**Vaccination -Source code**

**vaccineCenterApplication.java:**

**package** com.ecommerce.controller;

**import** java.util.List;

**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.PathVariable;

**import** org.springframework.web.bind.annotation.PostMapping;

**import** org.springframework.web.bind.annotation.RequestMapping;

**import** com.ecommerce.entity.Citizen;

**import** com.ecommerce.entity.VaccineCenter;

**import** com.ecommerce.service.CitizenService;

**import** com.ecommerce.service.VaccineCenterService;

@Controller

@RequestMapping("/vaccine-centers")

**public** **class** VaccineCenterController {

**private** **final** VaccineCenterService vaccineCenterService;

**private** **final** CitizenService citizenService;

**public** VaccineCenterController(VaccineCenterService vaccineCenterService, CitizenService

citizenService) {

**this**.vaccineCenterService = vaccineCenterService;

**this**.citizenService = citizenService;

}

@GetMapping

**public** String getAllVaccineCenters(Model model) {

List<VaccineCenter> vaccineCenters = vaccineCenterService.getAllVaccineCenters();

model.addAttribute("vaccineCenters", vaccineCenters);

**return** "vaccine-center-details";

}

@GetMapping("/add")

**public** String showAddVaccineCenterForm(Model model) {

model.addAttribute("vaccineCenter", **new** VaccineCenter());

**return** "add-vaccine-center";

}

@PostMapping("/add")

**public** String addVaccineCenter(@ModelAttribute("vaccineCenter") VaccineCenter

vaccineCenter) {

vaccineCenterService.addVaccineCenter(vaccineCenter);

**return** "redirect:/vaccine-centers";

}

@GetMapping("/{id}")

**public** String viewVaccineCenter(@PathVariable("id") Long id, Model model) {

VaccineCenter vaccineCenter = vaccineCenterService.getVaccineCenterById(id);

List<Citizen> citizens = citizenService.getCitizensByVaccinationCenter(vaccineCenter);

model.addAttribute("vaccineCenter", vaccineCenter);

model.addAttribute("citizens", citizens);

**return** "vaccine-center-view";

}

@GetMapping("/{id}/edit")

**public** String showEditVaccineCenterForm(@PathVariable("id") Long id, Model model) {

VaccineCenter vaccineCenter = vaccineCenterService.getVaccineCenterById(id);

**if** (vaccineCenter != **null**) {

model.addAttribute("vaccineCenter", vaccineCenter);

**return** "edit-vaccine-center";

} **else** {

**return** "redirect:/vaccine-centers";

}

}

@PostMapping("/{id}/edit")

**public** String updateVaccineCenter(@PathVariable("id") Long id,

@ModelAttribute("vaccineCenter") VaccineCenter updatedVaccineCenter) {

VaccineCenter vaccineCenter = vaccineCenterService.getVaccineCenterById(id);

**if** (vaccineCenter != **null**) {

vaccineCenter.setCenter(updatedVaccineCenter.getCenter());

vaccineCenter.setCity(updatedVaccineCenter.getCity());

vaccineCenterService.updateVaccineCenter(vaccineCenter);

}

**return** "redirect:/vaccine-centers";

}

@GetMapping("/{id}/delete")

**public** String deleteVaccineCenter(@PathVariable("id") Long id) {

vaccineCenterService.deleteVaccineCenter(id);

**return** "redirect:/vaccine-centers";

}

}

**VaccinationApplication.java :**

package com.ecommerce;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class VaccinationCenterApplication {

public static void main(String[] args) {

SpringApplication.run(VaccinationCenterApplication.class, args);

}

}

**UserController.java:**

**package** com.ecommerce.controller;

**import** org.springframework.boot.autoconfigure.security.SecurityProperties.User;

**import** org.springframework.http.ResponseEntity;

**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.RequestMapping;

**import** org.springframework.web.bind.annotation.RequestParam;

**import** com.ecommerce.repository.UserRepository;

**import** com.ecommerce.service.UserService;

@Controller

@RequestMapping("/")

**public** **class** UserController {

**private** **final** UserService userService;

**private** **final** UserRepository userRepository;

**public** UserController(UserService userService, UserRepository userRepository) {

**this**.userService = userService;

**this**.userRepository = userRepository;

}

@GetMapping

**public** String showRLoginPage() {

**return** "redirect:/login.html";

}

@PostMapping("/login")

**public** String processLogin(@RequestParam("username") String username,

@RequestParam("password") String password) {

User user = userRepository.findByUsername(username);

**if** (user != **null** && user.getPassword().equals(password)) {

**return** "redirect:/vaccine-centers";

} **else** {

**return** "redirect:/errorpage.html";

}

}

@GetMapping("/register")

**public** String showRegisterPage() {

**return** "redirect:/register.html";

}

@PostMapping("/register")

**public** ResponseEntity<String> registerUser(@RequestParam("username") String username,

@RequestParam("email") String email, @RequestParam("password") String password) {

User user = **new** User();

userService.registerUser((com.example.Authentication.entities.User) user);

**return** ResponseEntity.*ok*("User registered successfully");

}

}

**CitizenController.java:**

**package** com.ecommerce.controller;

**import** org.springframework.boot.autoconfigure.security.SecurityProperties.User;

**import** org.springframework.http.ResponseEntity;

**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.RequestMapping;

**import** org.springframework.web.bind.annotation.RequestParam;

**import** com.ecommerce.repository.UserRepository;

**import** com.ecommerce.service.UserService;

@Controller

@RequestMapping("/")

**public** **class** UserController {

**private** **final** UserService userService;

**private** **final** UserRepository userRepository;

**public** UserController(UserService userService, UserRepository userRepository) {

**this**.userService = userService;

**this**.userRepository = userRepository;

}

@GetMapping

**public** String showRLoginPage() {

**return** "redirect:/login.html";

}

@PostMapping("/login")

**public** String processLogin(@RequestParam("username") String username,

@RequestParam("password") String password) {

User user = userRepository.findByUsername(username);

**if** (user != **null** && user.getPassword().equals(password)) {

**return** "redirect:/vaccine-centers";

} **else** {

**return** "redirect:/errorpage.html";

}

}

@GetMapping("/register")

**public** String showRegisterPage() {

**return** "redirect:/register.html";

}

@PostMapping("/register")

**public** ResponseEntity<String> registerUser(@RequestParam("username") String username,

@RequestParam("email") String email, @RequestParam("password") String password) {

User user = **new** User();

userService.registerUser((com.example.Authentication.entities.User) user);

**return** ResponseEntity.*ok*("User registered successfully");

}

}

**VaccinationCenterApplication .java:**

package com.ecommerce;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class VaccinationCenterApplication {

public static void main(String[] args) {

SpringApplication.run(VaccinationCenterApplication.class, args);

}

}

**Citizen.java:**

**package** com.ecommerce.entiti;

**import** com.ecommerce.entity.VaccineCenter;

**import** entity.Entity;

**import** entity.Id;

@Entity

@Table(name = "citizens")

**public** **class** Citizen {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

**private** Long id;

@Column(name = "name")

**private** String name;

@Column(name = "city")

**private** String city;

@Column(name = "no\_of\_doses")

**private** **int** noOfDoses;

@Column(name = "vaccination\_status")

**private** String vaccinationStatus;

@ManyToOne(fetch = FetchType.LAZY)

@JoinColumn(name = "vaccination\_center")

**private** VaccineCenter vaccinationCenter;

// Constructors, getters, setters, and other properties

**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** getNoOfDoses() {

**return** noOfDoses;

}

**public** **void** setNoOfDoses(**int** noOfDoses) {

**this**.noOfDoses = noOfDoses;

}

**public** String getVaccinationStatus() {

**return** vaccinationStatus;

}

**public** **void** setVaccinationStatus(String vaccinationStatus) {

**this**.vaccinationStatus = vaccinationStatus;

}

**public** VaccineCenter getVaccinationCenter() {

**return** vaccinationCenter;

}

**public** **void** setVaccinationCenter(VaccineCenter vaccinationCenter) {

**this**.vaccinationCenter = vaccinationCenter;

}

}

**CitizenRepository.java:**

package com.ecommerce.repository;

import java.util.List;

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

import com.ecommerce.entity.Citizen;

import com.ecommerce.entity.VaccineCenter;

public interface CitizenRepository extends JpaRepository {

List<Citizen> findByVaccinationCenter(VaccineCenter vaccinationCenter);

void save(Citizen citizen);

void deleteById(Long id);

Object findById(Long id);

List<Citizen> findAll();

}

**UserRepository.java:**

**package** com.ecommerce.repository;

**public** **interface** UserRepository **extends** JpaRepository<User, Long> {

org.springframework.boot.autoconfigure.security.SecurityProperties.User findByUsername(String username);

**void** save(com.example.Authentication.entities.User user);

}

**VaccineCenterRepository.java:**

**package** com.ecommerce.repository;

**import** java.util.List;

**import** com.ecommerce.entity.VaccineCenter;

**public** **interface** VaccineCenterRepository **extends** JpaRepository<VaccineCenter, Long> {

VaccineCenter save(VaccineCenter vaccineCenter);

// Add custom query methods

Object findById(Object id);

**void** delete(VaccineCenter existingCenter);

List<VaccineCenter> findAll();

}

CitizenService.java:

**package** com.ecommerce.repository;

**import** java.util.List;

**import** com.ecommerce.entity.VaccineCenter;

**public** **interface** VaccineCenterRepository **extends** JpaRepository<VaccineCenter, Long> {

VaccineCenter save(VaccineCenter vaccineCenter);

// Add custom query methods

Object findById(Object id);

**void** delete(VaccineCenter existingCenter);

List<VaccineCenter> findAll();

}

**CitizenServiceImplimentation.java:**

**package** com.ecommerce.service;

**import** java.util.List;

**import** org.springframework.beans.factory.annotation.Autowired;

**import** org.springframework.stereotype.Service;

**import** com.ecommerce.entity.Citizen;

**import** com.ecommerce.entity.VaccineCenter;

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

@Service

**public** **class** CitizenServiceImplimentation **implements** CitizenService {

@Autowired

**private** CitizenRepository citizenRepository;

@Override

**public** **void** addCitizen(Citizen citizen) {

citizenRepository.save(citizen);

}

@Override

**public** **void** updateCitizen(Citizen citizen) {

citizenRepository.save(citizen);

}

@Override

**public** **void** deleteCitizen(Long id) {

citizenRepository.deleteById(id);

}

@Override

**public** Citizen getCitizenById(Long id) {

**return** citizenRepository.findById(id).orElseThrow(() -> **new** NotFoundException("Citizen not found"));

}

@Override

**public** List<Citizen> getAllCitizens() {

**return** citizenRepository.findAll();

}

@Override

**public** List<Citizen> getCitizensByVaccinationCenter(VaccineCenter vaccineCenter) {

**return** citizenRepository.findByVaccinationCenter(vaccineCenter);

}

**public** **class** NotFoundException **extends** RuntimeException {

/\*\*

\*

\*/

**private** **static** **final** **long** ***serialVersionUID*** = 1L;

**public** NotFoundException(String message) {

**super**(message);

}

}

}

**UserService.java:**

**package** com.ecommerce.service;

**import** org.springframework.stereotype.Service;

**import** com.ecommerce.repository.UserRepository;

**import** com.example.Authentication.entities.User;

@Service

**public** **class** UserService {

**private** **final** UserRepository userRepository;

**public** UserService(UserRepository userRepository) {

**this**.userRepository = userRepository;

}

**public** **void** registerUser(User user) {

userRepository.save(user);

}

}

**VaccineCenterService.java:**

package com.ecommerce.service;

import java.util.List;

import com.ecommerce.entity.VaccineCenter;

public interface VaccineCenterService {

VaccineCenter addVaccineCenter(VaccineCenter vaccineCenter);

void updateVaccineCenter(VaccineCenter updatedVaccineCenter);

void deleteVaccineCenter(Long id);

VaccineCenter getVaccineCenterById(Long id);

List<VaccineCenter> getAllVaccineCenters();

}

**VaccineCenterServiceImplimentation.java:**

**package** com.ecommerce.service;

**import** java.util.List;

**import** org.springframework.stereotype.Service;

**import** com.ecommerce.entity.VaccineCenter;

**import** com.ecommerce.repository.VaccineCenterRepository;

@Service

**public** **class** VaccineCenterServiceImplimentation **implements** VaccineCenterService {

**private** **final** VaccineCenterRepository vaccineCenterRepository;

**public** VaccineCenterServiceImplimentation(VaccineCenterRepository vaccineCenterRepository) {

**this**.vaccineCenterRepository = vaccineCenterRepository;

}

@Override

**public** VaccineCenter addVaccineCenter(VaccineCenter vaccineCenter) {

**return** vaccineCenterRepository.save(vaccineCenter);

}

@Override

**public** **void** updateVaccineCenter(VaccineCenter updatedVaccineCenter) {

VaccineCenter existingCenter =

((Object) vaccineCenterRepository.findById(updatedVaccineCenter.getId()))

.orElseThrow(() -> **new** NotFoundException("Vaccine center not found"));

existingCenter.setCenter(updatedVaccineCenter.getCenter());

existingCenter.setCity(updatedVaccineCenter.getCity());

vaccineCenterRepository.save(existingCenter);

}

@Override

**public** **void** deleteVaccineCenter(Long id) {

VaccineCenter existingCenter = vaccineCenterRepository.findById(id)

.orElseThrow(() -> **new** NotFoundException("Vaccine center not found"));

vaccineCenterRepository.delete(existingCenter);

}

@Override

**public** VaccineCenter getVaccineCenterById(Long id) {

**return** ((Object) vaccineCenterRepository.findById(id))

.orElseThrow(() -> **new** NotFoundException("Vaccine center not found"));

}

@Override

**public** List<VaccineCenter> getAllVaccineCenters() {

**return** vaccineCenterRepository.findAll();

}

**public** **class** NotFoundException **extends** RuntimeException {

/\*\*

\*

\*/

**private** **static** **final** **long** ***serialVersionUID*** = 1L;

**public** NotFoundException(String message) {

**super**(message);

}

}

}

**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.example</groupId>

<artifactId>Vaccination\_Centre</artifactId>

<version>0.0.1-SNAPSHOT</version>

<name>Vaccination\_Centre</name>

<description>Demo project for Spring Boot</description>

<properties>

<java.version>17</java.version>

</properties>

<dependencies>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-data-jpa</artifactId>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-thymeleaf</artifactId>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-web</artifactId>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-devtools</artifactId>

<scope>runtime</scope>

<optional>true</optional>

</dependency>

<dependency>

<groupId>com.mysql</groupId>

<artifactId>mysql-connector-j</artifactId>

<scope>runtime</scope>

</dependency>

<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-test</artifactId>

<scope>test</scope>

</dependency>

</dependencies>

<build>

<plugins>

<plugin>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-maven-plugin</artifactId>

</plugin>

</plugins>

</build>

</project>