**dashboard.component.spec.ts**

import { ComponentFixture, TestBed } from '@angular/core/testing';

import { DashboardComponent } from './dashboard.component';

describe('DashboardComponent', () => {

let component: DashboardComponent;

let fixture: ComponentFixture<DashboardComponent>;

beforeEach(async () => {

await TestBed.configureTestingModule({

declarations: [ DashboardComponent ]

})

.compileComponents();

});

beforeEach(() => {

fixture = TestBed.createComponent(DashboardComponent);

component = fixture.componentInstance;

fixture.detectChanges();

});

it('should create', () => {

expect(component).toBeTruthy();

});

});

**dashboard.component.ts**

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

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

import { CommonService } from 'src/app/common.service';

@Component({

selector: 'app-dashboard',

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

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

})

export class DashboardComponent implements OnInit {

userList: any = []

constructor(private commServ:CommonService, private router: Router) { }

ngOnInit(): void {

this.userList = [localStorage.getItem("name"), localStorage.getItem("email")] ;

console.log(this.userList);

}

logout(){

localStorage.clear() ;

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

}

}

**employee.component.spec.ts**

import { ComponentFixture, TestBed } from '@angular/core/testing';

import { EmployeeComponent } from './employee.component';

describe('EmployeeComponent', () => {

let component: EmployeeComponent;

let fixture: ComponentFixture<EmployeeComponent>;

beforeEach(async () => {

await TestBed.configureTestingModule({

declarations: [ EmployeeComponent ]

})

.compileComponents();

});

beforeEach(() => {

fixture = TestBed.createComponent(EmployeeComponent);

component = fixture.componentInstance;

fixture.detectChanges();

});

it('should create', () => {

expect(component).toBeTruthy();

});

});

**employee.component.ts**

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

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

import { EmployeeService } from '../../shared/employee.service';

import { Employee } from '../../shared/employee.model' ;

// import { map } from 'rxjs/operators';

declare var M: any;

@Component({

selector: 'app-employee',

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

styleUrls: ['./employee.component.css'],

providers: [EmployeeService]

})

export class EmployeeComponent implements OnInit {

constructor(public employeeService: EmployeeService) { }

ngOnInit() {

this.resetForm();

this.refreshEmployeeList();

}

resetForm(form?: NgForm) {

if (form)

form.reset();

this.employeeService.selectedEmployee = {

\_id: "",

name: "",

position: "",

office: "",

salary: null

}

}

onSubmit(form: NgForm) {

if (form.value.\_id == "") {

this.employeeService.postEmployee(form.value).subscribe((res) => {

this.resetForm(form);

this.refreshEmployeeList();

M.toast({ html: 'Saved successfully', classes: 'rounded' });

});

}

else {

this.employeeService.putEmployee(form.value).subscribe((res) => {

this.resetForm(form);

this.refreshEmployeeList();

M.toast({ html: 'Updated successfully', classes: 'rounded' });

});

}

}

refreshEmployeeList() {

this.employeeService.getEmployeeList().subscribe((res) => {

this.employeeService.employees = res as Employee[];

});

}

onEdit(emp: Employee) {

this.employeeService.selectedEmployee = emp;

}

onDelete(\_id: string, form: NgForm) {

if (confirm('Are you sure to delete this record ?') == true) {

this.employeeService.deleteEmployee(\_id).subscribe((res) => {

this.refreshEmployeeList();

this.resetForm(form);

M.toast({ html: 'Deleted successfully', classes: 'rounded' });

});

}

}

}

**login.component.spec.ts**

import { ComponentFixture, TestBed } from '@angular/core/testing';

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

describe('LoginComponent', () => {

let component: LoginComponent;

let fixture: ComponentFixture<LoginComponent>;

beforeEach(async () => {

await TestBed.configureTestingModule({

declarations: [ LoginComponent ]

})

.compileComponents();

});

beforeEach(() => {

fixture = TestBed.createComponent(LoginComponent);

component = fixture.componentInstance;

fixture.detectChanges();

});

it('should create', () => {

expect(component).toBeTruthy();

});

});

**login.component.ts**

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

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

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

import { CommonService } from 'src/app/common.service';

@Component({

selector: 'app-login',

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

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

})

export class LoginComponent implements OnInit {

login: any = FormGroup;

users:any = [] ;

constructor(private fb: FormBuilder, private router: Router, private commserv: CommonService) {}

ngOnInit(): void {

this.login = this.fb.group({

name: ['', Validators.required],

email: ['', Validators.compose([Validators.required, Validators.email])],

});

this.commserv.getUser().subscribe((data:any)=>{

this.users = data ;

})

}

loginSubmit(data:any){

let flag = 1 ;

console.log(data)

if(data.name){

this.users.forEach((item:any) => {

if(item.name === data.name && item.email === data.email){

localStorage.setItem("isLoggedIn","true");

localStorage.setItem("name",data.name);

localStorage.setItem("email",data.email);

flag = 1 ;

this.router.navigate(['home']);

}

else{

localStorage.clear();

flag = 0 ;

}

});

}

if(flag === 0){

alert("Invalid") ;

}

}

gotoSignup() {

this.router.navigate(['register']);

}

}

**register.component.spec.ts**

import { ComponentFixture, TestBed } from '@angular/core/testing';

import { RegisterComponent } from './register.component';

describe('RegisterComponent', () => {

let component: RegisterComponent;

let fixture: ComponentFixture<RegisterComponent>;

beforeEach(async () => {

await TestBed.configureTestingModule({

declarations: [ RegisterComponent ]

})

.compileComponents();

});

beforeEach(() => {

fixture = TestBed.createComponent(RegisterComponent);

component = fixture.componentInstance;

fixture.detectChanges();

});

it('should create', () => {

expect(component).toBeTruthy();

});

});

**register.component.ts**

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

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

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

import { CommonService } from 'src/app/common.service';

@Component({

selector: 'app-register',

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

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

})

export class RegisterComponent implements OnInit {

register: any = FormGroup;

id: any = Math.floor(Math.random()) ;

constructor(private fb: FormBuilder, private router: Router, private commServ:CommonService) {}

ngOnInit(): void {

this.register = this.fb.group({

name: ['', Validators.required],

email: ['', Validators.compose([Validators.required, Validators.email])],

});

}

registerSubmit(data: any) {

console.log(data);

let dataToPass = {

name: data.name, email:data.email ,

id:this.id++

}

this.commServ.addUser(dataToPass).subscribe((data:any)=>{

console.log(data);

alert("User registered")

})

}

gotoLogin() {

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

}

}

**OUTPUT:**





