CI/CD Deployment for Springboot Application..

Course-end Project 5

Description

Project Objective:

As a Full Stack Developer, you have to build a CI/CD pipeline to demonstrate continuous deployment and host the application on AWS EC2 instance.

Sourcecode

Jenkins pipeline script

```
pipeline{
  agent any
  tools{
      maven 'maven'
  }
  stages{
      stage('build maven war file'){
          steps{
           checkout scmGit(branches: [[name: '*/master']], extensions: [], userRemoteConfigs:
[[credentialsId: 'add3a6cc-aec7-4e37-a947-62384fd54448', url:
'https://github.com/navyamanju/Spring.git']])
            bat 'mvn clean install'
          }
      }
      stage('build docker image'){
          steps{
              script{
                 bat 'docker build -t navyamanju/awshost1 .'
              }
          }
      }
      stage('push the docker image to docker hub'){
          steps{
              script{

                   withCredentials([string(credentialsId: 'dockerpassword1', variable: 'dockerpasskey')])  {

                    bat 'docker login -u navyamanju -p %dockerpasskey%'
                }
            bat 'docker push navyamanju/awshost1'
          }
        }
      }

 }
}
```

Dockerfile

```
FROM openjdk:17
EXPOSE 8889
ADD  target/VaccineManagementSystem-0.0.1-SNAPSHOT.jar VaccineManagementSystem-0.0.1-SNAPSHOT.jar
ENTRYPOINT [ "java","-jar","/VaccineManagementSystem-0.0.1-SNAPSHOT.jar"]
```

Spring boot application

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.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.servlet.ModelAndView;

import com.ecommerce.entity.Citizen;
import com.ecommerce.entity.VaccinationCentre;
import com.ecommerce.repository.CitizenRepository;
import com.ecommerce.repository.VaccinationCentreRepository;

@Controller
public class CitizenController {

   @Autowired
    CitizenRepository citizenRepository;
```

```
@Autowired
        VaccinationCentreRepository vaccinationCentreRepository;

@GetMapping("/citizens")
public String showCitizens(Model model) {
    List<Citizen> citizens = citizenRepository.findAll();
    int totalCitizens = citizens.size();
    model.addAttribute("citizens", citizens);
    model.addAttribute("totalCitizens", totalCitizens);
    return "citizens";
}


@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";
    }



}
```

Vaccinecentercontroller

```
package com.ecommerce.controllers;
import java.util.Optional;
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.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 org.springframework.web.bind.annotation.RequestParam;
import org.springframework.web.servlet.ModelAndView;

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

import com.ecommerce.entity.*;

@Controller
public class VaccinationCentreController {


        @Autowired
         VaccinationCentreRepository vaccinationCentreRepository;

              @Autowired
              CitizenRepository citizenRepository;

             /*
              * @Autowired private VaccinationCentreService vaccinationCentreService;
              */

  @GetMapping("/vaccinationcentre")
  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";
}
```

}

Index.jsp

```
<!DOCTYPE html>
<html>
<head>
    <title>Vaccination Management</title>
    <link rel="stylesheet" href="/css/style.css">
    <style>
        .main-heading {
            text-align: center;
            margin-bottom: 20px;
        }

        .subheading {
            text-align: center;
            margin-bottom: 20px;
        }

        .button-container {
    display: flex;
    flex-direction: column;
    align-items: center;
    margin-top: 20px;
}

.login-button {
    background-color: green;
    color: white;
    padding: 10px 20px;
    border: none;
    text-decoration: none;
    cursor: pointer;
    margin-bottom: 10px;
}

.registration-button {
    background-color: red;
    color: white;
    padding: 10px 20px;
    border: none;
    text-decoration: none;
    cursor: pointer;
    margin-top: 10px;
}

    </style>
</head>
<body>
    <h1 class="main-heading">Welcome to Vaccination Management</h1>

    <div class="button-container">
```

```
        <form action="/login" method="GET">
            <button type="submit" class="login-button">If you are already registered,
Login Here</button>
        </form>
        <form action="/registration" method="GET">
            <button type="submit" class="registration-button">If you are a new user,
Register Here</button>
        </form>
    </div>
</body>
</html>
```