

Deploy Application on Cloud

Writeup

Steps

1. Start
2. Create a Spring Boot Project with dependencies – Spring Web, Spring JPA, My SQL, Dev Tools, Lombok

3. **Entities:**

Create a package for entities and create the following entities

- AdminUser
- Citizen
- VaccinationCenter
- City

Between Citizen and VaccinationCenter, maintain Many-to-One relationship between Citizen and VaccinationCenter as many citizens can be associated with a center

4. The cities are added manually by the Admin from the backend.

5. **Repositories:**

- Create a package for Repository Interfaces for all the four entities.
- The respective interfaces extend JpaRepository.
- Create custom queries and annotate them using @Query.

6. **Service Interfaces:**

Create a package for Service Interfaces for all the four entities.

7. **Service Implementations:**

- Create a package for Service Implementation Classes for all the four entities.
- Annotate the classes with `@Service` and then use `@Autowired` to create an object of the respective repository interfaces.

8. Controllers:

- Create a package for Controller classes for these three entities
 - AdminUser
 - Citizen
 - VaccinationCenter
- Controller for city is not created, as the cities are added from Backend in MySQL by the admin.
- The controller classes are annotated with `@Controller` tags.
- The AdminUserController contains the request mapping which leads to the base/home page. It has various checks and methods to handle the login and register scenarios – like, while registering, checks for duplicate email and username and while logging in, checks for incorrect password and non-existing usernames. When a user logs in, a session object is created and username is added to it.
- The CitizenController class has `@RequestMapping("/citizens")` in the beginning followed by the actions like, add, view, edit and delete in the methods with corresponding mappings.

While adding and editing citizen details, an extra intermediate method is used. This is used so that first the required city is selected, then only the centers available in that city are displayed to the Admin user.

- The VaccinationCenterController class has

@RequestMapping("/vaccinationcenter") in the beginning followed by the actions like, add, view, edit and delete in the methods with corresponding mappings.

As the Citizen entity is dependent on the VaccinationCenter entity, whenever a center is prompted to be deleted, first its corresponding citizen information is deleted then the center information is deleted.

- ModelAndView objects are used to go to the required pages/ redirect to the methods within the controller.

9. JSPs:

- While registering and logging in, the fields, username, email ID and password are made mandatory so that null values don't get associated with them.
- For each of the controller(s) actions, a JSP is provided to display the result as well as perform operations of passing the information to the controller.
- When the Admin User performs all the actions and finally prompts to log out, the session object is invalidated first in the logout.jsp and then it leads to the home page.
- A security JSP file is added and it's included in all the JSPs to ensure that once the session is invalidated, the user can't press the back button in the browser and go back to the operations/view pages. It redirects the user to the home page.

10. Install docker desktop.

11. Create a Dockerfile in the spring boot project and add the Java version, port number, war file name and command to be executed.

12. In the application.properties, mention the name of the container that will act as the host for MySQL.
13. Create an image with the spring boot application and push it to docker hub.
14. On AWS, create an EC2 instance with required security groups and storage.
15. On the EC2 console, install docker and pull the docker hub image uploaded and mysql.
16. Create the initial city table.
17. Create a container of the same name mentioned in application.properties and mention the same port number.
18. The application can now be accessed using the public IPv4 DNS URL with the appropriate port number.
19. End