Sourcecode:

dashboard.component.html

<divclass="container-fluid">

    <h2style="text-align: center;">Dashboard</h2>

    <imgsrc="https://as1.ftcdn.net/v2/jpg/03/48/39/74/1000\_F\_348397404\_wXuf22GUPNAh67htBZZnaDSx3Bj92yep.jpg"

            width="100%"style="opacity:1 ;">

</div>

signin.component.html

<divclass="container-fluid"style="padding-top: 40px; background-color: whitesmoke ;height: 500px;">

    <divclass="container">

        <divclass="row">

            <divclass="col-md-6 offset-md-3 ">

                <form[formGroup]="registerForm"(ngSubmit)="OnSubmit()">

                    <h2style="text-align:center">Sign in</h2>

                    <!--email-->

                    <divclass="form-group">

                        <label>Email</label>

                        <inputtype="email"formControlName="email"class="form-control"[ngClass]="{'is-invalid': submitted && f['email'].errors}"

                                placeholder="Email"/>

                        <!--Custom Validations-->

                        <div\*ngIf="submitted && f['email'].errors">

                            <div\*ngIf="submitted && f['email'].errors['required']">

                                Email  is required and Can't Be Empty

                            </div>

                        </div>

                    </div>

                    <!--Password-->

                    <divclass="form-group">

                        <label>Password</label>

                        <inputtype="password"formControlName="password"class="form-control"[ngClass]="{'is-invalid': submitted && f['password'].errors}"

                                placeholder="Password"/>

                        <!--Custom Validations-->

                        <div\*ngIf="submitted && f['password'].errors">

                            <div\*ngIf="submitted && f['password'].errors['required']">

                                Password must be minimum 8 character long and Can't Be Empty

                            </div>

                        </div>

                    </div>

                    <divclass="form-group text-center">

                        <buttonclass="btn btn-primary"routerLink="/dashboard">Login</button>

                    </div>

                </form>

            </div>

        </div>

    </div>

</div>

signin.component.ts

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

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

@Component({

  selector:'app-signin',

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

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

})

exportclassSigninComponentimplementsOnInit {

   registerForm:FormGroup;

  submitted:boolean=false;

  constructor(privatebuilder:FormBuilder) { }

  ngOnInit():void {

    this.registerForm=this.builder.group(

      {

        email:["",[Validators.required,Validators.email]],

        password:["",[Validators.required,Validators.minLength(8)]]

      }

    );

  }

  OnSubmit(){

    this.submitted=true;

    if(this.registerForm.invalid)

      return;

    else

      console.log("Form Submitted Successfully");

  }

  getf(){

    returnthis.registerForm.controls;

  }

}

signup.component.html

<divclass="jumbotron">

    <divclass="container">

        <divclass="row">

            <divclass="col-md-6 offset-md-3">

                <h2style="text-align:center">Sign Up</h2>

                <form[formGroup]="registerForm"(ngSubmit)="OnSubmit()">

                    <divclass="form-group">

                        <label>FirstName</label>

                        <inputtype="text"formControlName="firstName"class="form-control"[ngClass]="{'is-invalid': submitted && f['firstName'].errors}"

                                placeholder="First Name"/>

                        <!--Custom Validations-->

                        <div\*ngIf="submitted && f['firstName'].errors">

                            <div\*ngIf="submitted && f['firstName'].errors['required']">

                                FirstName  is required and Can't Be Empty

                            </div>

                        </div>

                    </div>

                    <!--Last Name-->

                    <divclass="form-group">

                        <label>LastName</label>

                        <inputtype="text"formControlName="lastName"class="form-control"[ngClass]="{'is-invalid': submitted && f['lastName'].errors}"

                        placeholder="Last Name"/>

                        <!--Custom Validations-->

                        <div\*ngIf="submitted && f['lastName'].errors">

                            <div\*ngIf="submitted && f['lastName'].errors['required']">

                                LastName  is required and Can't Be Empty

                            </div>

                        </div>

                    </div>

                    <!--email-->

                    <divclass="form-group">

                        <label>Email</label>

                        <inputtype="email"formControlName="email"class="form-control"[ngClass]="{'is-invalid': submitted && f['email'].errors}"

                        placeholder="Email"/>

                        <!--Custom Validations-->

                        <div\*ngIf="submitted && f['email'].errors">

                            <div\*ngIf="submitted && f['email'].errors['required']">

                                Email  is required and Can't Be Empty

                            </div>

                        </div>

                    </div>

                    <!--Password-->

                    <divclass="form-group">

                        <label>Password</label>

                        <inputtype="password"formControlName="password"class="form-control"[ngClass]="{'is-invalid': submitted && f['password'].errors}"

                        placeholder="Password"/>

                        <!--Custom Validations-->

                        <div\*ngIf="submitted && f['password'].errors">

                            <div\*ngIf="submitted && f['password'].errors['required']">

                                Password must be minimum 8 character long and Can't Be Empty

                            </div>

                        </div>

                    </div>

                    <divclass="form-group text-center">

                        <buttonclass="btn btn-primary"routerLink="/signin">Register</button>

                    </div>

                </form>

            </div>

        </div>

    </div>

</div>

signup.component.ts

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

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

@Component({

  selector:'app-signup',

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

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

})

exportclassSignupComponentimplementsOnInit {

  registerForm:FormGroup;

  submitted:boolean=false;

  constructor(privatebuilder:FormBuilder) { }

  ngOnInit():void {

    this.registerForm=this.builder.group(

      {

        firstName:["",Validators.required],

        lastName:["",Validators.required],

        email:["",[Validators.required,Validators.email]],

        password:["",[Validators.required,Validators.minLength(8)]]

      }

    );

  }

   OnSubmit(){

    this.submitted=true;

    if(this.registerForm.invalid)

      return;

    else

      alert("User Registered");

  }

  getf(){

    returnthis.registerForm.controls;

  }

}

app-routing-module.ts

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

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

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

import { SigninComponent } from'./signin/signin.component';

import { SignupComponent } from'./signup/signup.component';

constroutes:Routes= [

  {"path":"signin",component:SigninComponent},

  {"path":"signup",component:SignupComponent},

  {"path":"dashboard",component:DashboardComponent},

];

@NgModule({

  imports: [RouterModule.forRoot(routes)],

  exports: [RouterModule]

})

exportclassAppRoutingModule { }

app-component.css

.content{

    background-color: whitesmoke;

}

app-component.html

<navclass="navbar bg-dark navbar-dark">

  <arouterLink=" "class="navbar-brand"></a>

  <ulclass="nav ">

    <liclass="nav-item">

      <arouterLink="signin"class="nav-link">Signin</a>

    </li>

    <liclass="nav-item">

      <arouterLink="signup"class="nav-link">Signup</a>

    </li>

  </ul>

</nav>

<router-outlet></router-outlet>

app-component.spec.ts

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

import { RouterTestingModule } from'@angular/router/testing';

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

describe('AppComponent', () => {

  beforeEach(async () => {

    awaitTestBed.configureTestingModule({

      imports: [

        RouterTestingModule

      ],

      declarations: [

        AppComponent

      ],

    }).compileComponents();

  });

  it('should create the app', () => {

    constfixture=TestBed.createComponent(AppComponent);

    constapp=fixture.componentInstance;

    expect(app).toBeTruthy();

  });

  it(`should have as title 'MyPortfolio'`, () => {

    constfixture=TestBed.createComponent(AppComponent);

    constapp=fixture.componentInstance;

    expect(app.title).toEqual('MyPortfolio');

  });

  it('should render title', () => {

    constfixture=TestBed.createComponent(AppComponent);

    fixture.detectChanges();

    constcompiled=fixture.nativeElementasHTMLElement;

    expect(compiled.querySelector('.content span')?.textContent).toContain('MyPortfolio app is running!');

  });

});

app-component.ts

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

@Component({

  selector:'app-root',

  templateUrl:'./app.component.html',

  styleUrls: ['./app.component.css']

})

exportclassAppComponent {

  title='MyPortfolio';

}

app-module.ts

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

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

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

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

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

import { SigninComponent } from'./signin/signin.component';

import { SignupComponent } from'./signup/signup.component';

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

@NgModule({

  declarations: [

    AppComponent,

    SigninComponent,

    SignupComponent,

    DashboardComponent,

  ],

  imports: [

    BrowserModule,

    AppRoutingModule,

    ReactiveFormsModule,

    FormsModule

  ],

  providers: [],

  bootstrap: [AppComponent]

})

exportclassAppModule { }

Output:





