**Shahzadi Begum Shaikh Rafique**

**Angular Assignment : User Management App**

* **Install Angular**

npm install -g @angular/cli

* **Update npm to Latest version**

npm i npm@latest -g

* **To install a specific version, you can use npm install -g @angular/cli@1.4.9**
* **Create new project**

ng new user-management-app

or

skip to install node modules while creating project

ng new user-management-app –skip-install

npm install [run this command separately to load all dependencies]

* Run your application to get started

cd project Name

ng serve or ng serve -- open

* Create Components

ng g component home

ng g component login

ng g component add-user

ng g component edit-user

ng g component list-user

* Angular CLI Useful Commands

ng g component my-new-component

ng g directive my-new-directive

ng g pipe my-new-pipe

ng g service my-new-service

ng g class my-new-class

ng g guard my-new-guard

ng g interface my-new-interface

ng g enum my-new-enum

ng g module my-module

* Inside app-routing.module.ts file

import { NgModule } from '@angular/core';

// Import RouterModule

import { Routes, RouterModule } from '@angular/router';

import { LoginComponent } from './login/login.component';

import { AddUserComponent } from './add-user/add-user.component';

import { ListUserComponent } from './list-user/list-user.component';

import { EditUserComponent } from './edit-user/edit-user.component';

import { HomeComponent } from './home/home.component';

// Each route maps a URL path to a component

const routes: Routes = [

  // This route specifies the route to redirect to if the path

// is empty. In our case we are redirecting to /home

// pathMatch property value can be full or prefix. For now we

// will set it to full as we want to do a full match

  //{ path: '',redirectTo:'/home',pathMatch:'full' },

  { path: '',component: HomeComponent },

  { path: 'home',component:HomeComponent},

  { path: 'login', component: LoginComponent },

  { path: 'add-user', component: AddUserComponent },

  { path: 'list-user', component: ListUserComponent },

  { path: 'edit-user', component: EditUserComponent },

  // {path : '', component : LoginComponent}, // or

  { path: '\*\*', component: HomeComponent }

];

@NgModule({

  imports: [RouterModule.forRoot(routes)],

  exports: [RouterModule]

})

export class AppRoutingModule { }

* Create a JSON server that serves the data.

npm install -g json-server

* Run below command on terminal

json-server --watch db.json

* Add the data in db.json file

[

{

"id": 1,

"firstName": "Shazli",

"lastName": "Shaikh",

"email": "Shazli@gmail.com"

},

{

"id": 2,

"firstName": "Safa",

"lastName": "Siddiqui",

"email": "Safa@gmail.com"

},

{

"id": 3,

"firstName": "Rehan",

"lastName": "Mohammed",

"email": "Rehan@gmail.com"

},

{

"id": 4,

"firstName": "Shahzadi",

"lastName": "Shaikh",

"email": "Shahzadi@gmail.com"

},

{

"id": 5,

"firstName": "New",

"lastName": "User",

"email": "newuser@gmail.com"

}

]

* Run this on browser with the below given url

<http://localhost:3000/users>

* Create new folder model and add new file user.model.ts

export class User

{

    "id": number;

    "firstName": string;

    "lastName": string;

    "email": string;

}

* Create new folder services inside app folder and then create user service by typing below command

ng g s user

and Add below code inside user.service.ts

import { Injectable } from '@angular/core';

import { HttpClient } from '@angular/common/http';

import { User } from '../model/user.model';

@Injectable()

export class UserService {

  constructor(private http:HttpClient) { }

  baseUrl:string = 'http://localhost:3000/users';

  // Get All Users

  getUsers(){

    return this.http.get<User[]>(this.baseUrl);

  }

  // Get User By Id

  getUserById(id: number){

    return this.http.get<User>(this.baseUrl+'/'+id);

  }

  // Create User

  createUser(user: User) {

    return this.http.post(this.baseUrl, user);

  }

  // Modify User

  updateUser(user: User) {

    return this.http.put(this.baseUrl + '/' + user.id, user);

  }

  // Delete User

  deleteUser(id: number) {

    return this.http.delete(this.baseUrl + '/' + id);

  }

}

* Create new folder images inside assets folder and add one image
* Register your services and routing here, App.module.ts file

import { BrowserModule } from '@angular/platform-browser';

import { NgModule } from '@angular/core';

import { AppRoutingModule } from './app-routing.module';

import { AppComponent } from './app.component';

import { LoginComponent } from './login/login.component';

import { AddUserComponent } from './add-user/add-user.component';

import { EditUserComponent } from './edit-user/edit-user.component';

import { ListUserComponent } from './list-user/list-user.component';

import { ReactiveFormsModule } from '@angular/forms';

import { HttpClientModule } from '@angular/common/http';

import { UserService } from './services/user.service';

import { HomeComponent } from './home/home.component';

@NgModule({

  declarations: [

    AppComponent,

    LoginComponent,

AddUserComponent,

EditUserComponent,

    ListUserComponent,

    HomeComponent

  ],

  imports: [

    BrowserModule,

    AppRoutingModule,

    ReactiveFormsModule,

    HttpClientModule

  ],

  providers: [UserService],

  bootstrap: [AppComponent]

})

export class AppModule { }

* Modify App.component.html with below code

<!--The content below is only a placeholder and can be replaced.-->

<div style="color:white; background-color: purple; padding: 10px; ">

  <h2 style="text-align:center; margin-top: 20px; ">User Management Angular App ...! </h2>

</div>

<hr/>

<div class="container">

  <div class="row">

    <div class="col-md-8">

    </div>

    <div class="col-md-4">

      <br/>

        <img src="../assets/Image/Group Pic.png" width="300" height="300" style="right: 100px;"/>

    </div>

  </div>

  <br/><br/>

  <div>

      <footer><span class="text-danger"><strong>This site is copyright &copy; to LTI.

        All rights are reserved here ..!</strong></span></footer>

  </div>

</div>

<router-outlet></router-outlet>

* Add below code inside [home.component.html](http://home.component.html/) file

<br/><br/>

  <div class="col-md-6 col-md-offset-2">

    <h3>Angular</h3>

    <br/>

    <p>

      </p>

      <h5>In this demo, we will learn about </h5>

      <ul class="text-success" style="font-weight: bold">

        <li>

            Angular Architecture

            <ul>

              <li>Modules</li>

              <li>Components

                  <ol>

                    <li>Root Component or App Component</li>

                  </ol>

              </li>

              <li>Services and Dependency Injection</li>

              <li>Models</li>

              <li>Directives</li>

              <li>Pipes, etc</li>

            </ul>

        </li>

      </ul>

  </div>

  <a class="text-right" [routerLink]="['/login']"><h3>Login..</h3></a>

* Modify login.component.html file

<br/><br/>

  <div class="col-md-6 col-md-offset-2">

    <h3>Login</h3>

    <br/>

    <form [formGroup]="loginForm" (ngSubmit)="onSubmit()">

      <div class="form-group">

        <label for="email">Email address:</label>

        <input type="email" class="form-control" formControlName="email" id="email">

        <!-- <div \*ngIf="submitted && loginForm.controls.email.errors" class="error">

          <div class="text-danger" \*ngIf="loginForm.controls.email.errors.required">Email is required</div>

        </div> -->

      </div>

      <div class="form-group">

        <label for="pwd">Password:</label>

        <input type="password" class="form-control" formControlName="password" id="pwd">

        <!-- <div \*ngIf="submitted && loginForm.controls.password.errors" class="error">

          <div class="text-danger" \*ngIf="loginForm.controls.password.errors.required">Password is required</div>

        </div> -->

      </div>

      <button class="btn btn-success">Login</button> &nbsp; &nbsp;

      <button class="btn btn-danger" type="reset">Cancel</button>

      <br/><br/>

      <div \*ngIf="invalidLogin" class="error">

        <div class="text-danger">Invalid credentials.</div>

      </div>

    </form>

  </div>

* Modify **login.component.ts file**

import { Component, OnInit } from '@angular/core';

import {FormBuilder, FormGroup, Validators} from "@angular/forms";

import {Router} from "@angular/router";

@Component({

  selector: 'app-login',

  templateUrl: './login.component.html',

  styleUrls: ['./login.component.css']

})

export class LoginComponent implements OnInit {

  loginForm!: FormGroup;

  submitted: boolean = false;

  invalidLogin: boolean = false;

  constructor(private formBuilder: FormBuilder, private router: Router, )

  { }

  onSubmit()

  {

    this.submitted = true;

    if(this.loginForm.invalid){

      return;

  }

    if(this.loginForm.controls['email'].value =="Shahzadi@gmail.com" &&

        this.loginForm.controls['password'].value=="123")

        {

          localStorage.setItem("username",this.loginForm.controls['email'].value);

          this.router.navigate(['list-user']);

        }

    else

    {

      this.invalidLogin = true;

    }

  }

  ngOnInit() {

    this.loginForm = this.formBuilder.group({

      email: ['', Validators.required],

      password: ['', Validators.required]

    });

  }

}

* Modify list-user.component.html

<br/><br/>

<div class="col-lg-12">

  <h2> User Details</h2>

  <br/>

  <button class="btn btn-success" (click)="addUser()"> Add User</button> &nbsp;&nbsp;

  <button class="btn btn-danger" (click)="logOutUser()"> LogOff</button>

  <br/><br/>

  <table class="table table-striped table-responsive table-hover">

    <thead>

    <tr>

      <th class="text-hide">Id</th>

      <th>First Name</th>

      <th>Last Name</th>

      <th>Email</th>

      <th>Action</th>

    </tr>

    </thead>

    <tbody>

    <tr \*ngFor="let user of users">

      <td class="text-hide">{{user.id}}</td>

      <td>{{user.firstName | uppercase }}</td>

      <td>{{user.lastName |  lowercase }}</td>

      <td>{{user.email | titlecase }}</td>

      <td><button class="btn btn-danger" (click)="deleteUser(user)"> Delete</button>

        <button class="btn btn-warning" (click)="editUser(user)" style="margin-left: 20px;"> Edit</button></td>

    </tr>

    <tr \*ngIf="!users || users.length==0">

      <th colspan="5">

        <p class="text-right text-danger">Sorry No Users Available</p>

      </th>

    </tr>

    </tbody>

  </table>

</div>

* Modify **list-user.component.ts**

import { Component, OnInit } from '@angular/core';

import { User } from '../model/user.model';

import { UserService } from '../services/user.service';

import { Router } from '@angular/router';

@Component({

  selector: 'app-list-user',

  templateUrl: './list-user.component.html',

  styleUrls: ['./list-user.component.css']

})

export class ListUserComponent implements OnInit {

  users!: User[];

  constructor(private router: Router, private userService: UserService) { }

  // Initialize with default list of users

  ngOnInit() {

    if(localStorage.getItem("username")!=null){

    this.userService.getUsers()

      .subscribe(data=> {

        this.users = data;

      });

    }

    else

    this.router.navigate(['/login']);

  }

  // logOff user

  logOutUser():void{

      if(localStorage.getItem("username")!=null){

        localStorage.removeItem("username");

        this.router.navigate(['/login']);

      }

  }

// Delete User

deleteUser(user: User): void {

      let result = confirm('Do you want to delete the user?')

      if(result)

      {

        this.userService.deleteUser(user.id)

          .subscribe( \_data => {

            this.users = this.users.filter(u => u !== user);

          });

        }

};

// Modify USer

editUser(user: User): void {

  localStorage.removeItem("editUserId");

  localStorage.setItem("editUserId", user.id.toString());

  this.router.navigate(['edit-user']);

};

// Add New User

addUser(): void {

  this.router.navigate(['add-user']);

};

}

* **Modify add-user.component.ts**

<br/><br/>

<div class="col-md-6 col-md-offset-2">

  <h2>Add User</h2>

  <br/>

  <form [formGroup]="addForm" (ngSubmit)="onSubmit()">

    <div class="form-group">

      <label for="firstName">First Name:</label>

      <input formControlName="firstName" placeholder="First Name" name="firstName" class="form-control" id="firstName">

      <!-- <div \*ngIf="submitted && addForm.controls.firstName.errors" class="error">

          <div class="text-danger" \*ngIf="addForm.controls.firstName.errors.required">FirstName is required</div>

        </div> -->

    </div>

    <div class="form-group">

      <label for="lastName">Last Name:</label>

      <input formControlName="lastName" placeholder="Last name" name="lastName" class="form-control" id="lastName">

      <!-- <div \*ngIf="submitted && addForm.controls.lastName.errors" class="error">

          <div class="text-danger" \*ngIf="addForm.controls.lastName.errors.required">LastName is required</div>

        </div> -->

    </div>

    <div class="form-group">

      <label for="email">Email address:</label>

      <input type="email" formControlName="email" placeholder="Email" name="email" class="form-control" id="email">

      <!-- <div \*ngIf="submitted && addForm.controls.email.errors" class="error">

          <div class="text-danger" \*ngIf="addForm.controls.email.errors.required">Email is required</div>

        </div> -->

    </div>

    <button class="btn btn-success">Save</button>&nbsp; &nbsp;

    <button class="btn btn-danger" type="reset">Cancel</button>

&nbsp; &nbsp;

    <a class="btn btn-warning" href="/list-user">Back to List</a>

    <br/><br/>

  </form>

</div>

* Modify **edit-user.component.html**

<br/><br/>

<div class="col-md-6 col-md-offset-2">

  <h2>Edit User        &nbsp; &nbsp; &nbsp; &nbsp; &nbsp; &nbsp;

  <button class="btn btn-danger" (click)="logOutUser()"> LogOff</button> </h2>

  <br/>

  <form [formGroup]="editForm" (ngSubmit)="onSubmit()">

    <div class="form-group">

      <label for="firstName">First Name:</label>

      <input formControlName="firstName" placeholder="First Name" name="firstName" class="form-control" id="firstName">

      <!-- <div \*ngIf="submitted && editForm.controls.firstName.errors" class="error">

        <div class="text-danger" \*ngIf="editForm.controls.firstName.errors.required">FirstName is required</div>

      </div> -->

    </div>

    <div class="form-group">

      <label for="lastName">Last Name:</label>

      <input formControlName="lastName" placeholder="Last name" name="lastName" class="form-control" id="lastName">

      <!-- <div \*ngIf="submitted && editForm.controls.lastName.errors" class="error">

        <div class="text-danger" \*ngIf="editForm.controls.lastName.errors.required">LastName is required</div>

      </div> -->

    </div>

    <div class="form-group">

      <label for="email">Email address:</label>

      <input type="email" formControlName="email" placeholder="Email" name="email" class="form-control" id="email">

      <!-- <div \*ngIf="submitted && editForm.controls.email.errors" class="error">

        <div class="text-danger" \*ngIf="editForm.controls.email.errors.required">Email is required</div>

      </div> -->

    </div>

<button class="btn btn-success">Update</button>&nbsp; &nbsp;

    <a class="btn btn-danger" href="/list-user">Cancel</a>

    <br/><br/>

  </form>

</div>

* Modify **edit-user.component.ts**

import { Component, OnInit } from '@angular/core';

import { User } from '../model/user.model';

import { FormGroup, FormBuilder, Validators } from '@angular/forms';

import { Router } from '@angular/router';

import { UserService } from '../services/user.service';

import {first} from 'rxjs/operators';

@Component({

  selector: 'app-edit-user',

  templateUrl: './edit-user.component.html',

  styleUrls: ['./edit-user.component.css']

})

export class EditUserComponent implements OnInit {

  user!: User;

  editForm!: FormGroup;

  submitted: boolean = false;

  constructor(private formBuilder: FormBuilder,private router: Router,

    private userService: UserService) { }

  ngOnInit() {

    if(localStorage.getItem("username")!=null){

    let userId = localStorage.getItem("editUserId");

    if(!userId) {

      alert("Invalid action.")

      this.router.navigate(['list-user']);

      return;

    }

    this.editForm = this.formBuilder.group({

      id: [],

      email: ['', Validators.required],

      firstName: ['', Validators.required],

      lastName: ['', Validators.required]

    });

    this.userService.getUserById(+userId)

      .subscribe( data => {

        this.editForm.setValue(data);

      });

    }

    else

        this.router.navigate(['/login']);

  }

  onSubmit() {

    this.submitted = true;

    if(this.editForm.invalid){

      return;

    }

    this.userService.updateUser(this.editForm.value)

      .pipe(first())

      .subscribe(

        \_data => {

          this.router.navigate(['list-user']);

        },

        error => {

          alert(error);

        });

  }

   // logOff user

   logOutUser():void{

    if(localStorage.getItem("username")!=null){

      localStorage.removeItem("username");

      this.router.navigate(['/login']);

    }

}

}

**OutPut**

















