# Doc-Appoint

This is a complete end to end Doctor’s calendar and patient management tool. It complements doctors to stay active on a patient while managing the prescriptions, appointment history and doctor’s calendar.

Below are features:

## For Doctors

* Manage Patients
* Manage prescriptions
* Manage Appointment time with modifying and cancelling capability

## For Patients

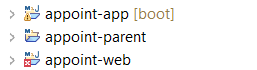
* Manage appointments
* View and download prescriptions
* View doctor’s availability

## Pre-requisites

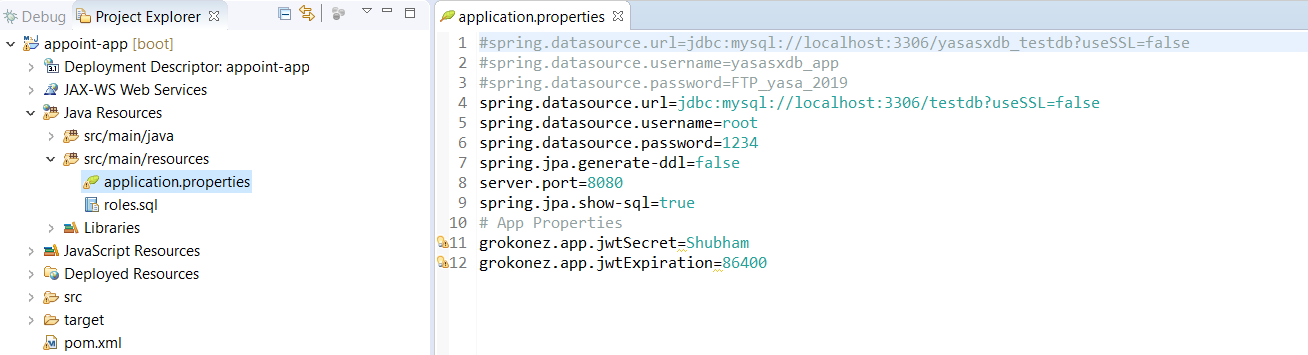
* Java 8
* Angular 8 CLI
* Node
* NPM
* Eclipse
* VSCode
* MySQL (Or any DB of choice)

## Installation

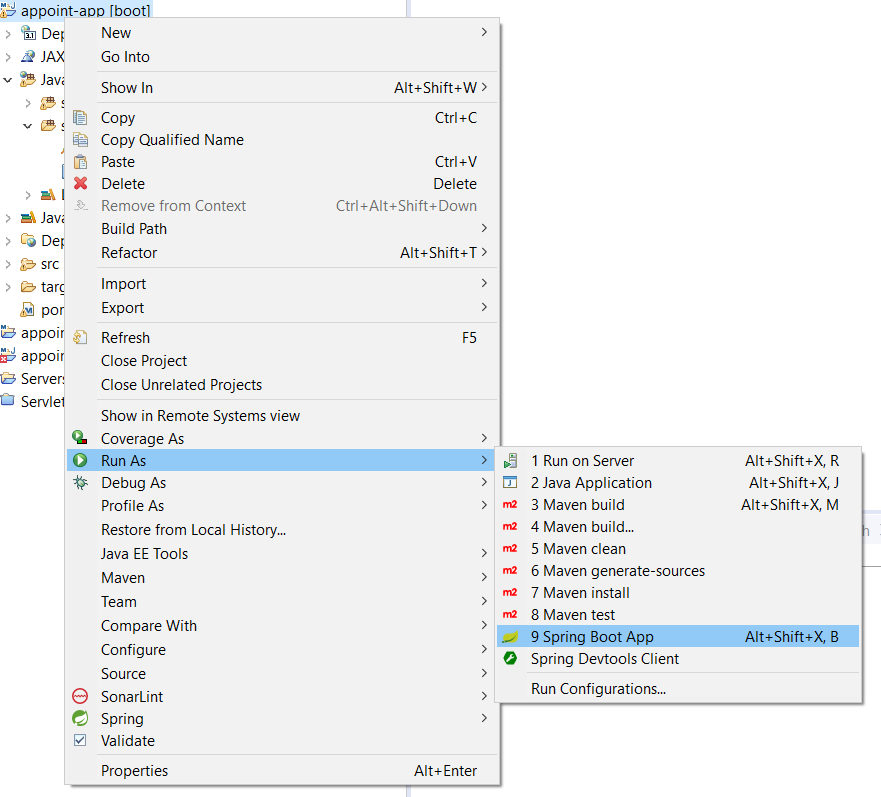
1. This project contains 3 Project. Top one is parent project which is responsible for packing the other two (front end and back end) into 1 single war file which can then be deployed on any tomcat installation.
2. Import the parent folder in eclipse as maven project.
3. This will take some time to build initial setup.
4. Once successfully imported the project setup will look like this



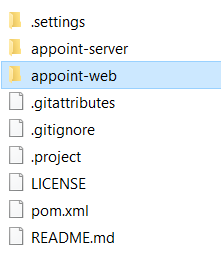
1. In the application.properties, change the configuration of DB connections:

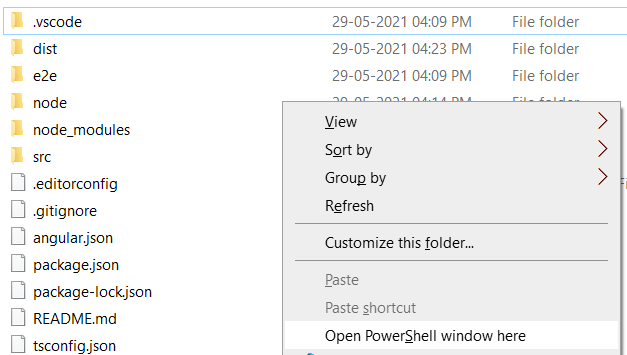


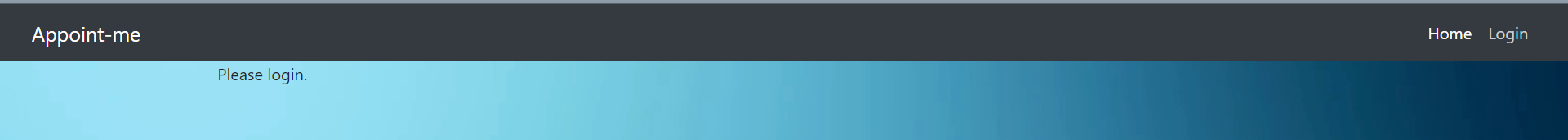
1. Run all the queries from roles.sql in your DB.
2. Right click on appoint-app and run the backend. (Ensure the DB is running and connectable)

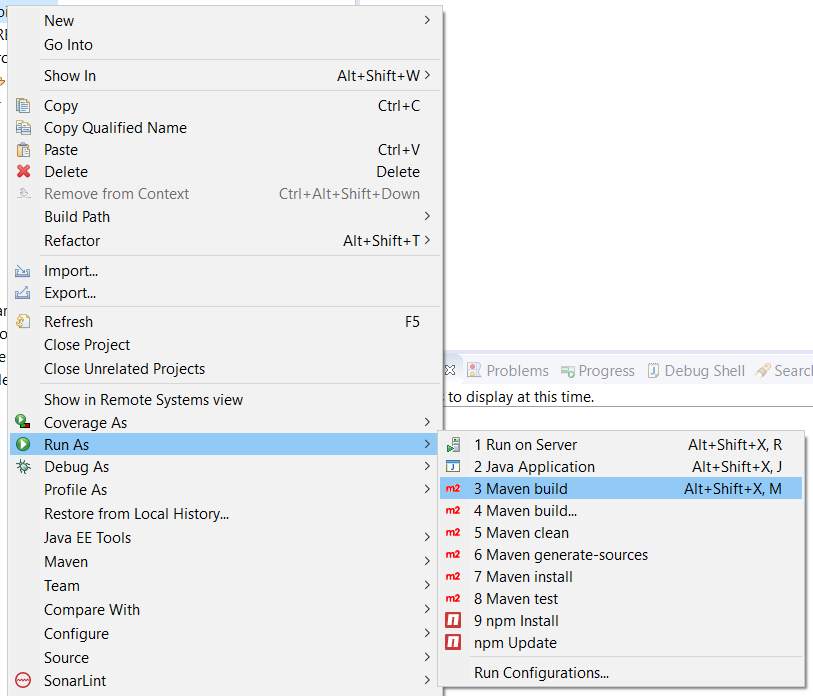


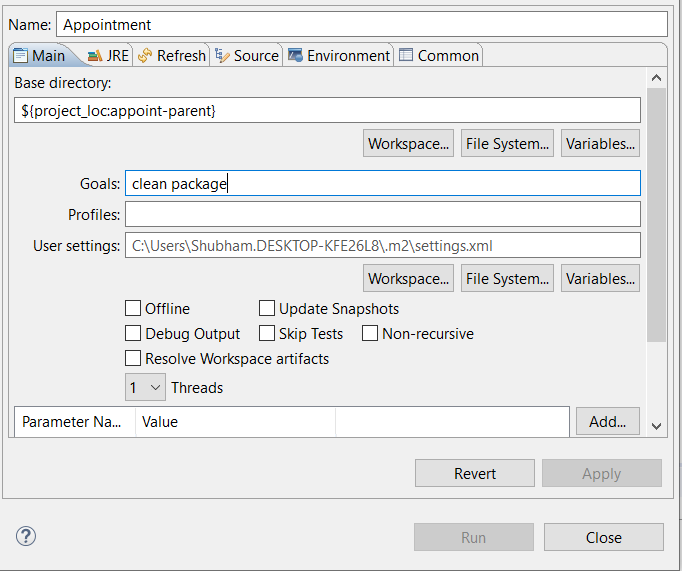
1. Now the backend should be up and running. Go to file explorer and open appoint-web\src\main\web folder



1. While holding shift key, right click anywhere in the folder and click open powershell/cmd window here
2. Once the cmd is open, run
3. npm install
4. ng serve -o
5. This will start the local node server to start on port 4200. You can access the application on <http://localhost:4200/>



1. Once on homescreen, click login > create new user for patient.
2. For Doctor’s user, use MySQL DB to insert User credentials in db since, this is not yet implemented.
3. Right click on appoint-parent folder
4. In the edit configuration, enter goals and click run:



1. Once build is successful, you will have a war file which can then be deployed on any tomcat installation.