E-Channelling Website

MEDICARE

Introduction

I Developed a Spring Boot Application called Medicare. Actually it's an E-Channelling website. The main service that users can get from this service is place an appointment to doctor and manage those appointments.

To do that I used 4 user categories in my application. And there are 8 servicers.

They are,

- **Visitors**: These are the un registered users.
- **Clients/Users**: These are registered users that can place an appointment.
- **Manager**: These users are belonging to their hospital. And their main responsibility is managing hospital channellings.
- **Admin**: This user can manage hospitals and can add managers to hospital and etc.

Here are my servicers.

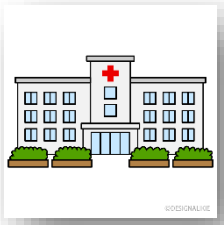
Category

This is the smallest service in my application. The main responsibility of this service is to manage Doctor's categories. Like General, Dentist, Neurologist, etc.

Doctor

I used this service to manage Doctor's and their categories (Which doctor have which categories).

Hospital



From this service I managed hospitals.

Channelling

This is a one of main servicers in my application. I used this to manage channellings. Like which doctor available in which hospital.

Appointment

This is also a one of main service in my application, from this I managed user's appointments.

Client

I used this service to manage all users of my application(Client's/User's/Managers/Admin)

OAuth

I managed all logins from this service. And I used password credentials flow.

Discovery

I used Eureka Discovery Service too.

How I start?

1. **Backend Development**
    - **a.** In this step I started to develop my micro-servicers. I started to develop them from simple servicers those are not tightly depend on other servicers. Like Category, Hospital, Doctor.
    - **b.** I developed all CURD operations that I want and did validations.
    - **c.** I tested developed servicers by using insomnia.
2. **Frontend Development**
    - **a.** I developed frontend angular app for my backend servicers. I developed only components for my micro-servicers.
    - **b.** And I did frontend validations too.
3. **Finalize**
    - a. In between step 1st and 2nd I developed Oauth2 service and User service when they needed. Because its little bit hard to test with Oauth2.
    - b. In this step I created user views like Home page, Profile page, Manager page, etc.
    - c. I create Discovery Service and put my servicers to Dockers too,
    - d. And I tested my application and fixed bugs.
    - e. In step it's so easy for me to create user views, test and bug fixing because I have already created components. So what I did is, place them in a correct place that I want. Testing and bug fixing is also easy this for me on this step because I already did most of them in 1st and 2nd steps.

What I used for this application?

- Maven
- Spring boot
- MySql
- Spring Data JPA
- Spring Mail
- OAuth2
- Eureka Discovery Service
- Angular
- Docker