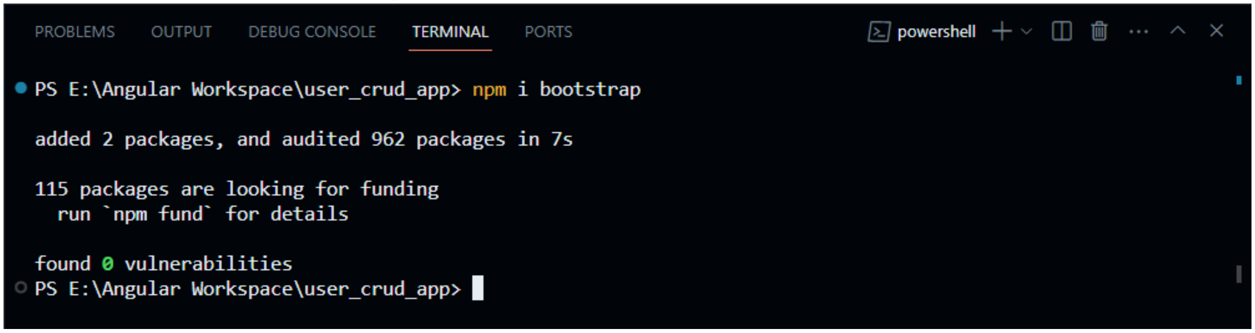
CJC
Complete Java Classes

by Kunal Sir

CRUD-IN-ANGULAR

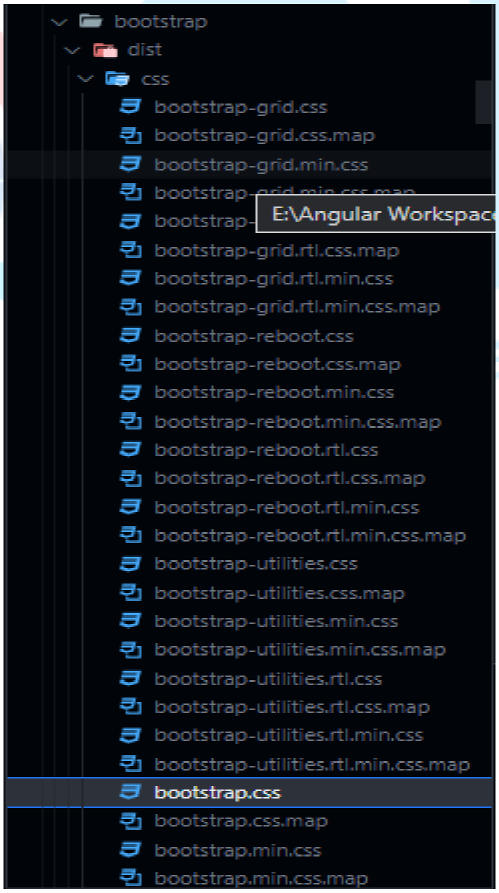
❖ Installing bootstrap in Angular project :-

1. **To install bootstrap in angular app use command :- npm i bootstrap**



```
PS E:\Angular Workspace\user_crud_app> npm i bootstrap

added 2 packages, and audited 962 packages in 7s

115 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
PS E:\Angular Workspace\user_crud_app>
```

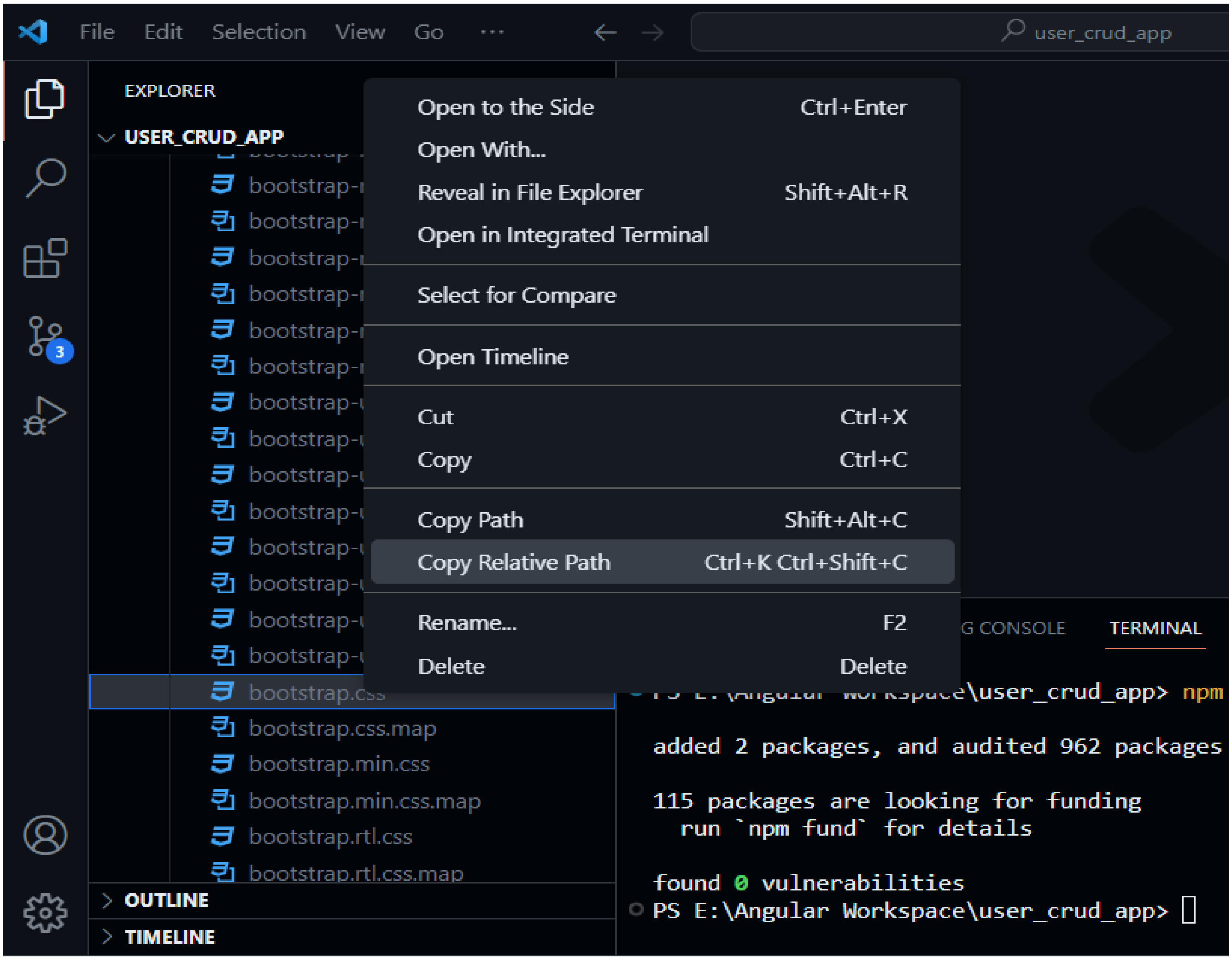
2. **Go to node_modules ---> bootstrap ---> dist ---> css ---> bootstrap --> bootstrap.css**



```
bootstrap
  dist
    css
      bootstrap-grid.css
      bootstrap-grid.css.map
      bootstrap-grid.min.css
      bootstrap-grid.min.css.map
      bootstrap- E:\Angular Workspac
      bootstrap-grid.rtl.css.map
      bootstrap-grid.rtl.min.css
      bootstrap-grid.rtl.min.css.map
      bootstrap-reboot.css
      bootstrap-reboot.css.map
      bootstrap-reboot.min.css
      bootstrap-reboot.min.css.map
      bootstrap-reboot.rtl.css
      bootstrap-reboot.rtl.css.map
      bootstrap-reboot.rtl.min.css
      bootstrap-reboot.rtl.min.css.map
      bootstrap-utilities.css
      bootstrap-utilities.css.map
      bootstrap-utilities.min.css
      bootstrap-utilities.min.css.map
      bootstrap-utilities.rtl.css
      bootstrap-utilities.rtl.css.map
      bootstrap-utilities.rtl.min.css
      bootstrap-utilities.rtl.min.css.map
      bootstrap.css
      bootstrap.css.map
      bootstrap.min.css
      bootstrap.min.css.map
```

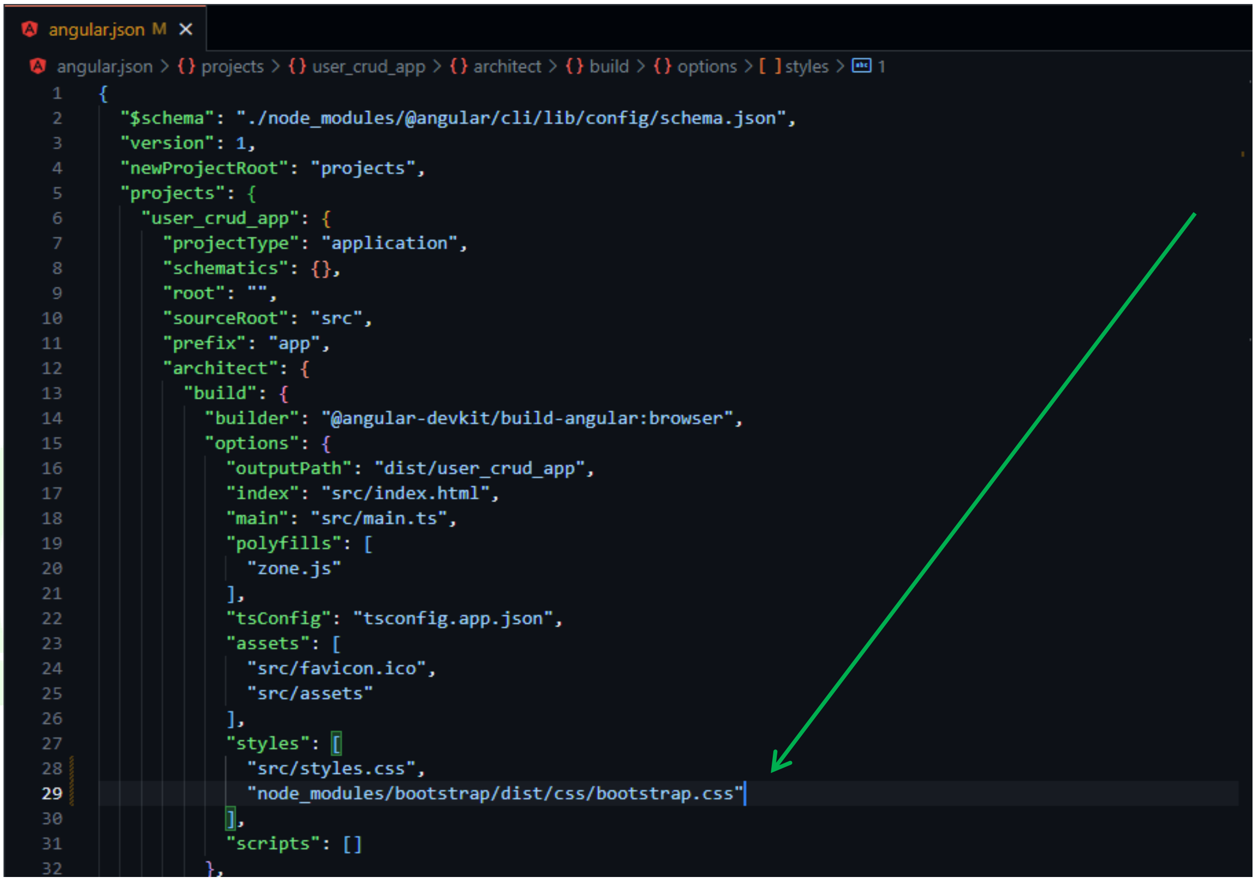
Scroll Down...!

3. Press right click button on ‘ bootstrap.css ‘ and choose copy relative path option.



```
PS E:\Angular Workspace\user_crud_app> npm

added 2 packages, and audited 962 packages

115 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
PS E:\Angular Workspace\user_crud_app>
```

4. Open angular.json file and in styles array paste the relative path of bootstrap.css file.

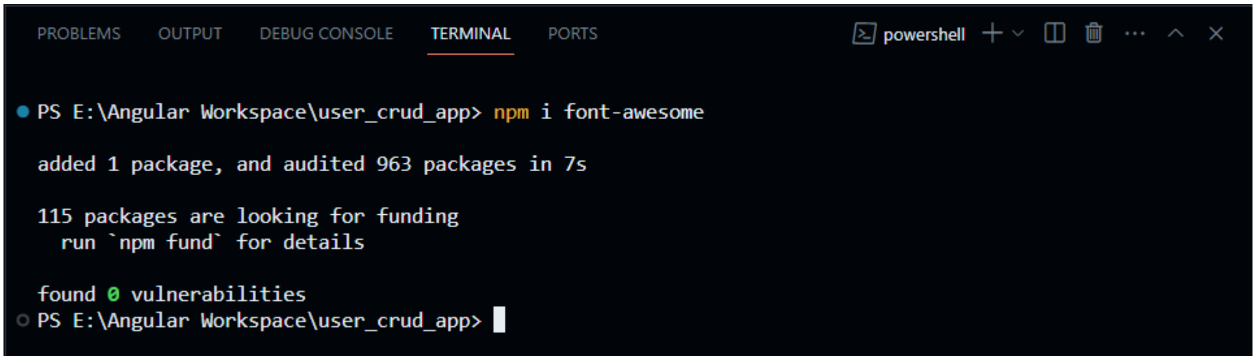

angular.json M ×
angular.json > {} projects > {} user_crud_app > {} architect > {} build > {} options > [ ] styles > 1

```
{
  "$schema": "./node_modules/@angular/cli/lib/config/schema.json",
  "version": 1,
  "newProjectRoot": "projects",
  "projects": {
    "user_crud_app": {
      "projectType": "application",
      "schematics": {},
      "root": "",
      "sourceRoot": "src",
      "prefix": "app",
      "architect": {
        "build": {
          "builder": "@angular-devkit/build-angular:browser",
          "options": {
            "outputPath": "dist/user_crud_app",
            "index": "src/index.html",
            "main": "src/main.ts",
            "polyfills": [
              "zone.js"
            ],
            "tsConfig": "tsconfig.app.json",
            "assets": [
              "src/favicon.ico",
              "src/assets"
            ],
            "styles": [
              "src/styles.css",
              "node_modules/bootstrap/dist/css/bootstrap.css"
            ],
            "scripts": []
          },
```

CRUD-IN-ANGULAR

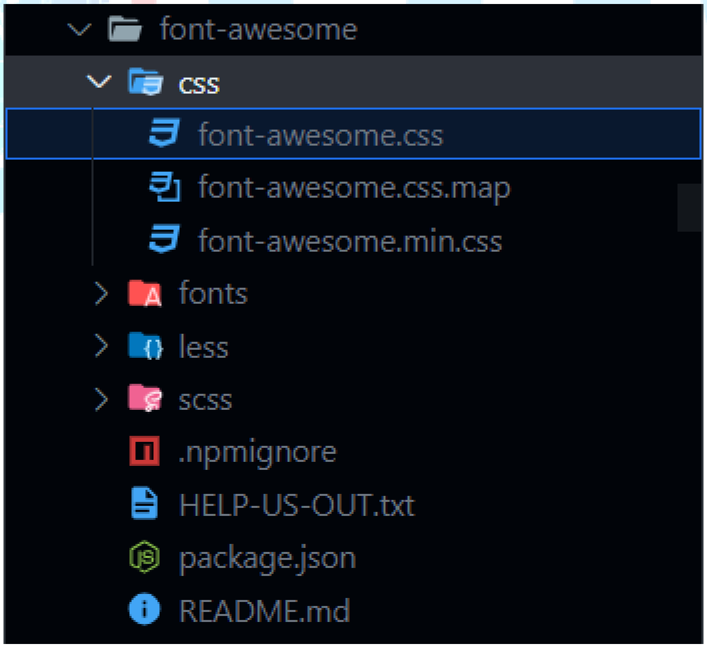
❖ Installing Font-awesome in Angular project :-

1. To install bootstrap in angular app use command :- npm i font-awesome



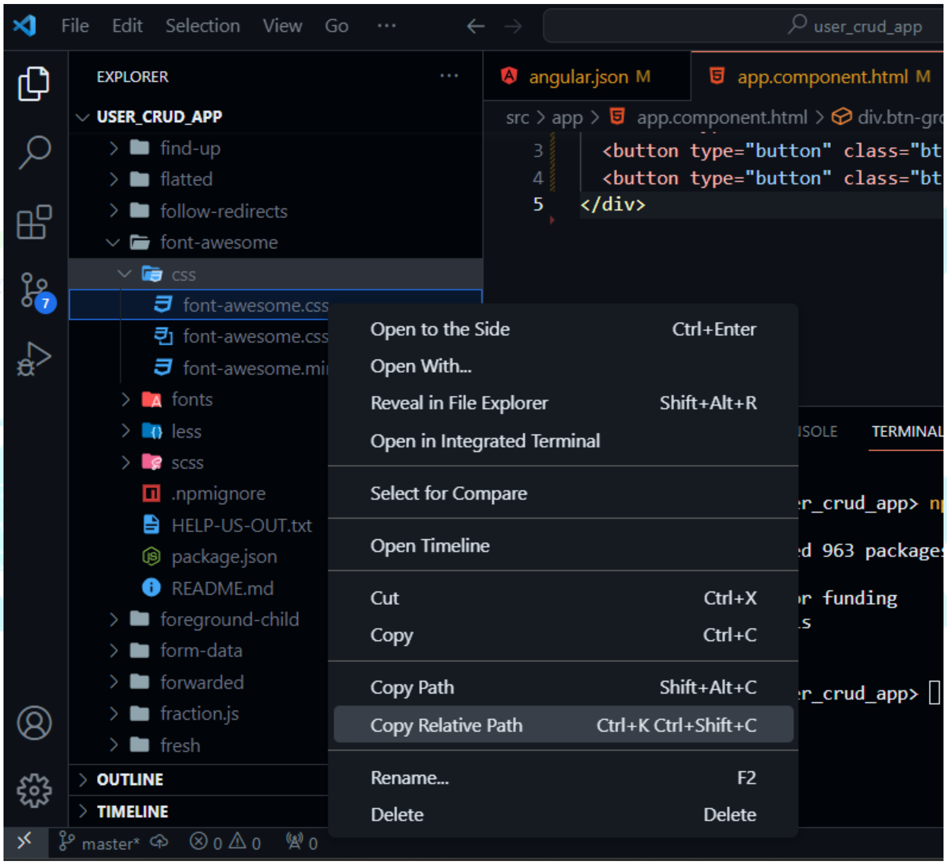
```
● PS E:\Angular Workspace\user_crud_app> npm i font-awesome

  added 1 package, and audited 963 packages in 7s

  115 packages are looking for funding
    run `npm fund` for details

  found 0 vulnerabilities
○ PS E:\Angular Workspace\user_crud_app> █
```

2. Go to node_modules/font-awesome/css/font-awesome.css file.



```
⌄ font-awesome
  ⌄ css
      font-awesome.css
      font-awesome.css.map
      font-awesome.min.css
  > fonts
  > less
  > scss
    .npmignore
    HELP-US-OUT.txt
    package.json
    README.md
```

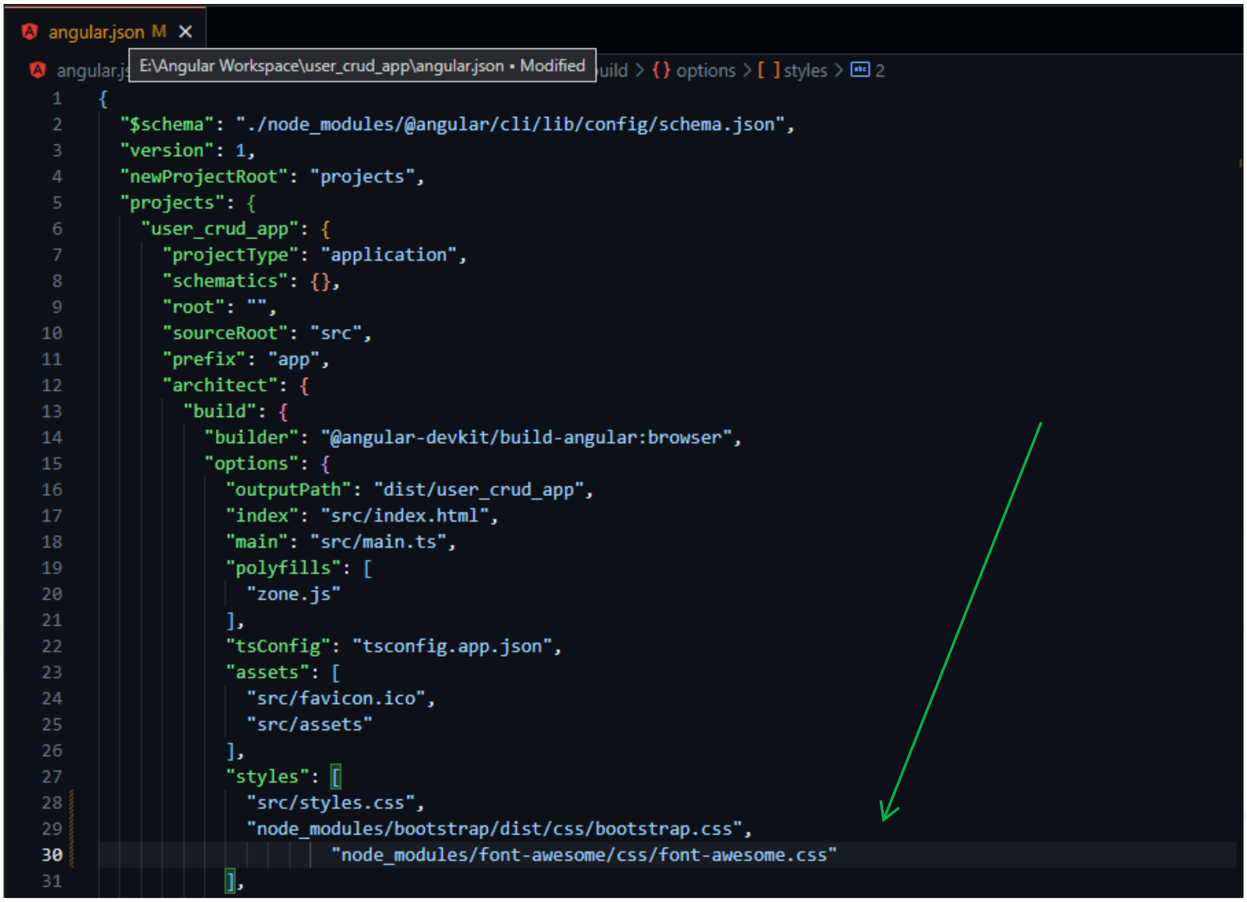
3. **Right click on font-awesome.css and choose copy relative path option.**



```
<button type="button" class="bt
<button type="button" class="bt
</div>
```

```
r_crud_app> n
d 963 package
r funding
s
r_crud_app>
```

4. **Open angular.json file and in styles array paste the relative path of font-awesome.css file.**

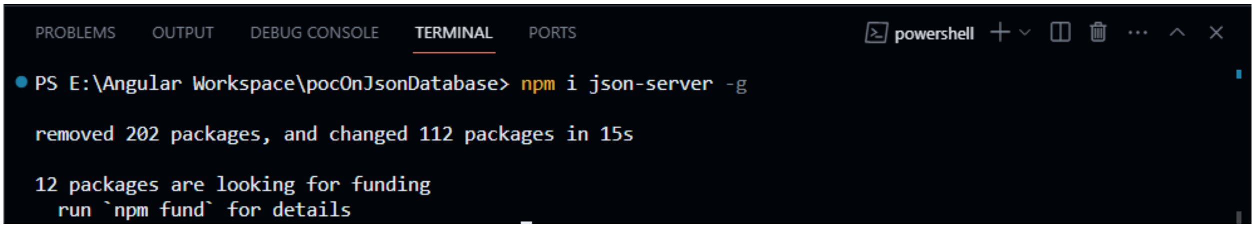

angular.json M ×
angular.js
E:\Angular Workspace\user_crud_app\angular.json • Modified
uild > {} options > [ ] styles > 2

```
{
  "$schema": "./node_modules/@angular/cli/lib/config/schema.json",
  "version": 1,
  "newProjectRoot": "projects",
  "projects": {
    "user_crud_app": {
      "projectType": "application",
      "schematics": {},
      "root": "",
      "sourceRoot": "src",
      "prefix": "app",
      "architect": {
        "build": {
          "builder": "@angular-devkit/build-angular:browser",
          "options": {
            "outputPath": "dist/user_crud_app",
            "index": "src/index.html",
            "main": "src/main.ts",
            "polyfills": [
              "zone.js"
            ],
            "tsConfig": "tsconfig.app.json",
            "assets": [
              "src/favicon.ico",
              "src/assets"
            ],
            "styles": [
              "src/styles.css",
              "node_modules/bootstrap/dist/css/bootstrap.css",
                      "node_modules/font-awesome/css/font-awesome.css"
            ],
```

CRUD-IN-ANGULAR

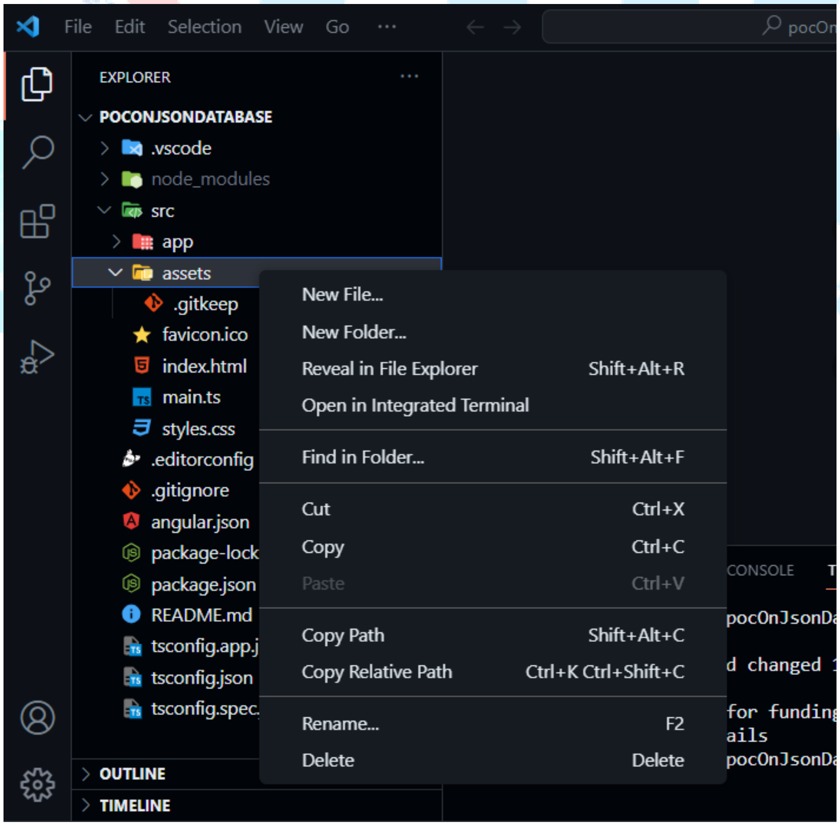
❖ Installing json-server in angular project :-

1. Install json-server in order to run and communicate to JSON databases using command : **npm i json-server -g**

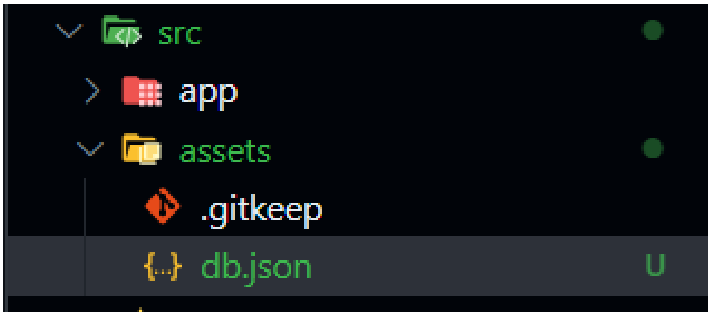


```
PS E:\Angular Workspace\pocOnJsonDatabase> npm i json-server -g

removed 202 packages, and changed 112 packages in 15s

12 packages are looking for funding
  run `npm fund` for details
```

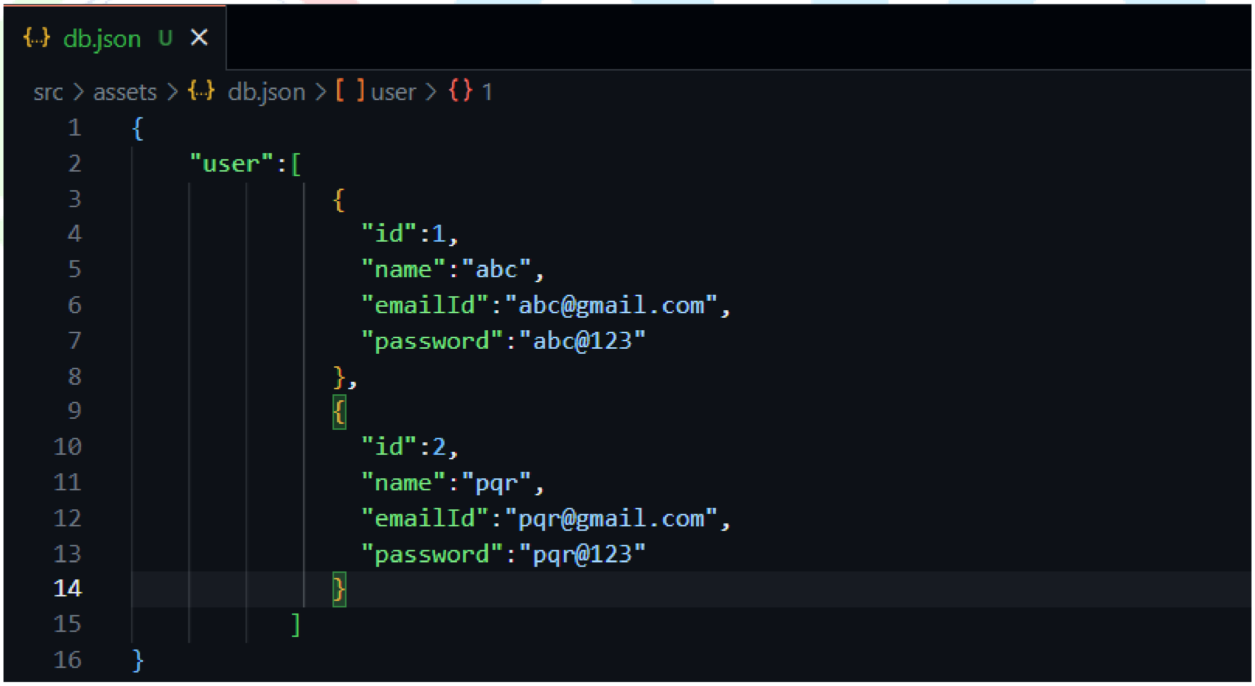
2. Right click on src/accets directory to create JOSN Document .



3. Choose ‘New File…’ option and name the file as db.json

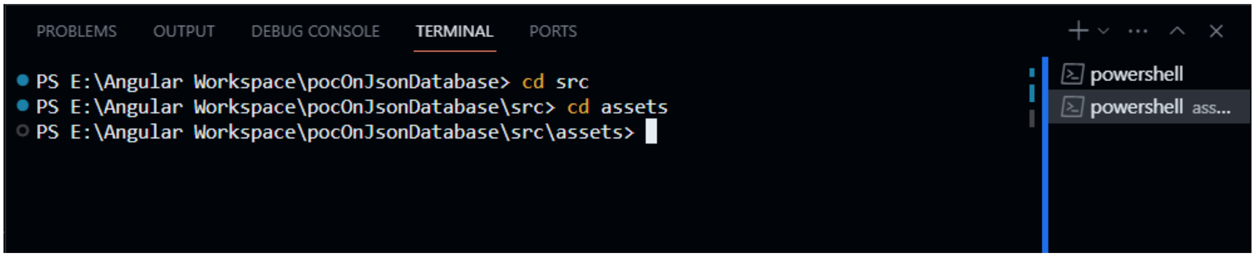


4. Add employee details in db.json in json format as given below.



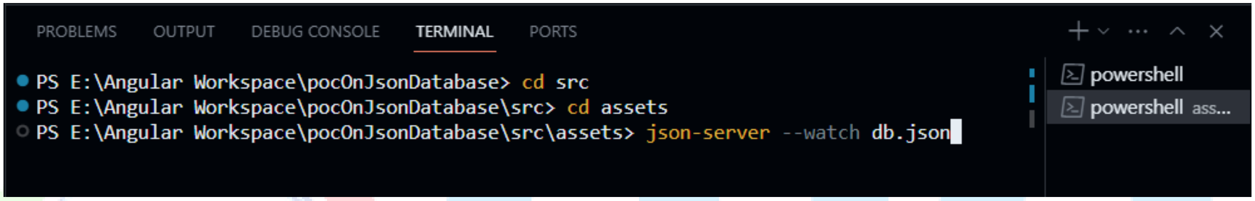
```json
{
    "user":[
            {
              "id":1,
              "name":"abc",
              "emailId":"abc@gmail.com",
              "password":"abc@123"
            },
            {
              "id":2,
              "name":"pqr",
              "emailId":"pqr@gmail.com",
              "password":"pqr@123"
            }
          ]
}
```

5. Open new terminal and change the terminal directory to src/accets as given bellow.



```
● PS E:\Angular Workspace\pocOnJsonDatabase> cd src
● PS E:\Angular Workspace\pocOnJsonDatabase\src> cd assets
○ PS E:\Angular Workspace\pocOnJsonDatabase\src\assets> 
```

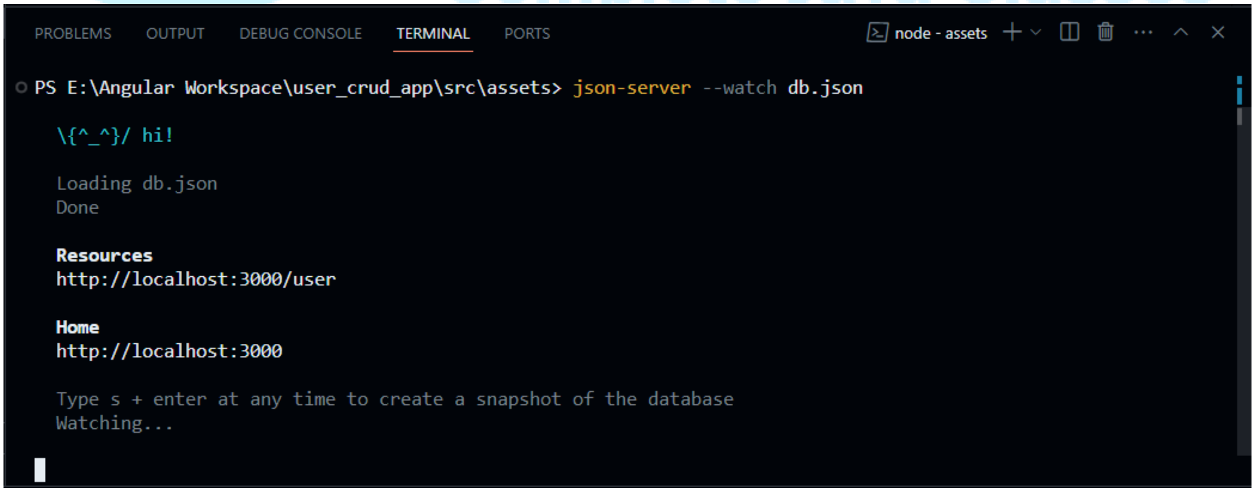
6. Now run json-server using command : **json-server --watch db.json**



```
● PS E:\Angular Workspace\pocOnJsonDatabase> cd src
● PS E:\Angular Workspace\pocOnJsonDatabase\src> cd assets
○ PS E:\Angular Workspace\pocOnJsonDatabase\src\assets> json-server --watch db.json
```

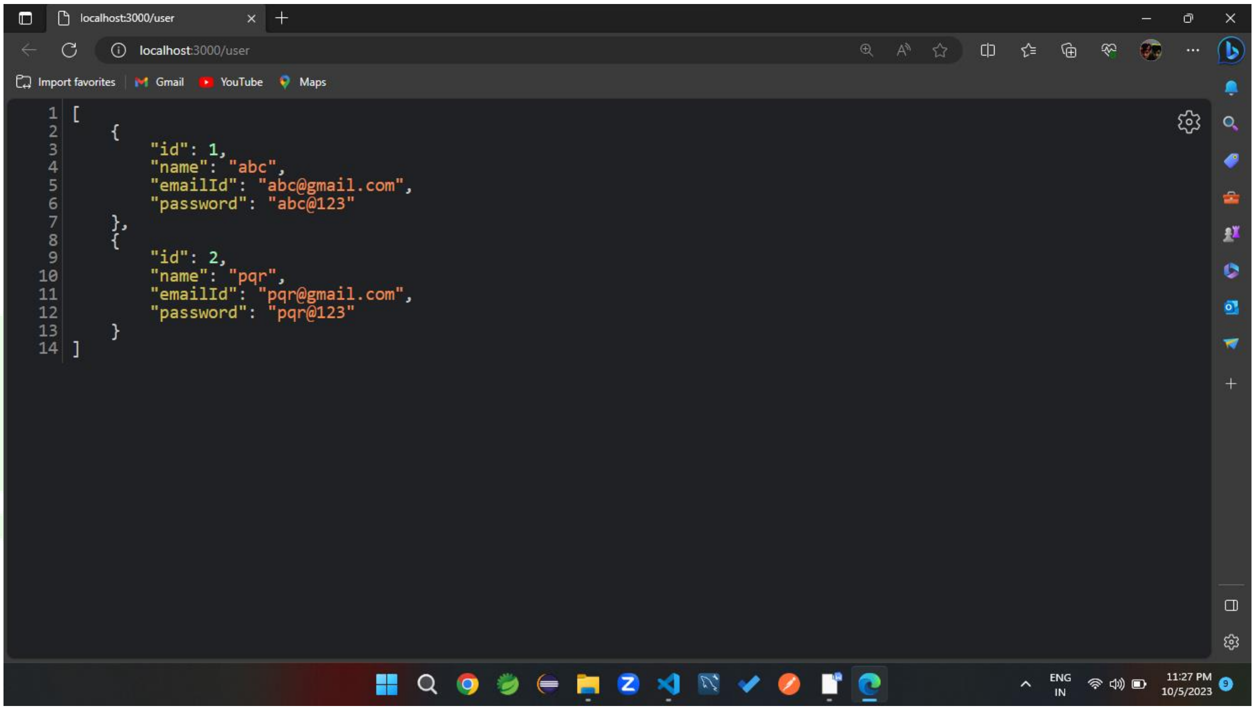
7. After pressed Enter button you can see, json server has been compiled our db.json file

And also json server have and endpoint, form which we can red the data of db.json document.



```
○ PS E:\Angular Workspace\user_crud_app\src\assets> json-server --watch db.json

    \{^_^}/ hi!

    Loading db.json
    Done

    Resources
    http://localhost:3000/user

    Home
    http://localhost:3000

    Type s + enter at any time to create a snapshot of the database
    Watching...
```

8. Open any of the browser window and use the endpoint, which is provided by json-server on terminal . in our scenario endpoint is --> http://localhost:3000/user
   (the default port number of json-server is 3000 .)

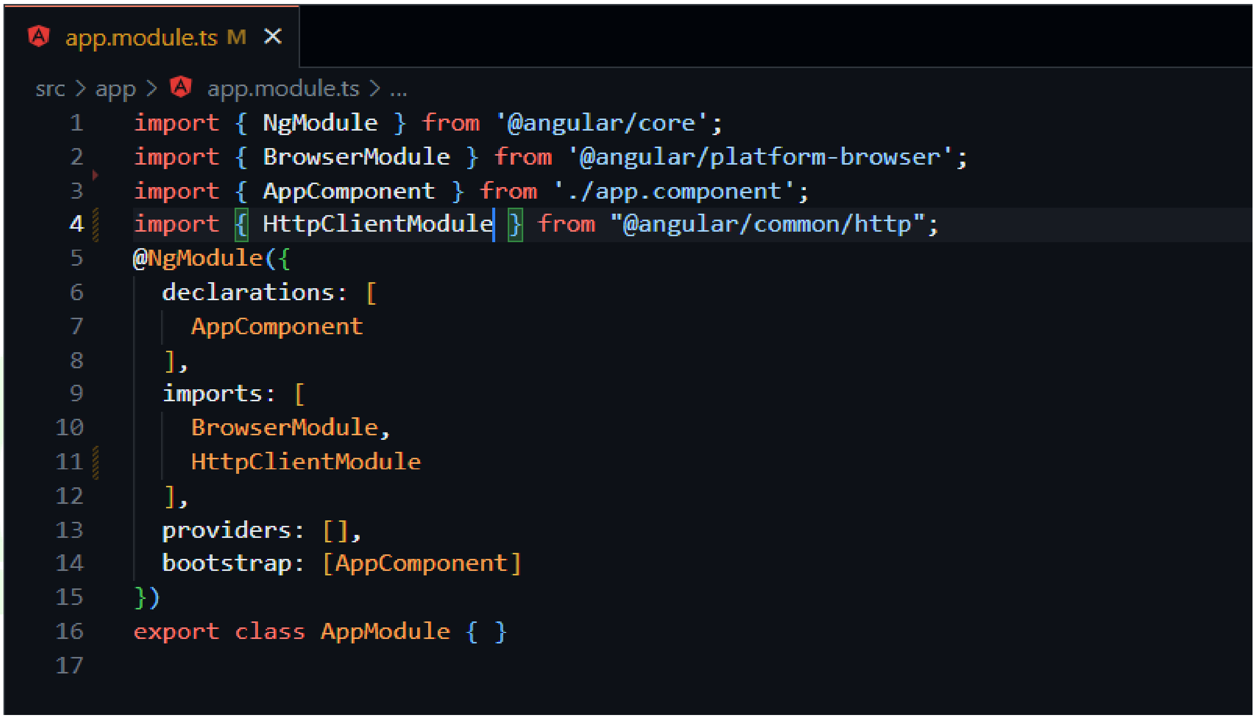


```
[
    {
        "id": 1,
        "name": "abc",
        "emailId": "abc@gmail.com",
        "password": "abc@123"
    },
    {
        "id": 2,
        "name": "pqr",
        "emailId": "pqr@gmail.com",
        "password": "pqr@123"
    }
]
```

CRUD-IN-ANGULAR

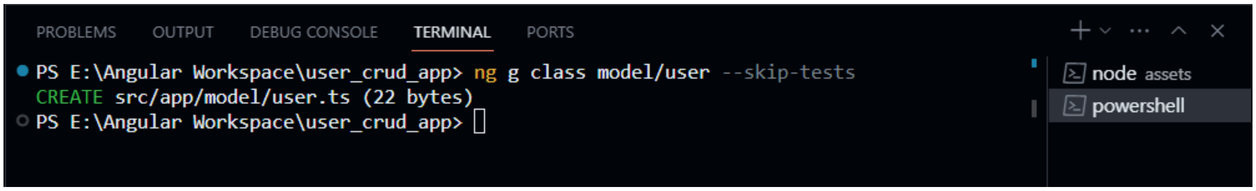
❖ **Start implementing CRUD operation on user data model :-**

1. **First import HttpClientModule in AppModule to use HttpClient methods in our service to assess endpoints generated by json-server.**

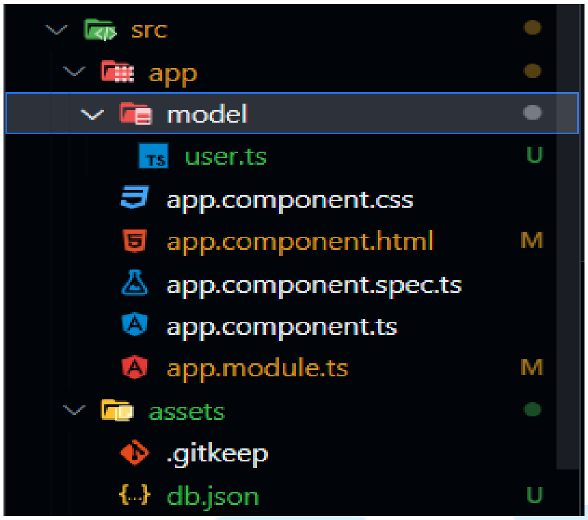


```
import { NgModule } from '@angular/core';
import { BrowserModule } from '@angular/platform-browser';
import { AppComponent } from './app.component';
import { HttpClientModule } from "@angular/common/http";
@NgModule({
  declarations: [
    AppComponent
  ],
  imports: [
    BrowserModule,
    HttpClientModule
  ],
  providers: [],
  bootstrap: [AppComponent]
})
export class AppModule { }

```

2. **Open new terminal and Create User class to store user details in model package using ng command : ng g class user --skip-tests**

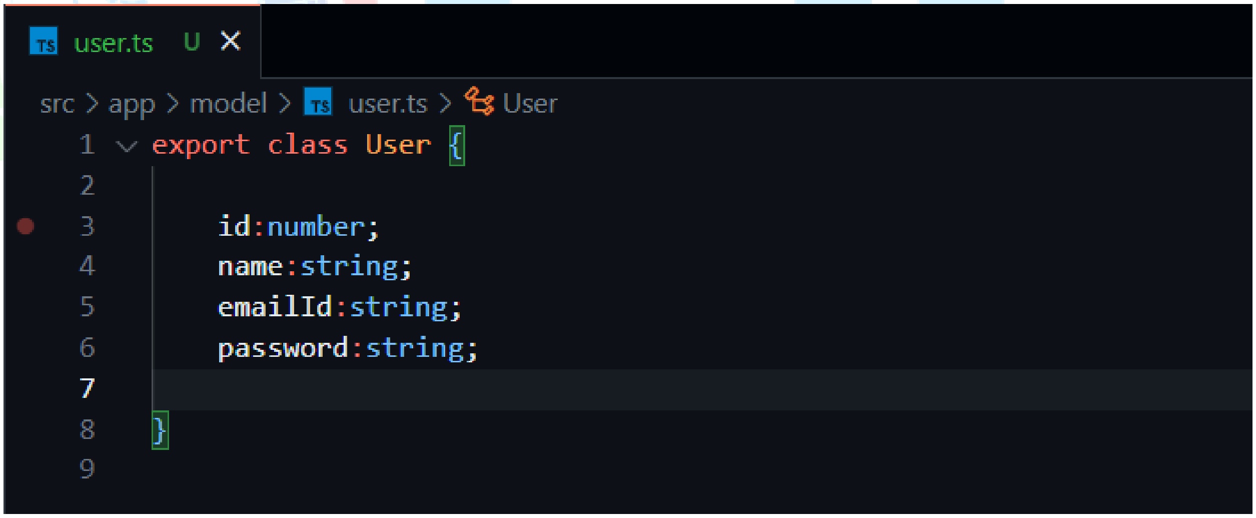


```
PS E:\Angular Workspace\user_crud_app> ng g class model/user --skip-tests
CREATE src/app/model/user.ts (22 bytes)
PS E:\Angular Workspace\user_crud_app>
```



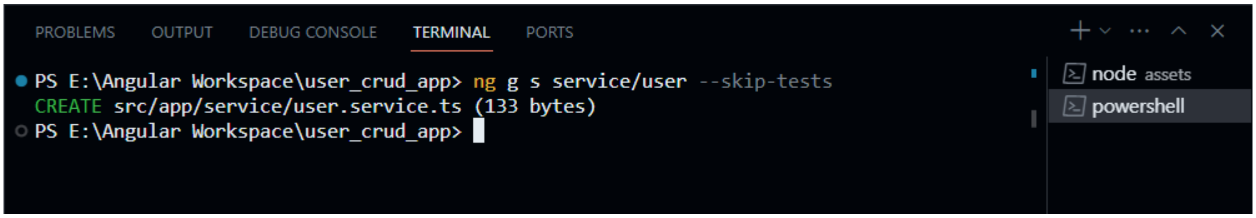
```
src
  app
    model
      user.ts                U
    app.component.css
    app.component.html       M
    app.component.spec.ts
    app.component.ts
    app.module.ts            M
  assets
    .gitkeep
    db.json                  U
```

3. **Define required properties (data-members) in User model.**

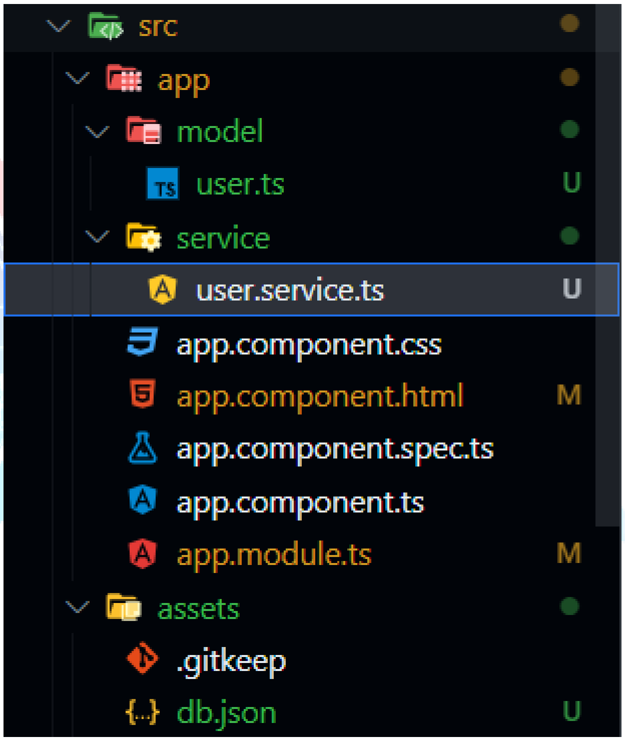


```
export class User {

    id:number;
    name:string;
    emailId:string;
    password:string;

}
```

4. Create UserService to communicate with http endpoints generated by json-server in service package using command : ng g s service/user --skip-tests

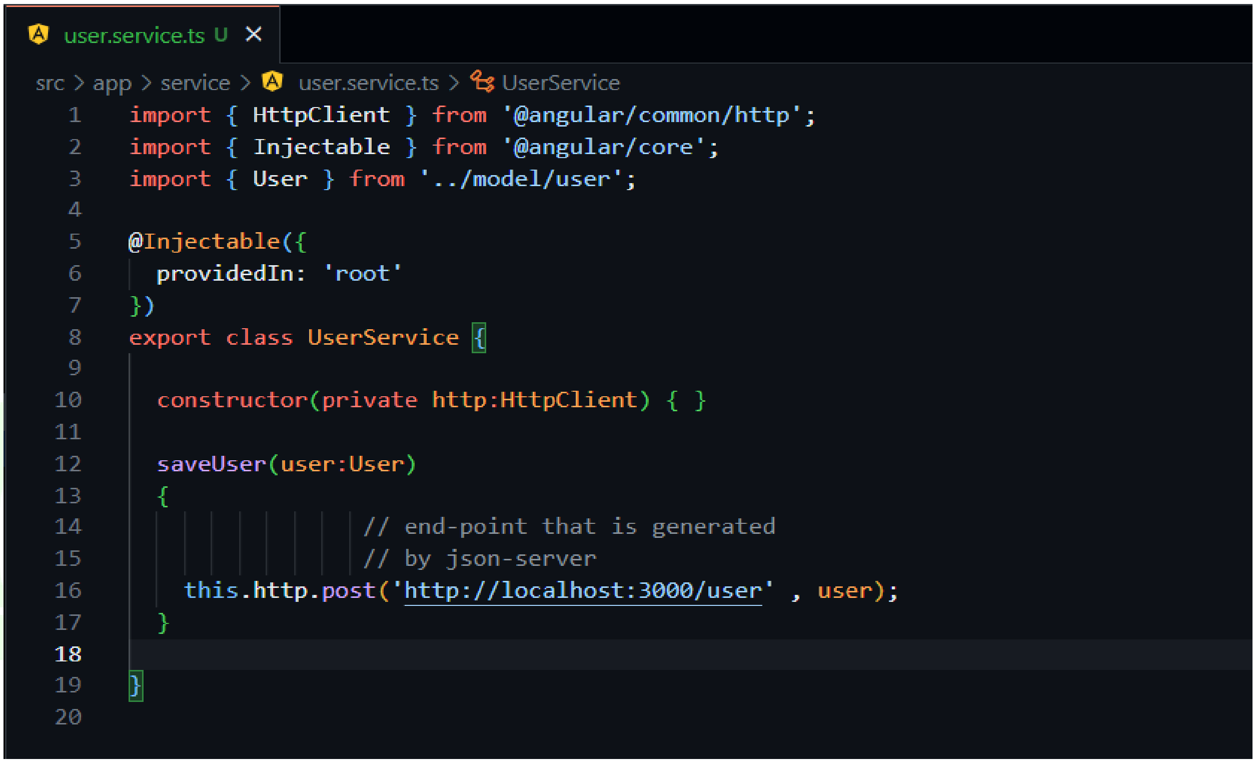


```
● PS E:\Angular Workspace\user_crud_app> ng g s service/user --skip-tests
  CREATE src/app/service/user.service.ts (133 bytes)
○ PS E:\Angular Workspace\user_crud_app> 
```



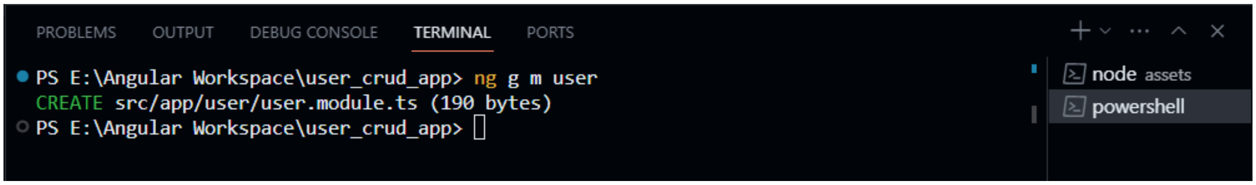
```
src
  app
    model
      user.ts                U
    service
      user.service.ts        U
    app.component.css
    app.component.html       M
    app.component.spec.ts
    app.component.ts
    app.module.ts            M
  assets
    .gitkeep
    db.json                  U
```

5. **In UserService class inject HttpClient class to consume HttpClient methods like get, post, put, delete. And also declare function to post user object, which will accept user object as parameter argument.**

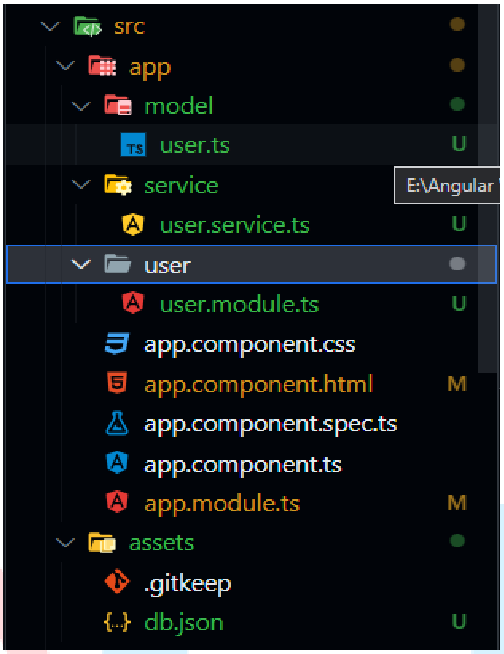


```
import { HttpClient } from '@angular/common/http';
import { Injectable } from '@angular/core';
import { User } from '../model/user';

@Injectable({
  providedIn: 'root'
})
export class UserService {

  constructor(private http:HttpClient) { }

  saveUser(user:User)
  {
               // end-point that is generated
               // by json-server
    this.http.post('http://localhost:3000/user' , user);
  }

}

```

6. **Create user module using command : ng g m user**

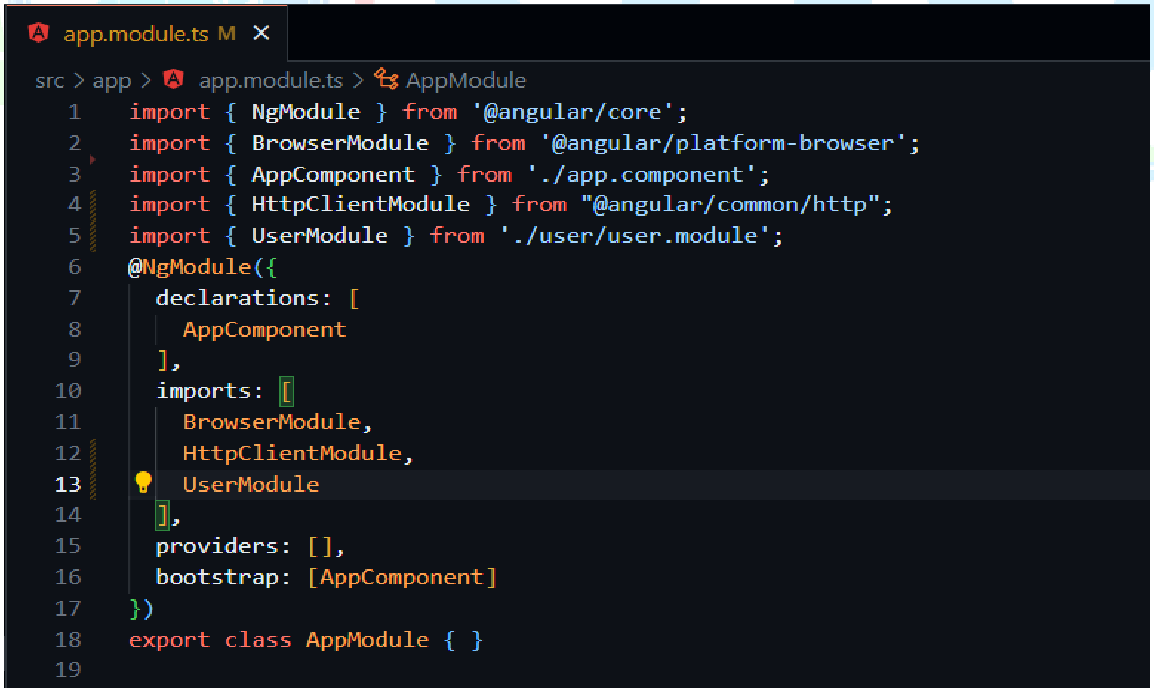


```
PS E:\Angular Workspace\user_crud_app> ng g m user
CREATE src/app/user/user.module.ts (190 bytes)
PS E:\Angular Workspace\user_crud_app>
```



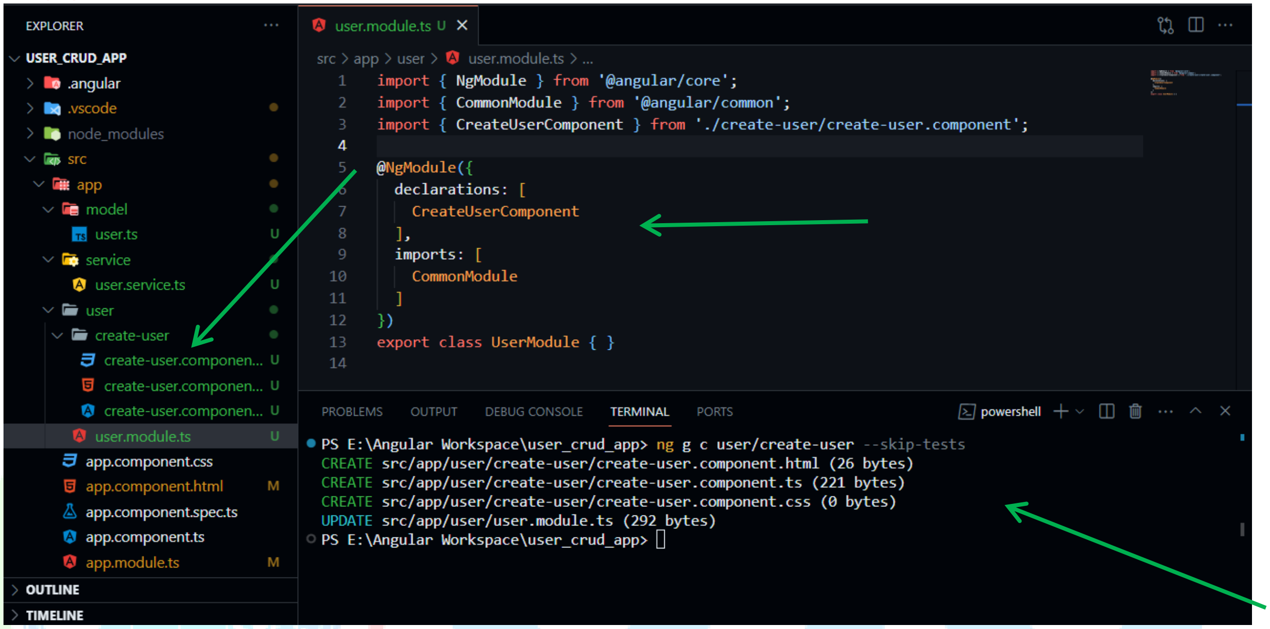
```
src
  app
    model
      user.ts                 U
    service
      user.service.ts         U
    user
      user.module.ts          U
    app.component.css
    app.component.html        M
    app.component.spec.ts
    app.component.ts
    app.module.ts             M
  assets
    .gitkeep
    db.json                   U
```

7. **Import this UserModule in AppModule.**



```
import { NgModule } from '@angular/core';
import { BrowserModule } from '@angular/platform-browser';
import { AppComponent } from './app.component';
import { HttpClientModule } from "@angular/common/http";
import { UserModule } from './user/user.module';
@NgModule({
  declarations: [
    AppComponent
  ],
  imports: [
    BrowserModule,
    HttpClientModule,
    UserModule
  ],
  providers: [],
  bootstrap: [AppComponent]
})
export class AppModule { }
```

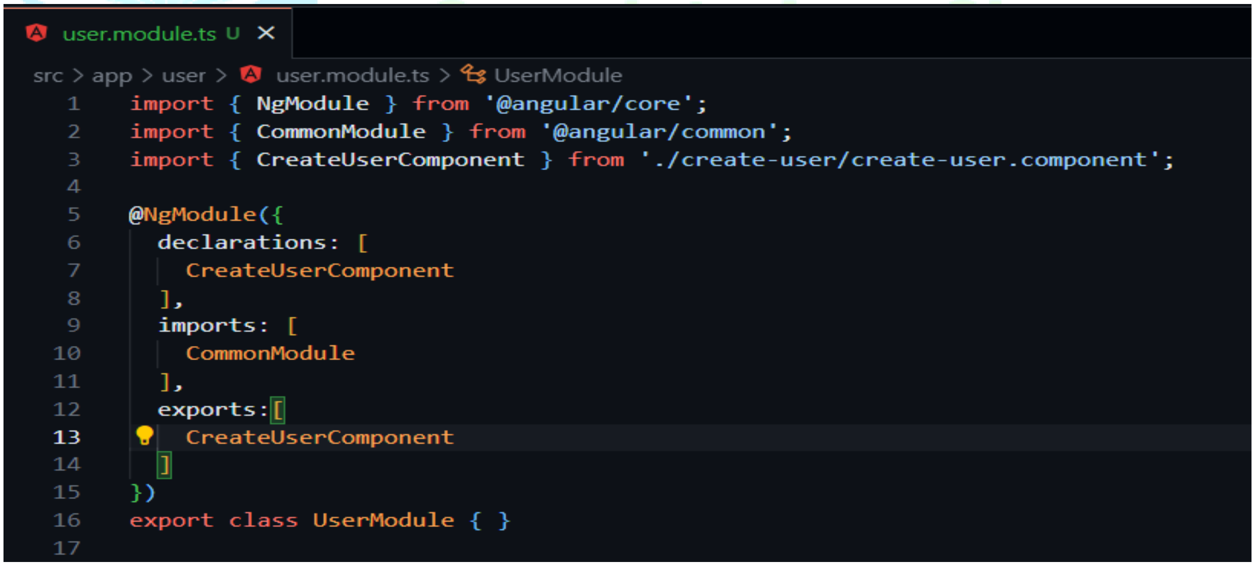
8. Create CreateUserComponent in UserModule using command :
   ng g c user/create-user --skip-tests



```typescript
import { NgModule } from '@angular/core';
import { CommonModule } from '@angular/common';
import { CreateUserComponent } from './create-user/create-user.component';

@NgModule({
  declarations: [
    CreateUserComponent
  ],
  imports: [
    CommonModule
  ]
})
export class UserModule { }
```

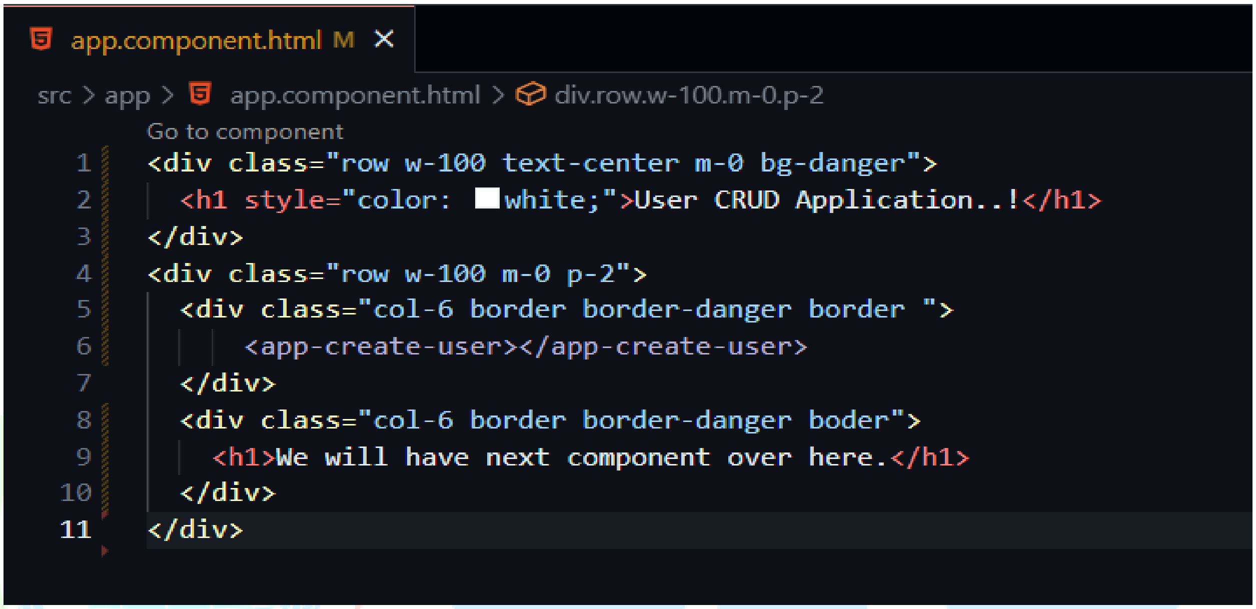
```
PS E:\Angular Workspace\user_crud_app> ng g c user/create-user --skip-tests
CREATE src/app/user/create-user/create-user.component.html (26 bytes)
CREATE src/app/user/create-user/create-user.component.ts (221 bytes)
CREATE src/app/user/create-user/create-user.component.css (0 bytes)
UPDATE src/app/user/user.module.ts (292 bytes)
PS E:\Angular Workspace\user_crud_app>
```

9. Export CreateUserComponent from UserModule.



```typescript
import { NgModule } from '@angular/core';
import { CommonModule } from '@angular/common';
import { CreateUserComponent } from './create-user/create-user.component';

@NgModule({
  declarations: [
    CreateUserComponent
  ],
  imports: [
    CommonModule
  ],
  exports:[
    CreateUserComponent
  ]
})
export class UserModule { }
```

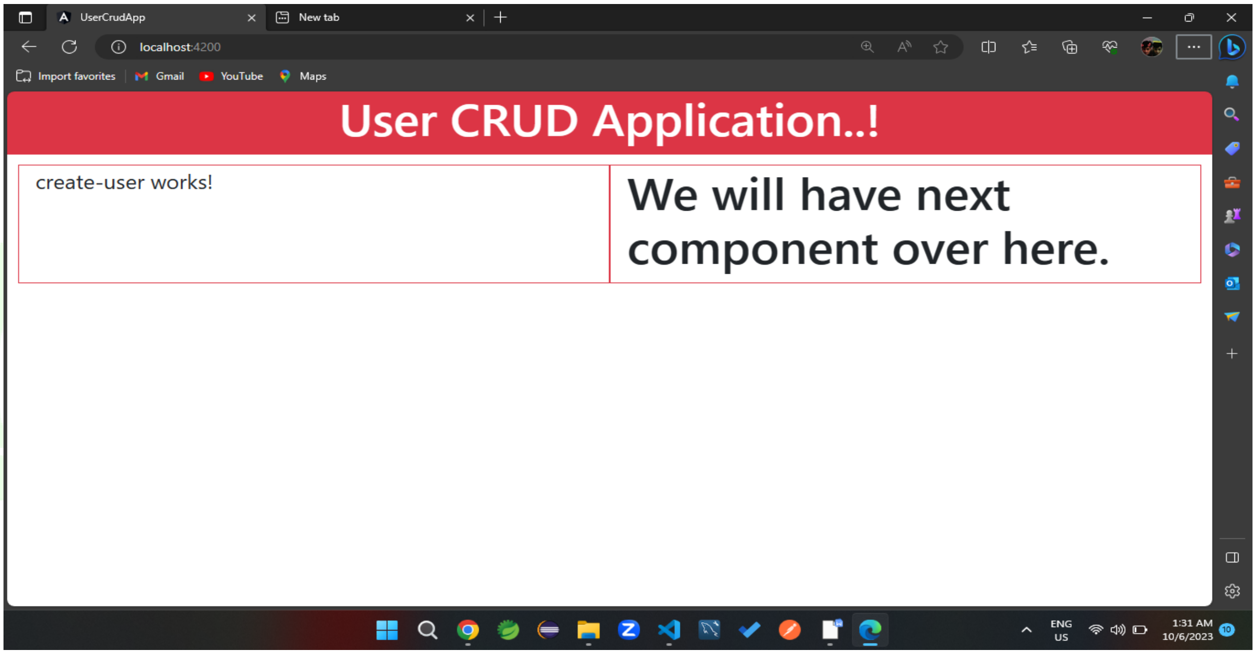
10. Add selector of CreateUserComponent in app.component.html as given bellow.



```
<div class="row w-100 text-center m-0 bg-danger">
  <h1 style="color: white;">User CRUD Application..!</h1>
</div>
<div class="row w-100 m-0 p-2">
  <div class="col-6 border border-danger border ">
    <app-create-user></app-create-user>
  </div>
  <div class="col-6 border border-danger boder">
    <h1>We will have next component over here.</h1>
  </div>
</div>
```

```
<div class="row w-100 text-center m-0 bg-danger">
<h1 style="color: white;">User CRUD Application..!</h1>
</div>
<div class="row w-100 m-0 p-2">
<div class="col-6 border border-danger border ">
<app-create-user></app-create-user>
</div>
<div class="col-6 border border-danger boder">
<h1>We will have next component over here.</h1>
</div>
</div>
```

- Unto now our output should look like as below.

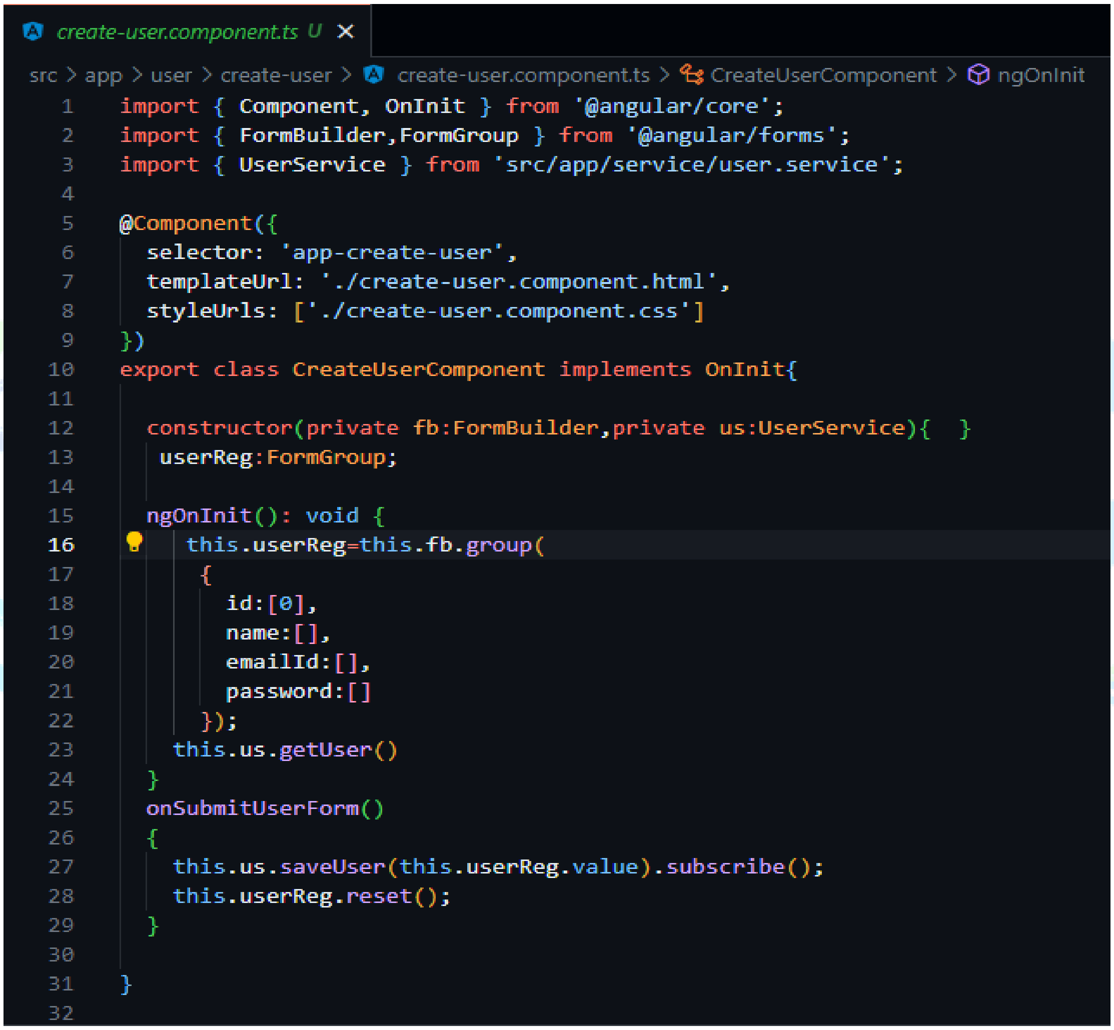


User CRUD Application..!

create-user works!

We will have next component over here.

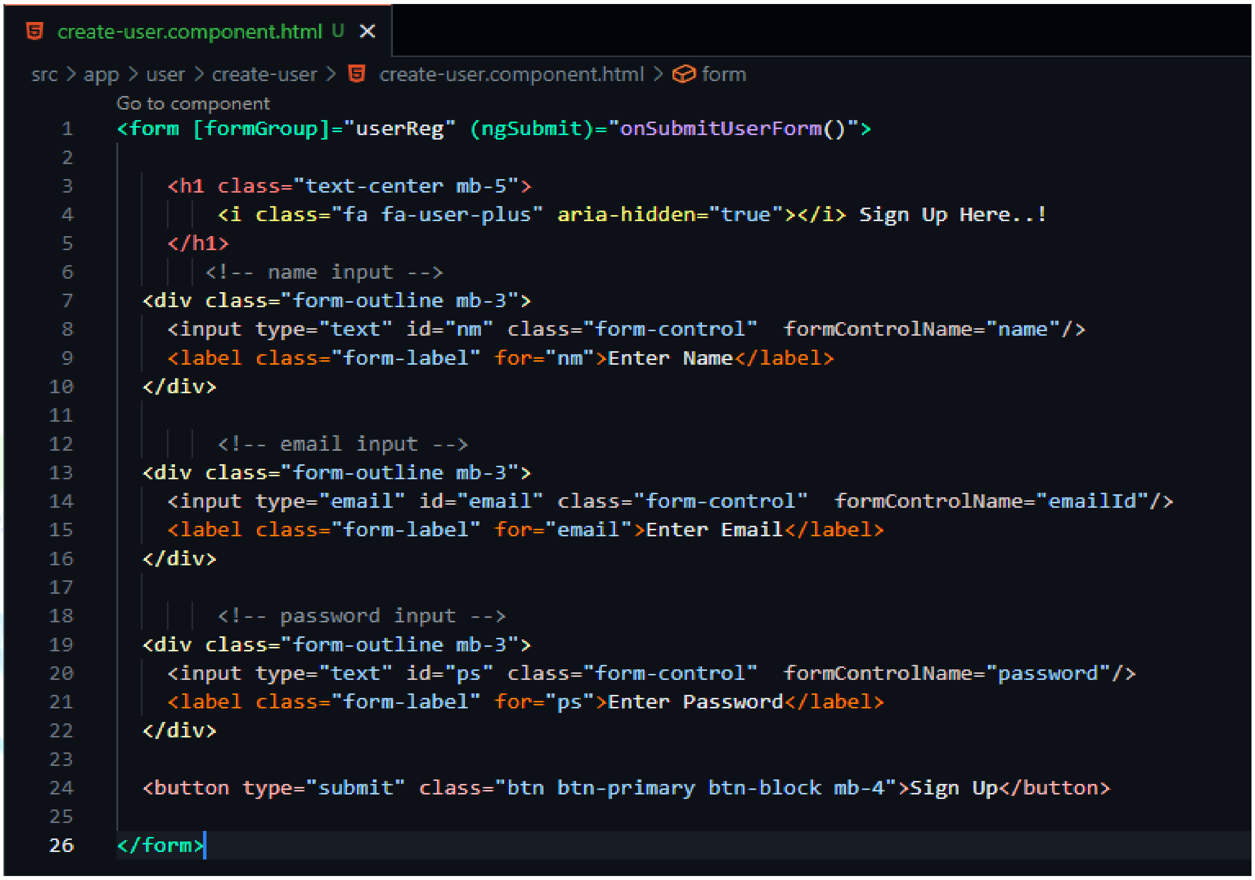
11. In create-user.component.ts inject FormBuilder and UserService also declare user-registration form and initialize it in ngOnInit hook then declare on function to subscribe
saveUser function of a UserService.


create-user.component.ts U ×
src > app > user > create-user > create-user.component.ts > CreateUserComponent > ngOnInit

```
import { Component, OnInit } from '@angular/core';
import { FormBuilder,FormGroup } from '@angular/forms';
import { UserService } from 'src/app/service/user.service';

@Component({
  selector: 'app-create-user',
  templateUrl: './create-user.component.html',
  styleUrls: ['./create-user.component.css']
})
export class CreateUserComponent implements OnInit{

  constructor(private fb:FormBuilder,private us:UserService){   }
   userReg:FormGroup;

  ngOnInit(): void {
     this.userReg=this.fb.group(
      {
        id:[0],
        name:[],
        emailId:[],
        password:[]
      });
    this.us.getUser()
  }
  onSubmitUserForm()
  {
    this.us.saveUser(this.userReg.value).subscribe();
    this.userReg.reset();
  }

}

```

CRUD-IN-ANGULAR

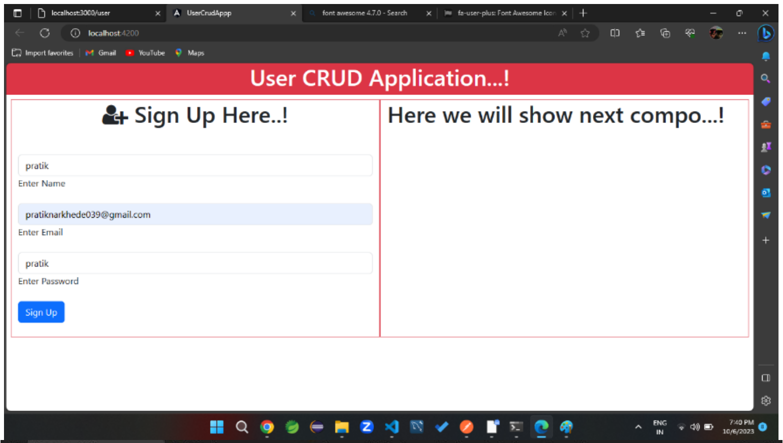
12. In create-user.component.html bind the user-registration form and on ngSubmit event call onSubmitUserForm function.



create-user.component.html U ×
src > app > user > create-user > create-user.component.html > form
Go to component

```
<form [formGroup]="userReg" (ngSubmit)="onSubmitUserForm()">

    <h1 class="text-center mb-5">
        <i class="fa fa-user-plus" aria-hidden="true"></i> Sign Up Here..!
    </h1>
       <!-- name input -->
  <div class="form-outline mb-3">
    <input type="text" id="nm" class="form-control"  formControlName="name"/>
    <label class="form-label" for="nm">Enter Name</label>
  </div>

        <!-- email input -->
  <div class="form-outline mb-3">
    <input type="email" id="email" class="form-control"  formControlName="emailId"/>
    <label class="form-label" for="email">Enter Email</label>
  </div>

        <!-- password input -->
  <div class="form-outline mb-3">
    <input type="text" id="ps" class="form-control"  formControlName="password"/>
    <label class="form-label" for="ps">Enter Password</label>
  </div>

  <button type="submit" class="btn btn-primary btn-block mb-4">Sign Up</button>

</form>
```

OUTPUT:-



User CRUD Application...!

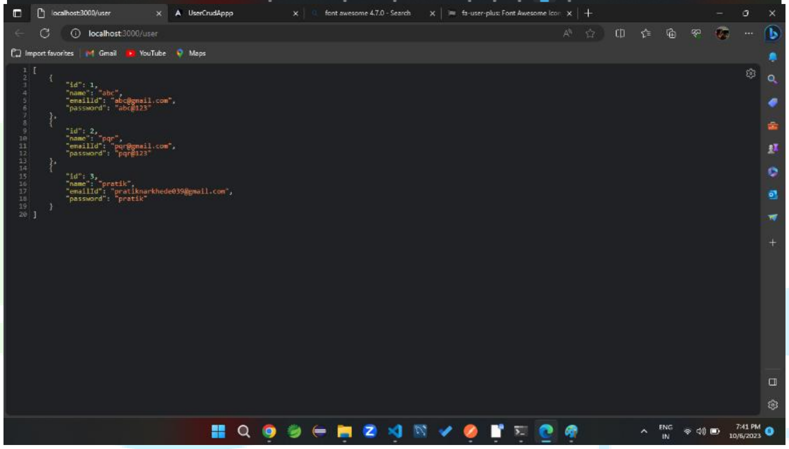
Sign Up Here..!

pratik
Enter Name

pratiknarkhede039@gmail.com
Enter Email

pratik
Enter Password

Here we will show next compo...!



```
[
    {
        "id": 1,
        "name": "abc",
        "emailId": "abc@gmail.com",
        "password": "abc@123"
    },
    {
        "id": 2,
        "name": "pqr",
        "emailId": "pqr@gmail.com",
        "password": "pqr@123"
    },
    {
        "id": 3,
        "name": "pratik",
        "emailId": "pratiknarkhede039@gmail.com",
        "password": "pratik"
    }
]
```