addObject("citizens", citizens);
    modelAndView.addObject("totalCitizens", totalCitizens);
    return modelAndView;
}



@GetMapping("/add-citizen")
public String showAddCitizenForm(Model model) {
    List<String> cities = vaccinationCentreRepository.getAllCities();
    List<String> vaccineCentres= vaccinationCentreRepository.findAllCentres();
    model.addAttribute("cities", cities);
    model.addAttribute("vaccineCentres", vaccineCentres);
    model.addAttribute("centre", new VaccinationCentre());
    model.addAttribute("citizen", new Citizen());
    return "add-citizen";
}

@PostMapping("/add-citizen")
public String addCitizen(@ModelAttribute("citizen") Citizen citizen) {
        citizen.setNumberOfDoses(0);
    citizen.setVaccinationStatus("Not Vaccinated");

    citizenRepository.save(citizen);


    return "redirect:/citizensuccess";
}

@GetMapping("/citizensuccess")
public String showCitizenSuccessPage() {
    return "citizensuccess";
}


@GetMapping("/edit-citizen/{id}")
public String showEditCitizenForm(@PathVariable("id") Long id, Model model) {
    Citizen citizen = citizenRepository.findById(id).orElse(null);
    List<String> vaccineCentres = vaccinationCentreRepository.findAllCentres();
    List<String> cities = vaccinationCentreRepository.getAllCities();

    model.addAttribute("vaccineCentres", vaccineCentres);
    model.addAttribute("cities", cities);
    model.addAttribute("citizen", citizen);

    return "edit-citizen";
}
@PostMapping("/edit-citizen/{id}")
public String editCitizen(@ModelAttribute("citizen") Citizen citizen) {

          if (citizen.getNumberOfDoses() == 1) {
        citizen.setVaccinationStatus("Partially Vaccinated");
    } else if (citizen.getNumberOfDoses() == 2) {
        citizen.setVaccinationStatus("Fully Vaccinated");
    } else {
        citizen.setVaccinationStatus("Not Vaccinated");
    }

    citizenRepository.save(citizen);

    return "editsuccess";
}


@GetMapping("/view-citizen/{id}")
public String viewCitizen(@PathVariable("id") Long id, Model model) {
    Citizen citizen = citizenRepository.findById(id).orElse(null);

    model.addAttribute("citizen", citizen);
    return "citizenview";
}

@GetMapping("/delete-citizen/{id}")
public String deleteCitizen(@PathVariable("id") Long id) {
    citizenRepository.deleteById(id);
    return "citizendelete";
}

@ResponseBody
@GetMapping("/api/citizens/{id}")
public Citizen getCitizenById(@PathVariable("id") Long id) {
    return citizenRepository.findById(id).orElse(null);
}


@ResponseBody
```

```
    @GetMapping("/api/citizens/all")
    public List<Citizen> getAllCitizens() {
      return citizenRepository.findAll();
    }



    // Create a new citizen
    @PostMapping("/api/citizens")
    public Citizen createCitizen(@RequestBody Citizen citizen) {
      citizen.setNumberOfDoses(0);
      citizen.setVaccinationStatus("Not Vaccinated");
      return citizenRepository.save(citizen);
    }

    // Update an existing citizen by ID
    @PutMapping("/api/citizens/{id}")
    public Citizen updateCitizen(@PathVariable("id") Long id, @RequestBody Citizen updatedCitizen) {
      Citizen existingCitizen = citizenRepository.findById(id).orElse(null);
      if (existingCitizen != null) {
        existingCitizen.setName(updatedCitizen.getName());
        existingCitizen.setCity(updatedCitizen.getCity());
        // Set other attributes as needed
        citizenRepository.save(existingCitizen);
      }
      return existingCitizen;
    }

    // Delete a citizen by ID
    @DeleteMapping("/api/citizens/{id}")
    public String deleteCitizenById(@PathVariable("id") Long id) {
      citizenRepository.deleteById(id);
      return "Citizen with ID " + id + " has been deleted.";
    }
  }
```

VaccineCenterController

```
package com.ecommerce.controllers;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.*;
import org.springframework.web.servlet.ModelAndView;

import com.ecommerce.repository.CitizenRepository;
import com.ecommerce.repository.VaccinationCentreRepository;
import com.ecommerce.entity.*;

import java.util.List;

@RestController
@RequestMapping("/vaccinationcentre")
public class VaccinationCentreController {

    @Autowired
    VaccinationCentreRepository vaccinationCentreRepository;

    @Autowired
    CitizenRepository citizenRepository;

    @GetMapping
    public ModelAndView showVaccinationCentres() {
        ModelAndView modelAndView = new ModelAndView("vaccinationcentre");
        List<VaccinationCentre> centres = vaccinationCentreRepository.findAll();
        modelAndView.addObject("vaccinecentre", centres);
        return modelAndView;
    }

    @GetMapping("/add-centre")
    public String showNewCentreForm(Model model) {
        List<String> cities = vaccinationCentreRepository.getAllCities();
        model.addAttribute("cities", cities);
        model.addAttribute("centre", new VaccinationCentre());
        return "newcentre";
    }

    @PostMapping("/add-centre")
    public String addCentre(@ModelAttribute("centre") VaccinationCentre centre) {
        vaccinationCentreRepository.save(centre);
        return "addsuccess";
    }

    @GetMapping("/edit-centre/{id}")
    public String showEditCentreForm(@PathVariable("id") Long id, Model model) {
        VaccinationCentre centre = vaccinationCentreRepository.findById(id).orElse(null);
        List<String> cities = vaccinationCentreRepository.getAllCities();
        model.addAttribute("centre", centre);
        model.addAttribute("cities", cities);
        return "editcentre";
    }

    @PostMapping("/edit-centre/{id}")
    public String editCentre(@ModelAttribute("centre") VaccinationCentre centre) {
        vaccinationCentreRepository.save(centre);
        return "updatesuccess";
    }

    @GetMapping("/delete-centre/{id}")
    public String deleteCentre(@PathVariable("id") Long id) {
        vaccinationCentreRepository.deleteById(id);
        return "deletesuccess";
    }

    @GetMapping("/view-centre/{id}")
    public String viewCentre(@PathVariable("id") Long id, Model model) {
        VaccinationCentre centre = vaccinationCentreRepository.findById(id).orElse(null);
        List<Citizen> citizens = citizenRepository.findAllById(id);
        model.addAttribute("centre", centre);
        model.addAttribute("citizens", citizens);
        return "view";
    }

    // Add the @ResponseBody annotation to indicate JSON response
    @ResponseBody
    @GetMapping("/api/vaccinationcentre/{id}")
    public VaccinationCentre getVaccinationCentreById(@PathVariable("id") Long id) {
        return vaccinationCentreRepository.findById(id).orElse(null);
    }
    @ResponseBody
    @GetMapping("/api/vaccinationcentre/all")
    public List<VaccinationCentre> getAllVaccinationCentres() {
        return vaccinationCentreRepository.findAll();
    }
    @DeleteMapping("/api/vaccinationcentre/{id}")
    public String deleteVaccinationCentreById(@PathVariable("id") Long id) {
        vaccinationCentreRepository.deleteById(id);
        return "Vaccination Centre with ID " + id + " has been deleted.";
    }
    @PutMapping("/api/vaccinationcentre/{id}")
    public VaccinationCentre updateVaccinationCentre(@PathVariable("id") Long id, @RequestBody
VaccinationCentre centre) {
        VaccinationCentre existingCentre = vaccinationCentreRepository.findById(id).orElse(null);
        if (existingCentre != null) {
            existingCentre.setName(centre.getName());
            existingCentre.setCity(centre.getCity());
            // Set other attributes as needed
            return vaccinationCentreRepository.save(existingCentre);
        }
        return null; // If vaccination centre with given ID doesn't exist
    }
    @PostMapping("/api/vaccinationcentre")
    public VaccinationCentre createVaccinationCentre(@RequestBody VaccinationCentre centre) {
        return vaccinationCentreRepository.save(centre);
    }

    // Add other JSON API methods as needed
}
```

LoginController

```
package com.ecommerce.controllers;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.RequestParam;
import org.springframework.web.servlet.ModelAndView;
import org.springframework.web.servlet.mvc.support.RedirectAttributes;

import com.ecommerce.entity.User;
import com.ecommerce.repository.UserRepository;

@Controller
public class LoginController {
    @Autowired
    private UserRepository userRepository;

    @GetMapping("/login")
    public ModelAndView showLoginForm() {
        return new ModelAndView("login");
    }

    @PostMapping("/login")
    public String loginUser(
            @RequestParam("email") String email,
            @RequestParam("password") String password,
            RedirectAttributes redirectAttributes) {

        // Query the database to check if the credentials are valid
        User user = userRepository.findByEmailAndPassword(email, password);
        if (user != null) {
            redirectAttributes.addFlashAttribute("message", "Login successful!");
            return "redirect:/vaccinationcentre";
        } else {
            redirectAttributes.addFlashAttribute("error", "Invalid credentials,User not found!");
            return "redirect:/login";
        }
```

```
  }
}
```

RegistartionController

```
package com.ecommerce.controllers;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.RequestParam;
import org.springframework.web.servlet.ModelAndView;
import org.springframework.web.servlet.mvc.support.RedirectAttributes;

import com.ecommerce.repository.UserRepository;
import com.ecommerce.entity.User;
@Controller
public class RegistrationController {
    @Autowired
    private UserRepository userRepository;

    @GetMapping("/registration")
    public ModelAndView showRegistrationForm() {
        return new ModelAndView("registration");
    }

    @PostMapping("/registration")
    public String registerUser(
            @RequestParam("firstName") String firstName,
            @RequestParam("lastName") String lastName,
            @RequestParam("email") String email,
            @RequestParam("password") String password,
            RedirectAttributes redirectAttributes) {

        User user = new User();
        user.setFirstName(firstName);
        user.setLastName(lastName);
        user.setEmail(email);
        user.setPassword(password);

        userRepository.save(user);

        redirectAttributes.addFlashAttribute("message", "Registration successful!");
        return "redirect:/login";
    }
}
```

Repository

VaccinecenterRepository

```java
package com.ecommerce.repository;

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

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 com.ecommerce.entity.Citizen;
import com.ecommerce.entity.VaccinationCentre;

@Repository
public interface VaccinationCentreRepository extends JpaRepository<VaccinationCentre, Long> {
        List<VaccinationCentre> findAll();

        @Query("SELECT DISTINCT city FROM VaccinationCentre")
   List<String> getAllCities();

        Optional<VaccinationCentre> findById(Long id);

         @Query("SELECT DISTINCT name FROM VaccinationCentre")
           List<String> findAllCentres();



}
```

CitizenRepository

```java
package com.ecommerce.repository;

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

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 com.ecommerce.entity.Citizen;
import com.ecommerce.entity.VaccinationCentre;

@Repository
public interface CitizenRepository extends JpaRepository<Citizen, Long> {
        List<Citizen> findAll();

        List<Citizen> findAllById(Long id);
}
```

Userrepository

```
package com.ecommerce.repository;
import org.springframework.data.jpa.repository.Query;
import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.stereotype.Repository;

import com.ecommerce.entity.User;
@Repository
public interface UserRepository extends JpaRepository<User, Long> {
        @Query("SELECT u FROM User u WHERE u.email = ?1 AND u.password = ?2")
  User findByEmailAndPassword(String email, String password);
}
```

Entity

User

```
package com.ecommerce.entity;

import jakarta.persistence.*;

@Entity
@Table(name="user")
public class User {
  @Id
  @GeneratedValue(strategy = GenerationType.IDENTITY)
  private Long id;

  private String firstName;

  private String lastName;

  private String email;
private String password;


public User() {

}


    public Long getId() {
        return id;
    }


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


    public String getFirstName() {
        return firstName;
    }


    public void setFirstName(String firstName) {
        this.firstName = firstName;
    }


    public String getLastName() {
        return lastName;
    }


    public void setLastName(String lastName) {
        this.lastName = lastName;
    }


    public String getEmail() {
        return email;
    }


    public void setEmail(String email) {
        this.email = email;
    }
```

```
	public String getPassword() {
		return password;
	}


	public void setPassword(String password) {
		this.password = password;
	}


}
```

Citizen

```
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.JoinColumn;
import jakarta.persistence.ManyToOne;
import jakarta.persistence.Table;

@Entity
@Table(name = "citizen")
public class Citizen {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    @Column(name = "citizen_id")
    private Long id;

    private String name;
    private String city;

    @Column(name = "number_of_doses")
    private int numberOfDoses;

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

    @Column(name = "vaccination_center")
    private String vaccinationCenter;
```

```
public Citizen() {
}

    public Long getId() {
        return id;
    }

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

    public String getName() {
        return name;
    }

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

    public String getCity() {
        return city;
    }

    public void setCity(String city) {
        this.city = city;
    }

    public int getNumberOfDoses() {
        return numberOfDoses;
    }

    public void setNumberOfDoses(int numberOfDoses) {
        this.numberOfDoses = numberOfDoses;
    }

    public String getVaccinationStatus() {
        return vaccinationStatus;
    }

    public void setVaccinationStatus(String vaccinationStatus) {
        this.vaccinationStatus = vaccinationStatus;
    }

    public String getVaccinationCenter() {
        return vaccinationCenter;
    }
```

```
        public void setVaccinationCenter(String vaccinationCenter) {
                this.vaccinationCenter = vaccinationCenter;
        }


}
```

vaccinecenter

```
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.JoinColumn;
import jakarta.persistence.Table;

@Entity
@Table(name = "vaccinecentre")
public class VaccinationCentre {

   @Id

   @GeneratedValue(strategy = GenerationType.IDENTITY)


   private Long id;

   private String name;

   private String city;


 public VaccinationCentre() {

   }



        public Long getId() {
                return this.id;
        }

        public void setId(Long id) {
                this.id = id;
        }
	public String getName() {
		return this. name;
	}

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

	public String getCity() {
		return this.city;
	}

	public void setCity(String city) {
		this.city = city;
	}


}
```

Applicationproperties

```
server.port=8889

spring.mvc.view.prefix=/WEB-INF/views/
spring.mvc.view.suffix=.jsp
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
spring.datasource.url=jdbc:mysql://localhost:3306/vaccineCenter
spring.datasource.username=root
spring.jpa.hibernate.ddl-auto=update
spring.datasource.password=Anusuyamanju@123
spring.jpa.database-platform=org.hibernate.dialect.MySQL8Dialect
spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQL8Dialect

spring.jpa.show-sql=true
spring.jpa.properties.hibernate.formate_sql=true;
```

JSP pages

Login

```
<!DOCTYPE html>
<html>
<head>
    <title>Login</title>
    <link rel="stylesheet" type="text/css"
href="${pageContext.request.contextPath}/css/style.css">
    <style>
        .center {
            text-align: center;
        }
    </style>
</head>
<body>
    <h1 class="center">Login</h1>

    <c:if test="${not empty message}">
        <p class="center">${message}</p>
    </c:if>
    <c:if test="${not empty error}">
        <p class="center">${error}</p>
    </c:if>

    <form action="/login" method="POST" class="center">
        <label for="email">Username/Email:</label>
        <input type="email" id="email" name="email" required><br><br>

        <label for="password">Password:</label>
        <input type="password" id="password" name="password" required><br><br>

        <input type="submit" value="Login">
    </form>

    <form action="/registration" method="GET" class="center">
        <input type="submit" value="Register">
    </form>
</body>
</html>
```

Registration

```
<!DOCTYPE html>
<html>
<head>
    <title>Registration</title>
    <link rel="stylesheet" type="text/css"
href="${pageContext.request.contextPath}/css/style.css">
<style>
        .center {
            text-align: center;
        }
    </style>
</head>
<body>
    <h1 class="center">Registration</h1>
    <form action="/registration" method="POST" class="center">
        <label for="firstName">First Name:</label>
        <input type="text" id="firstName" name="firstName" required><br><br>
        <label for="lastName">Last Name:</label>
        <input type="text" id="lastName" name="lastName" required><br><br>

        <label for="email">Email:</label>
        <input type="email" id="email" name="email" required><br><br>

        <label for="password">Password:</label>
        <input type="password" id="password" name="password" required><br><br>

        <input type="submit" value="Register">
    </form>
</body>
</html>
```

Citizens

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"
%>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<!DOCTYPE html>
<html>
<head>
    <title>Citizens</title>
    <link rel="stylesheet" type="text/css"
href="${pageContext.request.contextPath}/css/style.css">
    <style>
        .center {
            text-align: center;
        }
        .table {
            margin: 20px auto;
            border-collapse: collapse;
            width: 80%;
        }
        .table th,
        .table td {
            padding: 10px;
            border: 1px solid #ddd;
        }
        .table th {
            background-color: #f2f2f2;
        }
        .btn {
            display: inline-block;
            padding: 5px 10px;
            text-decoration: none;
            color: #fff;
            background-color: #333;
            border-radius: 3px;
        }
        .btn-view {
            background-color: #337ab7;
        }
        .btn-edit {
            background-color: #5cb85c;
        }
        .btn-delete {
            background-color: #d9534f;
        }
        .btn-add {
            background-color: #337ab7;
            float: right;
            margin-top: -40px;
        }
    </style>
</head>
<body>
     <h1 class="center">Citizens Dashboard</h1>

    <div class="center">
        <a href="/citizens" class="btn">Citizens</a>
        <a href="/vaccinationcentre" class="btn">Vaccination Centers</a>
        <a href="/login" class="btn">Logout</a>
    </div>

    <h2 class="center">Centers</h2>
<a href="/add-citizen" class="btn btn-add">Add new Citizen</a>

    <table class="table center">
        <thead>
            <tr>
                <th>ID</th>
                <th>Name</th>
                <th>City</th>
                <th>Number of Doses</th>
                <th>Vaccination Status</th>
                <th>Vaccination Center</th>
                <th>Action</th>
            </tr>
        </thead>
        <tbody>
            <c:forEach items="${citizens}" var="citizen">
                <tr>
                    <td>${citizen.id}</td>
                    <td>${citizen.name}</td>
                    <td>${citizen.city}</td>
                    <td>${citizen.numberOfDoses}</td>
                    <td>${citizen.vaccinationStatus}</td>
                    <td>${citizen.vaccinationCenter}</td>
                    <td>
                        <a href="/view-citizen/${citizen.id}" class="btn btn-
view">View</a>
                        <a href="/edit-citizen/${citizen.id}" class="btn btn-
edit">Edit</a>
                        <a href="/delete-citizen/${citizen.id}" class="btn btn-
delete">Delete</a>
                    </td>
                </tr>
            </c:forEach>
            <tr>
                 <td colspan="6" style="text-align: right;"><b>Total Citizens
Found:</b></td>
                <td>${totalCitizens}</td>
            </tr>
        </tbody>
    </table>
</body>
</html>
```

VaccineCenter

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>

<!DOCTYPE html>
<html>
<head>
    <title>Vaccination Center Dashboard</title>
    <link rel="stylesheet" type="text/css"
href="${pageContext.request.contextPath}/css/style.css">
    <style>
        .center {
            text-align: center;
        }
        .table {
            margin: 20px auto;
            border-collapse: collapse;
            width: 80%;
        }
        .table th,
        .table td {
            padding: 10px;
            border: 1px solid #ddd;
        }
        .table th {
            background-color: #f2f2f2;
        }
        .btn {
            display: inline-block;
            padding: 5px 10px;
            text-decoration: none;
            color: #fff;
            background-color: #333;
            border-radius: 3px;
        }
        .btn-view {
            background-color: #337ab7;
        }
        .btn-edit {
            background-color: #5cb85c;
        }
        .btn-delete {
            background-color: #d9534f;
        }
        .btn-add {
            background-color: #337ab7;
            float: right;
            margin-top: -40px;
        }
    </style>
</head>
<body>
    <h1 class="center">Vaccination Center Dashboard</h1>

    <div class="center">
        <a href="/citizens" class="btn">Citizens</a>
        <a href="/vaccinationcentre" class="btn">Vaccination Centers</a>
        <a href="/login" class="btn">Logout</a>
    </div>

    <h2 class="center">Centers</h2>
<a href="/add-centre" class="btn btn-add">Add new Center</a>
    <table class="table center">
        <thead>
            <tr>
                <th>ID</th>
                <th>Name</th>
                <th>City</th>
                <th>Action</th>
            </tr>
        </thead>
        <tbody>
            <c:forEach items="${vaccinecentre}" var="centre">
                <tr>
                    <td>${centre.id}</td>
                    <td>${centre.name}</td>
                    <td>${centre.city}</td>
                    <td>
                        <a href="/view-centre/${centre.id}" class="btn btn-view">View</a>
                        <a href="/edit-centre/${centre.id}" class="btn btn-edit">Edit</a>
                        <a href="/delete-centre/${centre.id}" class="btn btn-delete">Delete</a>
                    </td>
                </tr>
            </c:forEach>
            <tr>
                <td colspan="3" style="text-align: right;"><b>Total Centers Available:</b></td>
                <td>${vaccinecentre.size()}</td>
            </tr>
        </tbody>
    </table>


</body>
</html>
```