**Documentation for Spring Boot & Angular Application**

**Spring Boot Application:**

**Features:**

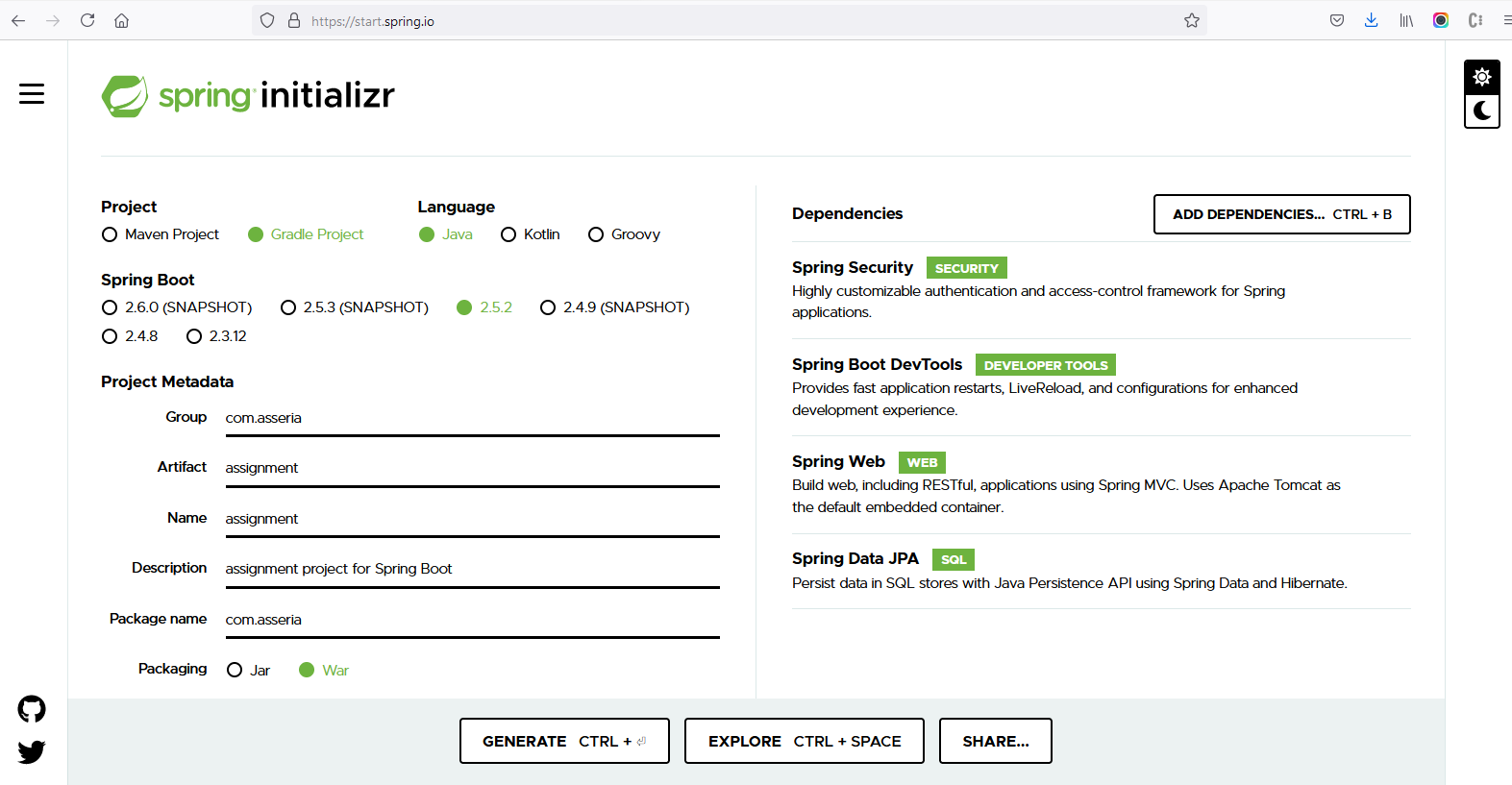
* Jwt token authentication
* Mysql database
* Jpa (hibernate)
* Centrally logging using AOP
* Swagger-ui documentation
* Spring Data Specifications for search
* Session handling using jwt token
* Mapper implementation for dto to entity and entity to dto conversions
* Customize Exception handling

**Development:**

This application is developed in intellij idea tool. This tool is specially developed for spring boot development and very similar to eclipse as well it provide internal services like maven, databases connectivity and much more.

Application is developed from spring boot online application creating service

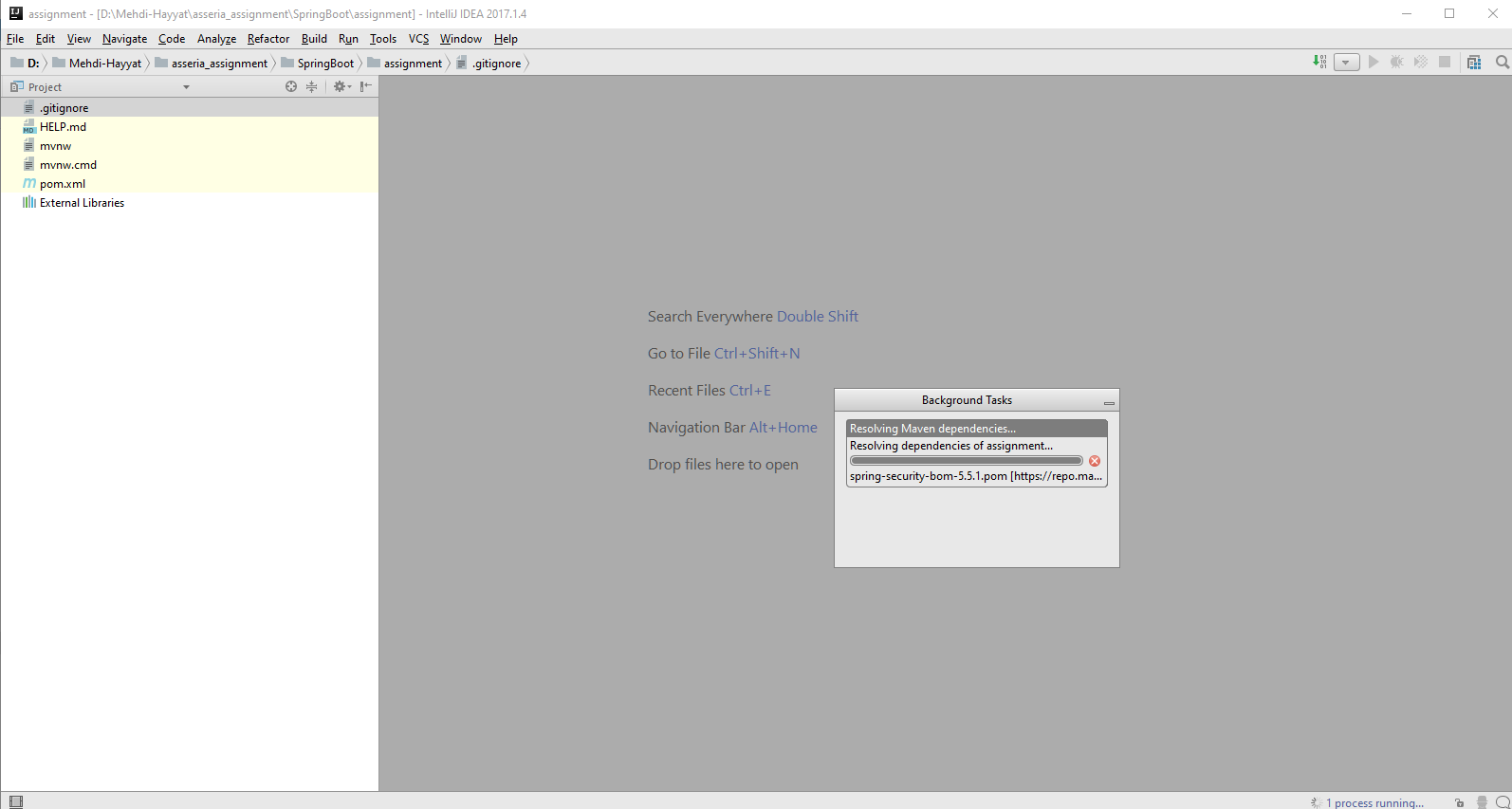
[**https://start.spring.io/**](https://start.spring.io/)

****

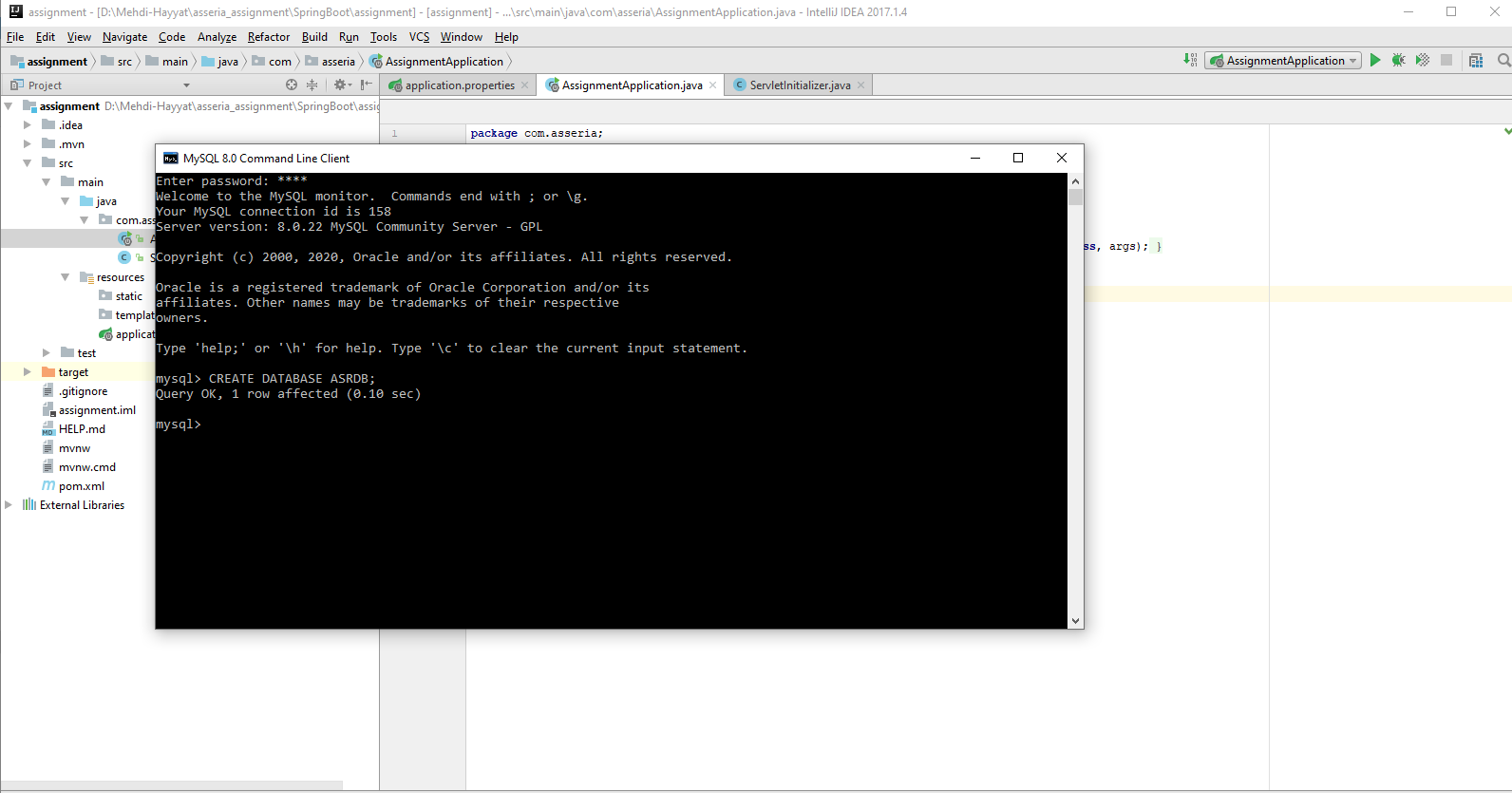
On this screen I added my required dependencies and project metadata like group, name and package option as war as shown in image above

After setting all of these things just click on generate button a ready-made application with default configurations will be downloaded.

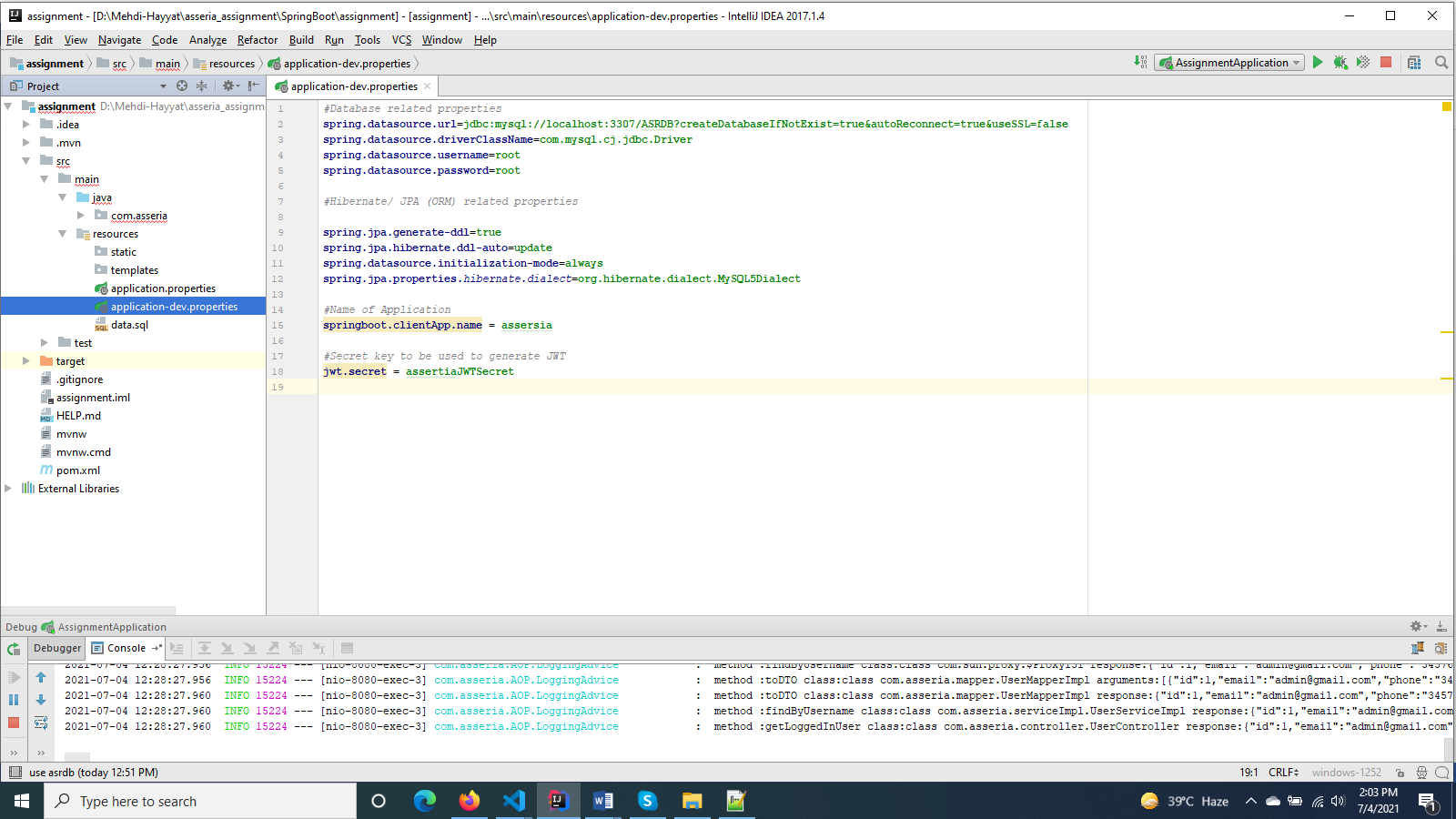
After that import application in intellij idea at starting it will download all dependencies given in pom.xml file as shown below.



After importing application and downloading all dependencies the second thing to do it to configure database connectivity so first of all I created database by using Mysql server 8.1 as shown below.



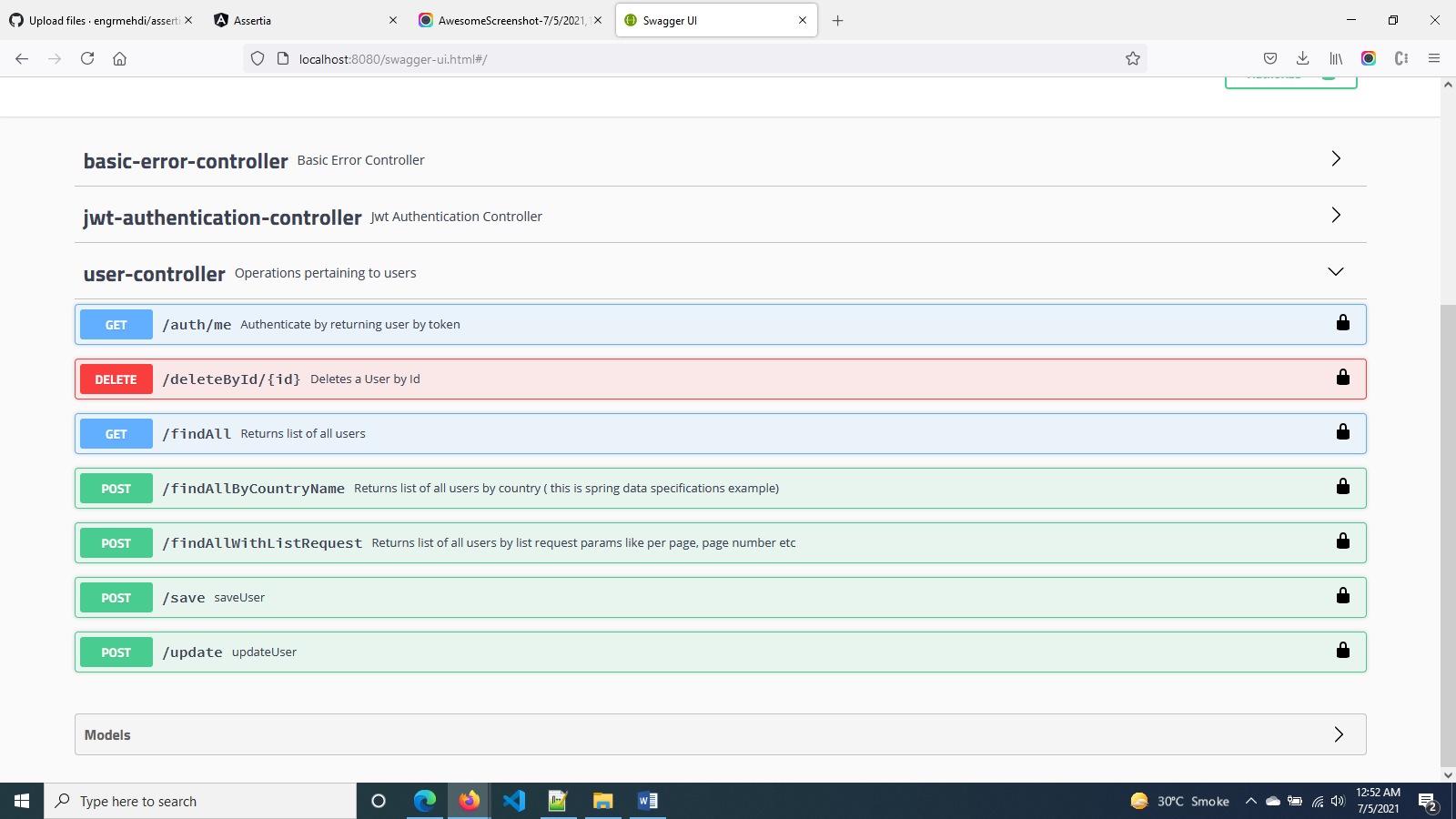
After creating database successfully there need to add some configurations in application.properties file to make connection as shown below.



Now the next things is to add some configurations related to security and swagger before that I created my all packages like controllers, services, repositories and entity classes etc.

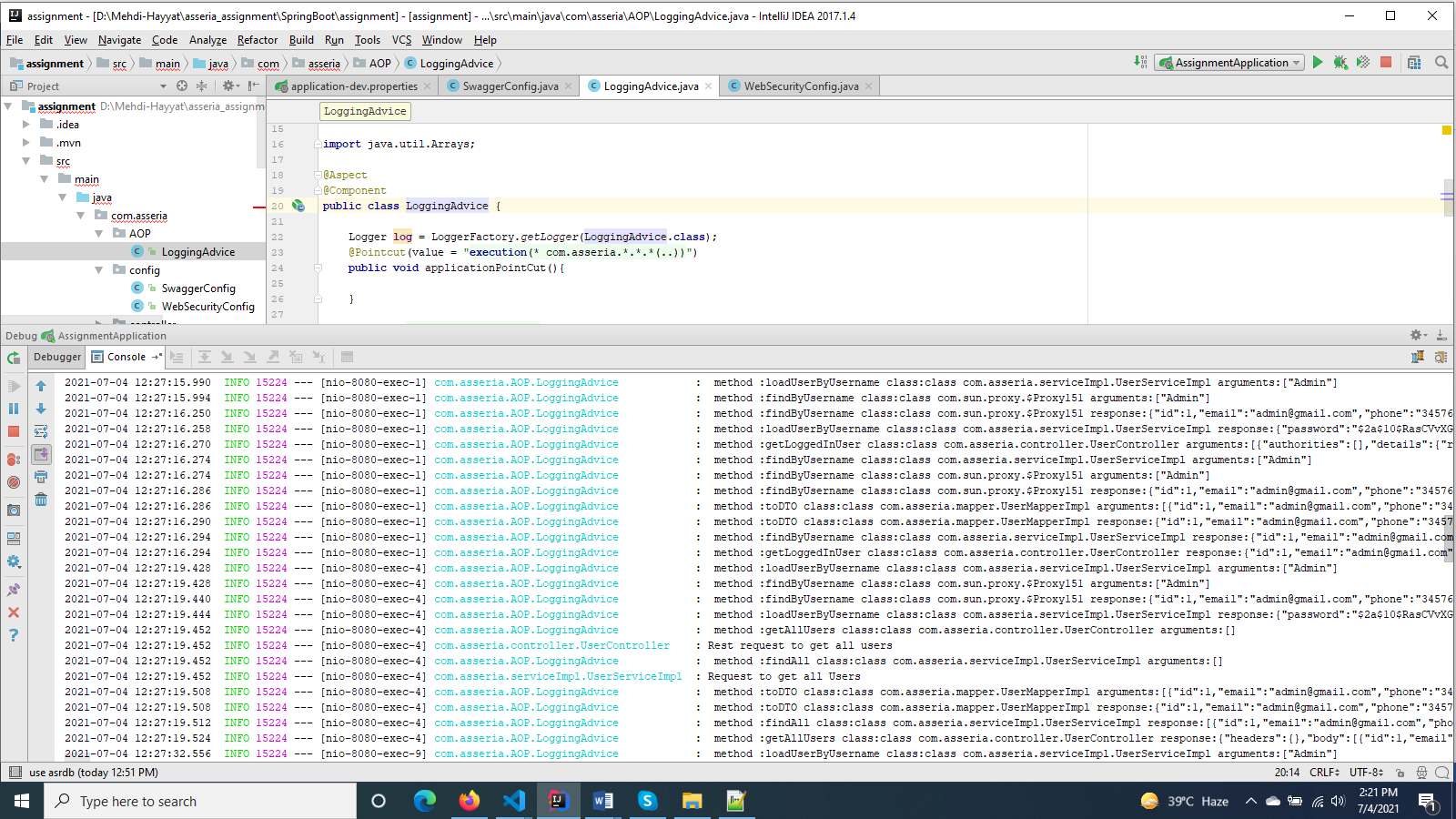
So starting from adding WebSecurityConfig which extends WebSecurityConfigurerAdapter and override it configure method to apply our own security.

Then I added configuration for swagger documentation in swaggerConfig class and after adding we can access all api as shown below.



Now the last configuration is about logging. As we need a central logging service for whole application we are using AOP ( aspect oriented programing ). For this we added dependencies for this service and then add its configuration in LoggingAdvise.java class.it can log in scope of com.assertia for each server call with the following information

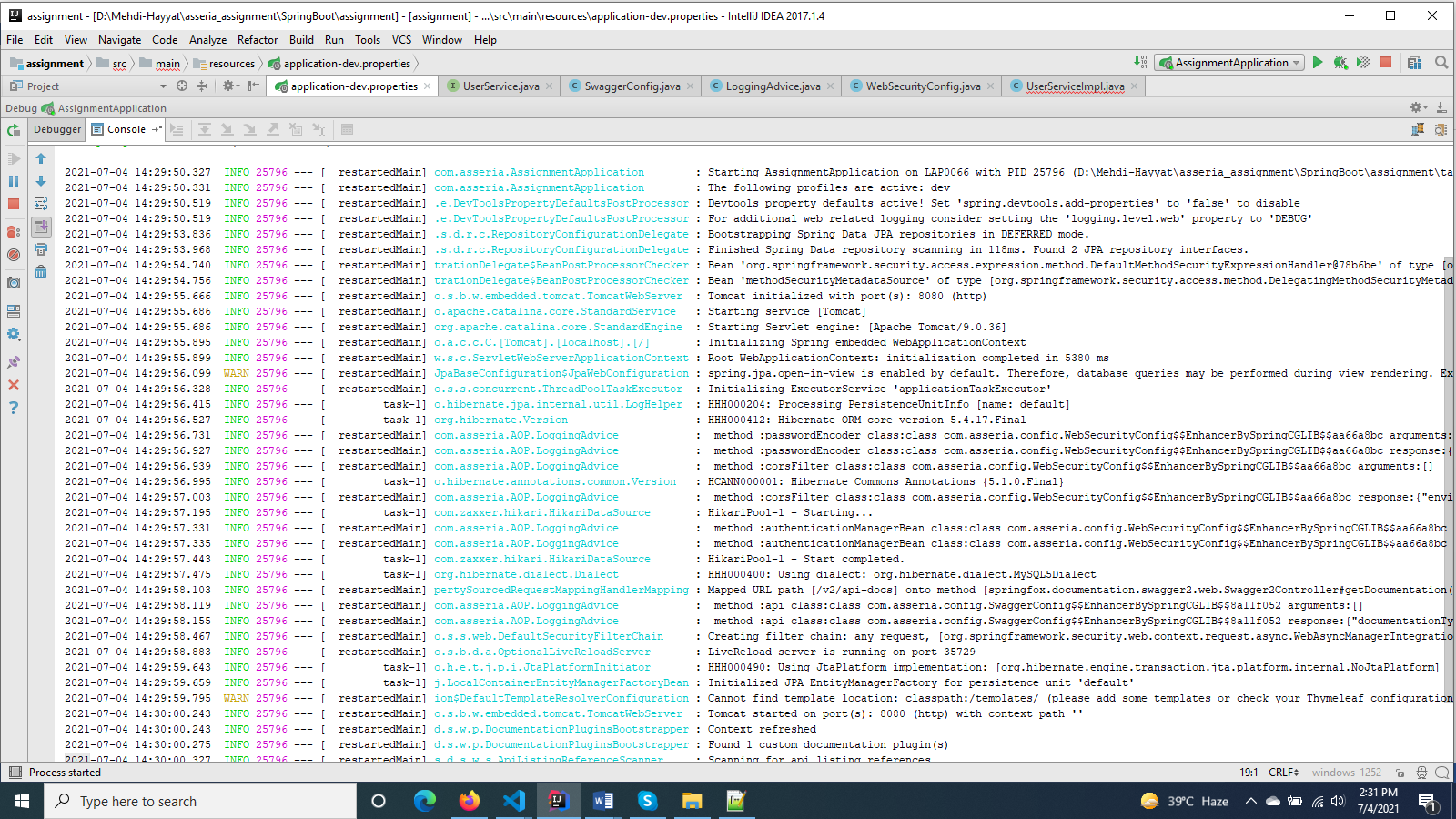
Method name, Class name and Arguments.

We can add or remove these as per our choice. On console window this will show like this. 

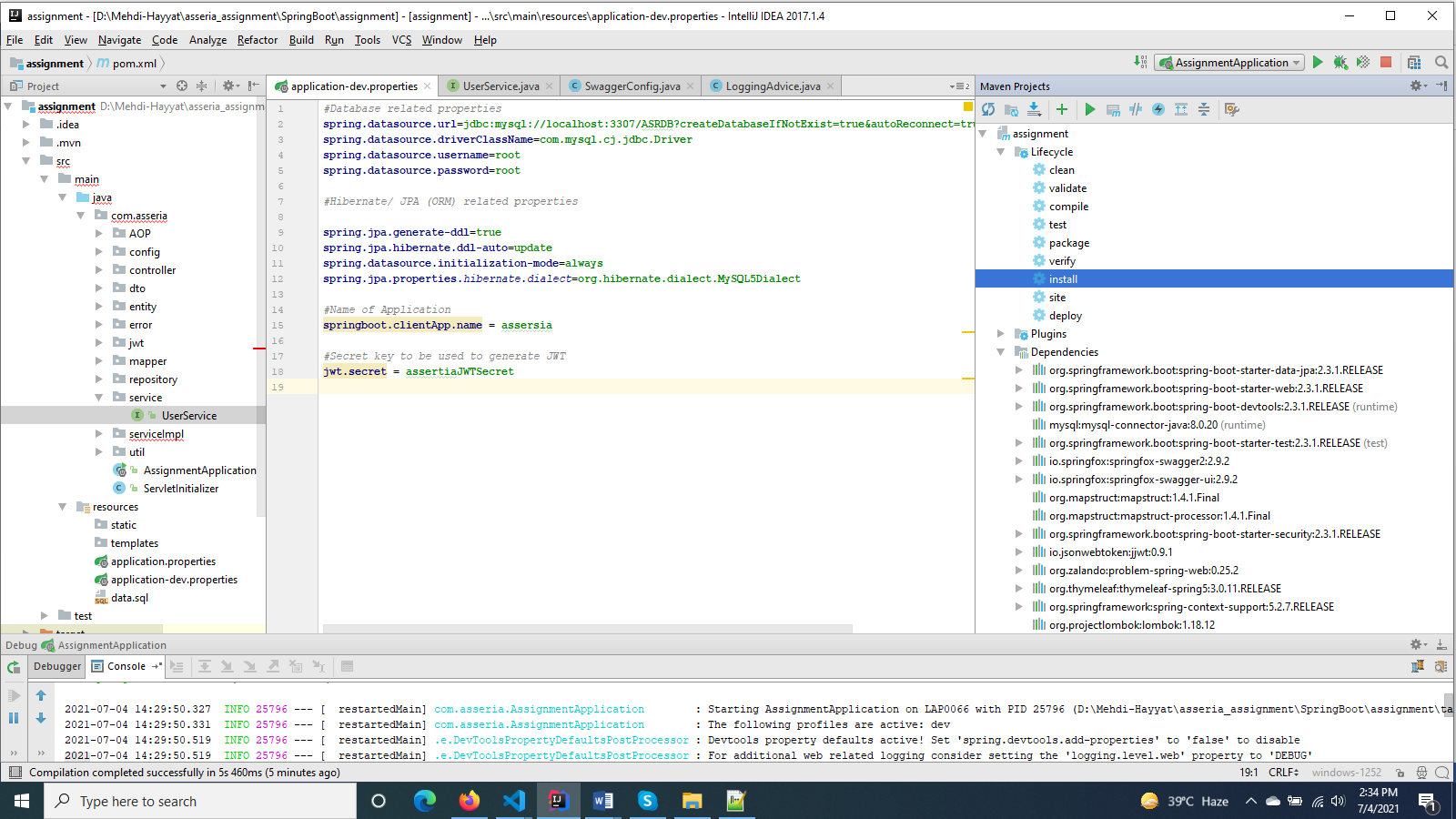
Now the basic configuratons are added and next thing is to develop application core functionalities so I started from creating entity classes then its repositories and then created servies and at the last I created rest api which are rest controllers.

**Running And Deploying Application:**

When all done just click on run or debug option on top right side in intellij idea application will start from reading properties file and then checking entities and repositories and then running default sql and then start default internal tomcat and deploy the application there and will start it as shown below.



For external deployment there is an option to make war file from maven so just go to maven project tab and go into lifecycle first clean and then install it as shown below.

war file will be created in target folder then this war file can be deployed on external tomcat server.

**Angular Application:**

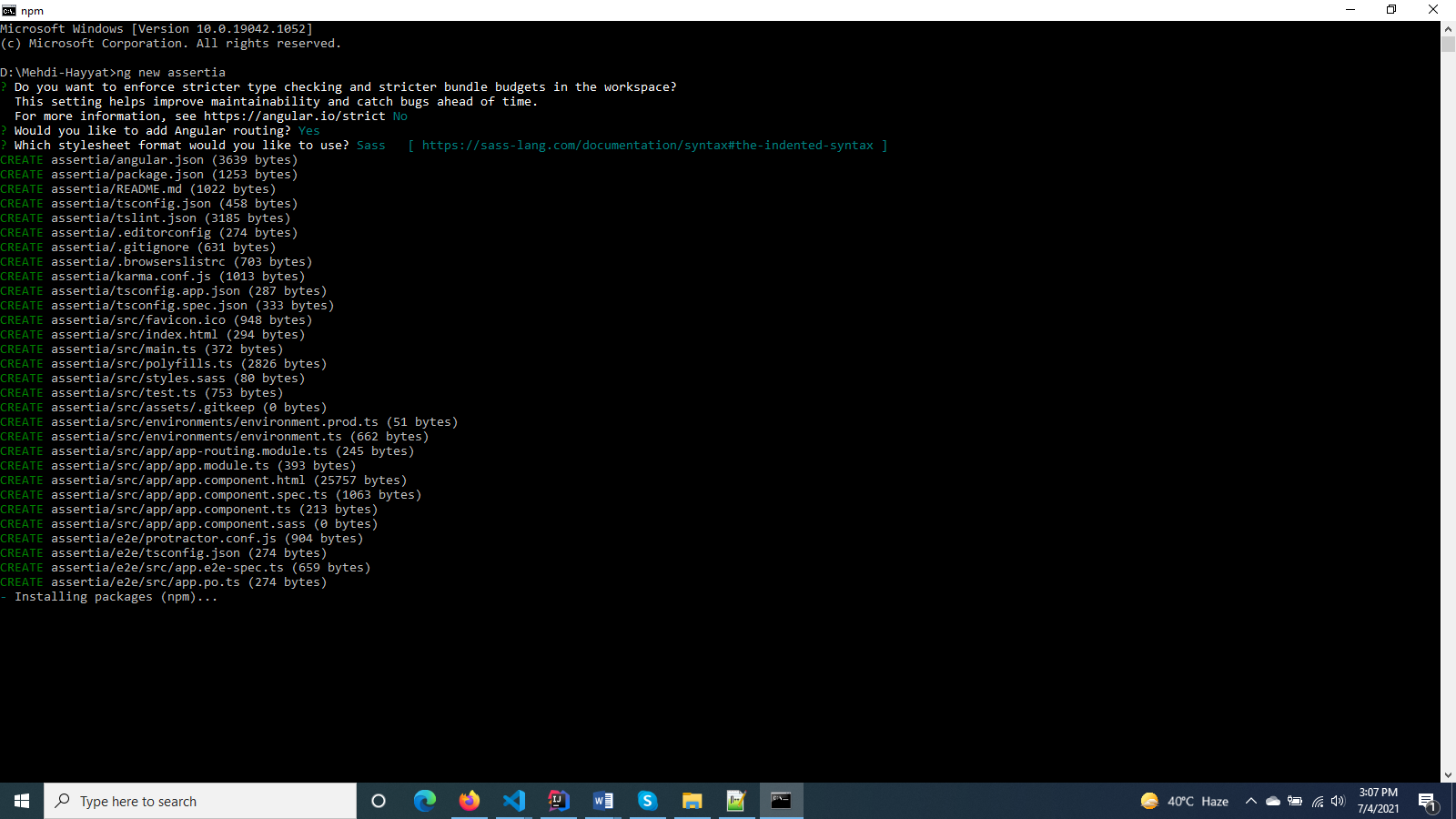
**Features:**

* Login with jwt token authentication
* Interceptor for centrally handling all rest apis
* Application loader on starting application
* Three level left side menu
* Reactive form for submitting and showing json response
* fxFlexLayout for html pages design
* auth guard for authentication of routing
* i18n for Multi language implementation ( currently on login page only)
* Custom pipe and filter implementation
* Generic component for all alerts, errors and warnings messages
* Confirmation alert component centrally developed

**Development:**

Angular application is very easy to develop it can be created just with one command before that there should be install angular and node.js on system after setting angular environment. Just go to path and open cmd and run this command.

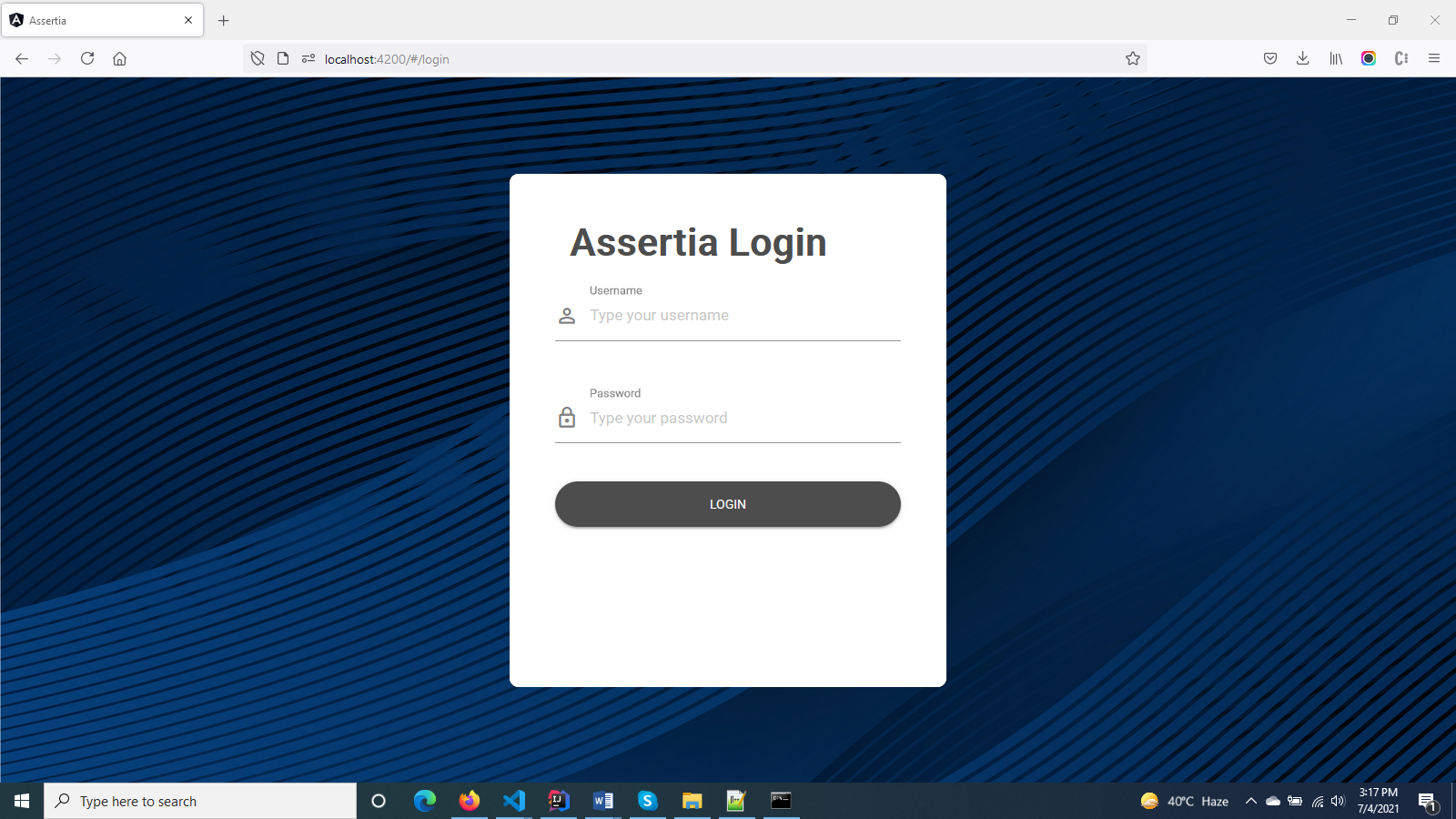
ng new assertia.

It will create angular application with all default structure having one app module default and with all node module dependencies as shown below. 

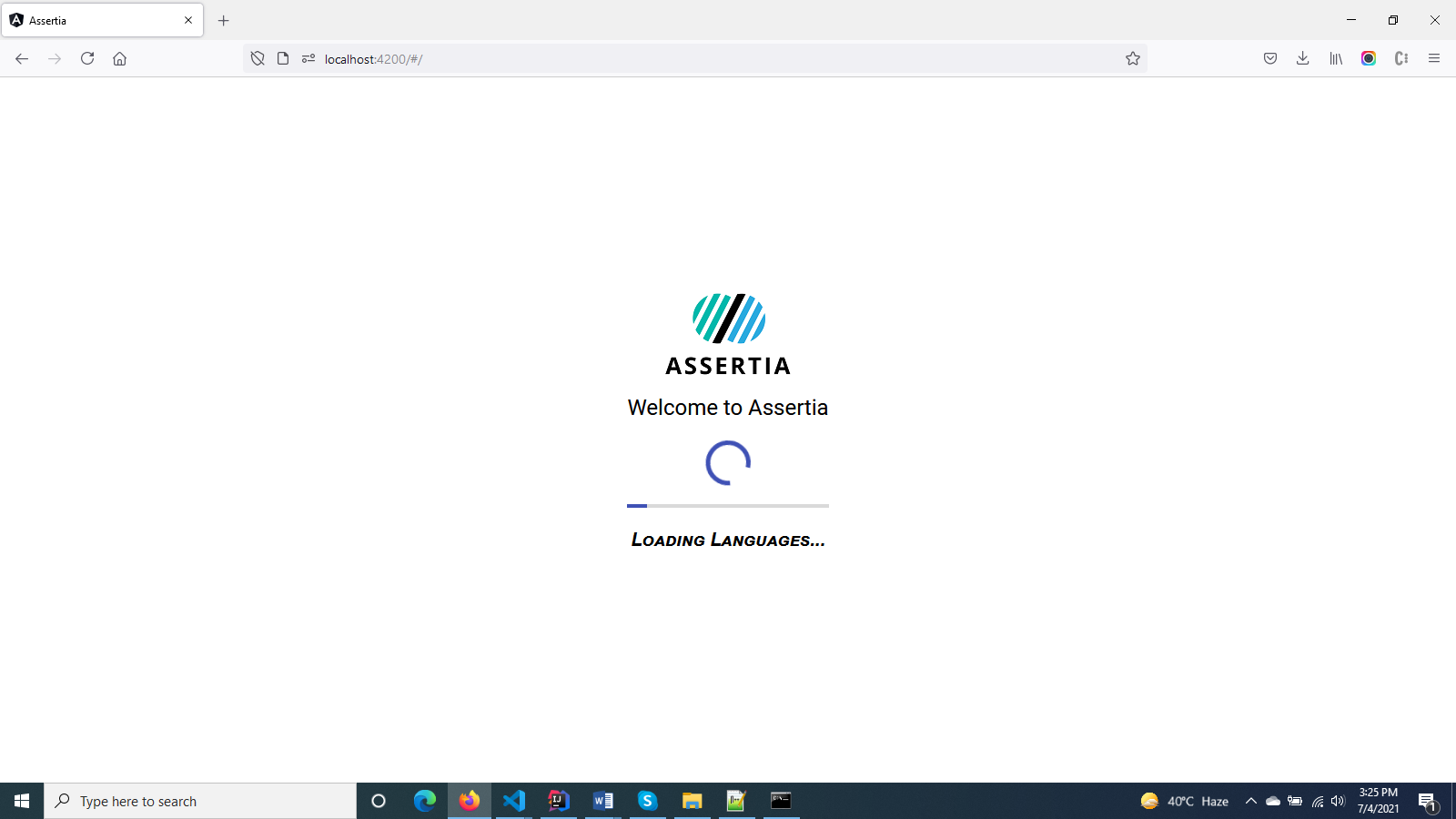
So the next thing is to develop application so open application with visual studio code and at the start we must apply auth guard to secure our routes same as we done on spring boot to secure apis with security configurations. So here auth guard will work as filter on spring boot. Here auth gard will check token validation and if it has a valid token it will route the component other wise it will redirect to login page. So after setting auth guard we will create login page. For creating new component just run this command

‘ng g m login’ for module and ‘ng g c login’ for component

After creating component I just created login service class and design page with reactive forms and fxFlexLayout for styling and boxing.

So our login page is look like this. 

Now this will take username and password and will call server side authentication from login service class. Now what should be happened when successfully login.

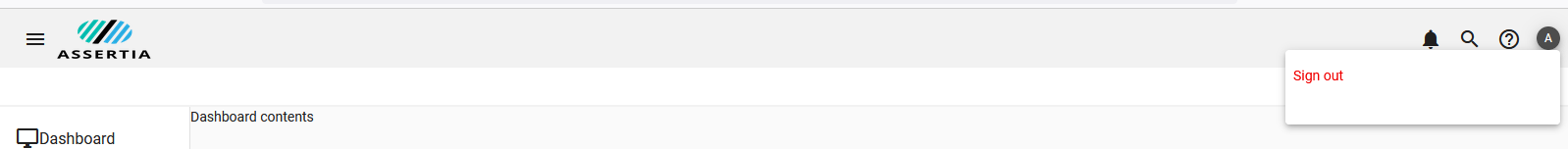
Here we have a module to load on successfully login which named as mainLayout and we have a service to store user info like token in out borowser local storage so that when next time I will open application I don’t need to login again application will have my token and will validate me from server. Actually angular application has a hierarichal strutures it contain one top module that is app.module and then child modules and eatch module has multuple components so the login component is the part of main module and after login it load mainLayout module which contain all core components of application. In the middile of these we have a appliction loader so when login applicatoin right after that loader gets called and it load all initial server calls like user info , application parameters and others calls which can be required in future.look like this. 

So here we have developed login page and after login we have application loader now the next thing is to develop out mainLayout page where out all secured components will be loaded.

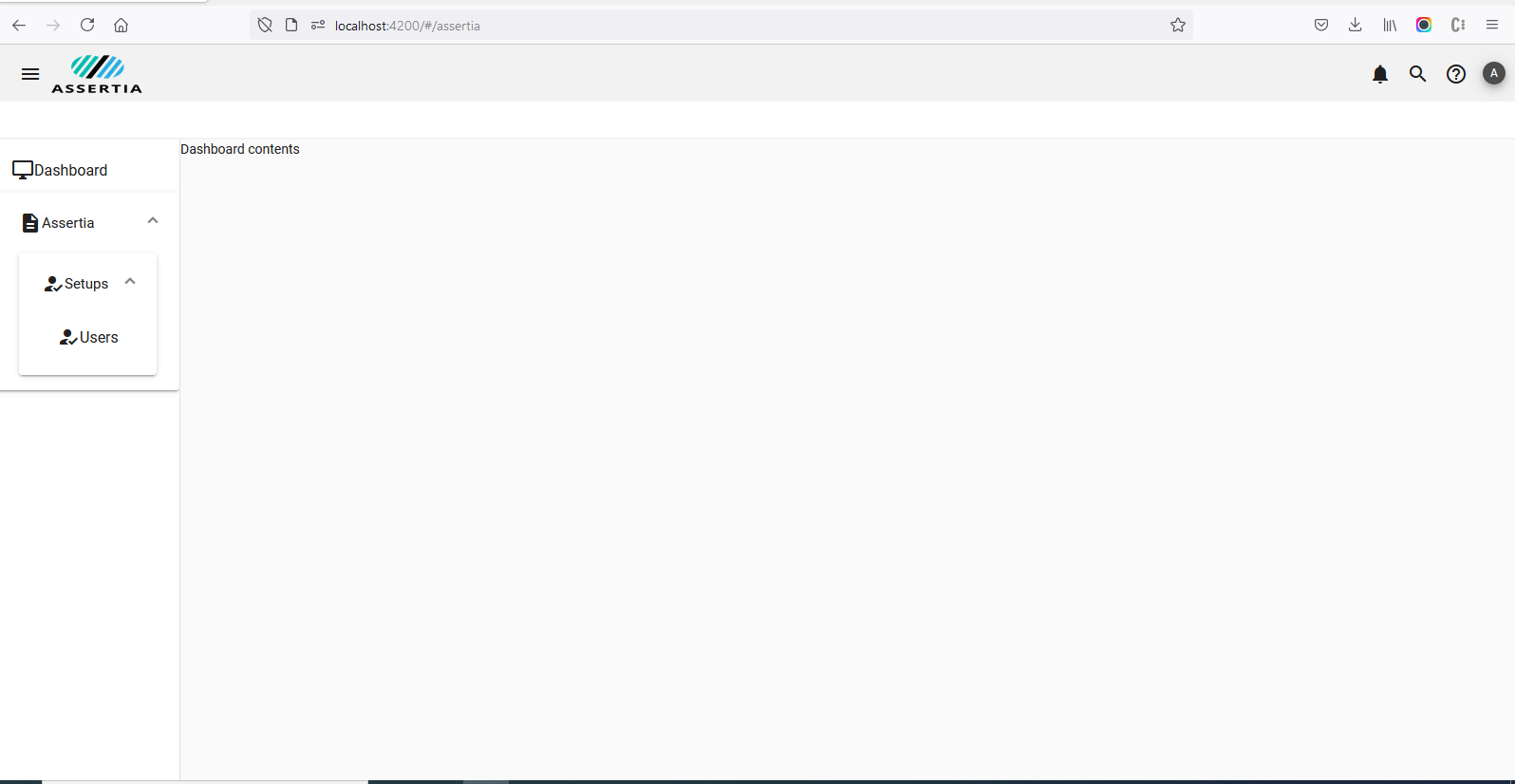
Main layout have the following child components

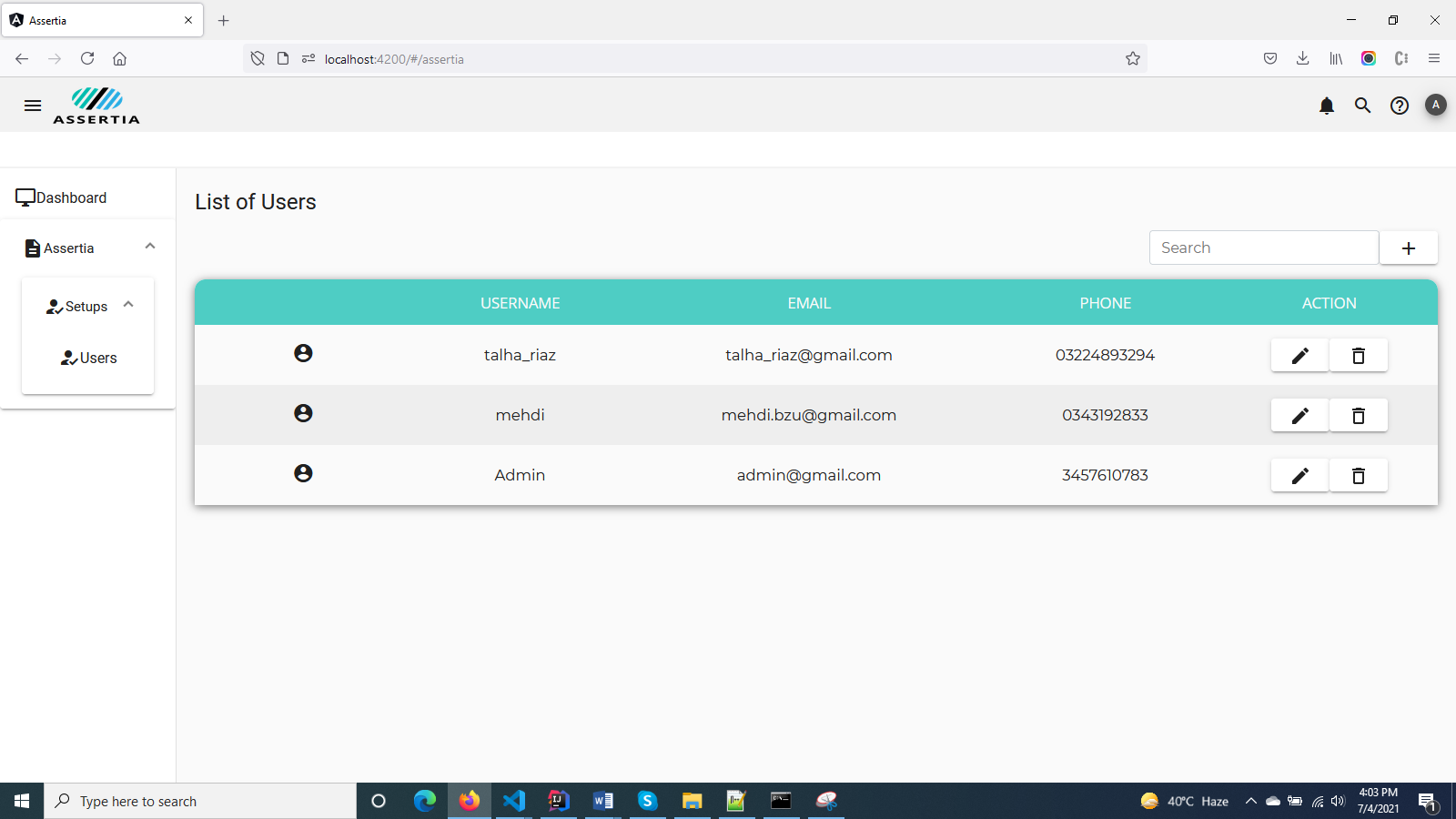
* Header
* Side menu
* Content area

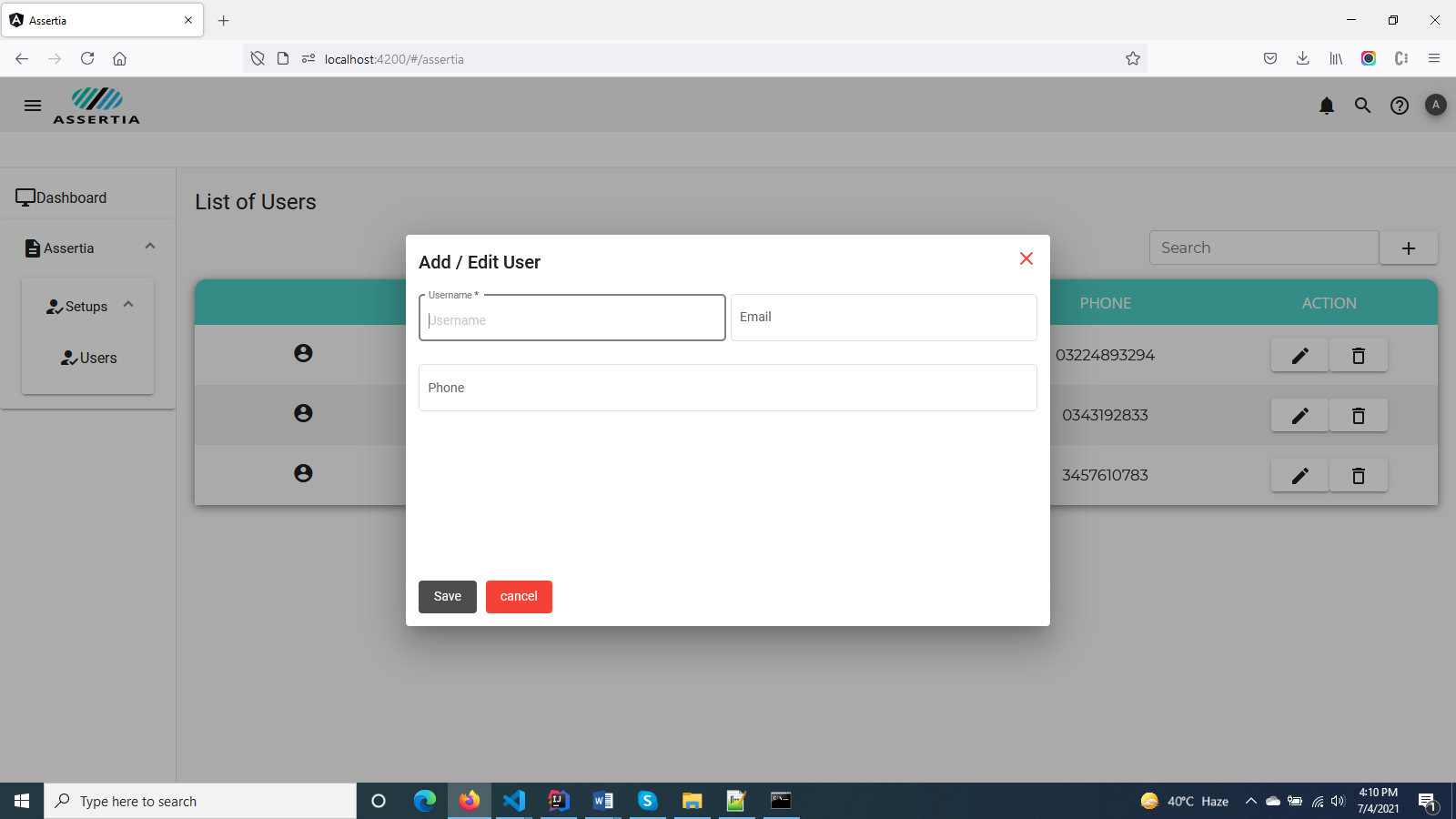
Starting from header I developed all separate components for header side bar breadcrumb area. I used mat-icon for icons and mat-sidenav for side navigation menu and mat-sidenav-content for loading content using router-outlet. So all the child components and module will be loaden inside this router-oulet. I added a routing file in mainLayout module as each module has its own routing class to route its child components. At header we have the following options first a menu toggling the side menu option second company icon and then on right side we have just icons for notifications search and user menu inside that there is a logout option as shown below.



And the side navigation with content area is looked like this.



After that I developed crud operations for user. Here one thing which is need to be noted in assignment there was mentioned for separate entity classes for country, state and streats which is implemented on spring boot side but I am not showing them on client side as I have not created separate crud operations for them as this is not asked in assignment so I just added entity classes and on adding user I am just adding that information is backen side in my code. So here is the listing page of users list with all options to seach , delete, add and edit. 

Add edit user component is just a modal window why I am using modal instead of a routed component. Because if I use a routed component I need to manage a separate service to maintain state secodn there will be an extra data handling between components navigation so the best approach is to deal with add edit form is to use modal windows . I have applied just a simple form. We can make it more customize by implementing maximum and minimum screen option and also by creating a separate crud templete and inside that we can place dynamically our components. It is just looking like this. 

**Running And Deploying Application:**

To start angular application on local browser we just run this command

npm start or ng serve

it will be running on browser on 4200 port to access on browser just type on borwser <http://localhost:4200>  
For making build to deploy on server we need to make build so we can make build by using this command inside project directory

ng build --prod

this will create a build version inside dist folder which exists in prject direcotry. Now what is the step to deploy this build. To deploy this on tomcat server we will copy this build files and will place inside spring boot applicaion resources filder there we will make a webapp folder and copy angular build inside this. Then will make spring boot build by maven clean and install this will make a complete build which will be a war file and this war file can be deployed on tomcat server simply as web application deployed.