**Assignment 2-C**

Create an Angular application which will do following actions: Register User, Login User, Show User Data on Profile Component.

Code –

1.test.ts

*// This file is required by karma.conf.js and loads recursively all the .spec and framework files*

*import* 'zone.js/testing';

*import* { getTestBed } *from* '@angular/core/testing';

*import* {

  BrowserDynamicTestingModule,

  platformBrowserDynamicTesting

} *from* '@angular/platform-browser-dynamic/testing';

declare const require: {

  context(*path*: string, *deep*?: boolean, *filter*?: RegExp): {

    <T>(*id*: string): T;

    keys(): string[];

  };

};

*// First, initialize the Angular testing environment.*

getTestBed().initTestEnvironment(

  BrowserDynamicTestingModule,

  platformBrowserDynamicTesting(),

);

*// Then we find all the tests.*

const context = require.context('./', true, /\.spec\.ts*$*/);

*// And load the modules.*

context.keys().map(context);

2.main.ts

*import* { enableProdMode } *from* '@angular/core';

*import* { platformBrowserDynamic } *from* '@angular/platform-browser-dynamic';

*import* { AppModule } *from* './app/app.module';

*import* { environment } *from* './environments/environment';

*if* (environment.production) {

  enableProdMode();

}

platformBrowserDynamic().bootstrapModule(AppModule)

  .catch(*err* => console.error(*err*));

3.dashboard.component.html

<div class="container wrapper">

    <div class="">

      <h3>User profile</h3>

        <h5>Name: {{userList[0]}}</h5>

        <h5>Email: {{userList[1]}}</h5>

        <button class="btn btn-primary" (click)="logout()">Logout</button>

    </div>

  </div>

4. 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();

  });

});

5. 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']) ;

  }

}

6. login.component.html

<div class="container wrapper">

  <div class="block">

    <h3>Login</h3>

    <form [formGroup]="login">

      <input type="text" class="form-control" placeholder="Enter Name" formControlName="name" />

      <input type="text" class="form-control" placeholder="Enter Email" formControlName="email" />

      <div class="btn-group">

        <button class="btn btn-primary" type="button" (click)="loginSubmit(login.value)" [disabled]="login.invalid">

          Login

        </button>

        <button class="btn btn-primary" (click)="gotoSignup()">Register</button>

      </div>

    </form>

  </div>

</div>

7. dashboard.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();

  });

});

8. dashboard.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']);

  }

}

9. login.component.html

<div class="container wrapper">

    <div class="block">

      <h3>Register</h3>

      <form [formGroup]="register">

        <input type="text" class="form-control" placeholder="Enter Name" formControlName="name" />

        <input type="text" class="form-control" placeholder="Enter Email" formControlName="email" />

        <div class="btn-group">

          <button class="btn btn-primary" (click)="gotoLogin()" >

            Login

          </button>

          <button class="btn btn-primary" (click)="registerSubmit(register.value)" [disabled]="register.invalid">Register</button>

        </div>

      </form>

    </div>

  </div>

10. login.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();

  });

});

11. 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-register',

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

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

})

*export* class RegisterComponent implements OnInit {

  register: any = FormGroup;

  id: any = 7 ;

  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(['']);

  }

}

12.db.js

{

  "user": [

    {

      "name": "John",

      "email": "John@gmail.com",

      "id": 1

    },

    {

      "name": "Bob",

      "email": "Bob@gmail.com",

      "id": 2

    },

    {

      "name": "Alice",

      "email": "Alice@gmail.com",

      "id": 3

    },

    {

      "name": "disha",

      "email": "disha@gmail.com",

      "id": 4

    },

    {

      "name": "user1",

      "email": "user1@gmail.com",

      "id": 5

    },

    {

      "name": "disha123",

      "email": "disha123@gmail.com",

      "id": 7

    },

    {

      "name": "Disha",

      "email": "Disha@gmail.com",

      "id": 8

    }

  ]

}

**Output –**

