**SimpliLearn**

**Assesment project 1: MyPortFolio**

**Source code**

**Index.html**

<html>

<body>

<form>

fav subject

<select name="subjects">

<option value="c">C</option>

<option value="java">Java</option>

<option value="c#">C#</option>

<option value="python">python</option>

</select>

</form>

</body>

</html>

**Contact.component.html**

<!--\*\*\* CONTACT \*\*\*-->

<h1 id="contactsHeading">Get In Touch</h1>

<div id="contactsContainer">

<form [formGroup]="contactForm" (ngSubmit)="emailForm()">

<div class="form-group">

<label class="control-label">Name:</label>

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

[ngClass]="{'is-invalid': contactForm.get('name').errors && contactForm.get('name').touched}">

<div class="invalid-feedback"

\*ngIf="contactForm.get('name').hasError('required') && contactForm.get('name').touched">

Name is required</div>

<div class="invalid-feedback"

\*ngIf="contactForm.get('name').hasError('pattern') && contactForm.get('name').touched">

Invalid characters in Name</div>

</div>

<div class="form-group">

<label class="control-label">Email:</label>

<input type="email" class="form-control" formControlName="email" placeholder="Enter Email..."

[ngClass]="{'is-invalid': contactForm.get('email').errors && contactForm.get('email').touched}">

<div class="invalid-feedback"

\*ngIf="contactForm.get('email').hasError('required') && contactForm.get('email').touched">

Email is required</div>

<div class="invalid-feedback"

\*ngIf="contactForm.get('email').hasError('email') && contactForm.get('email').touched">

Invalid Email Address</div>

</div>

<div class="form-group">

<label class="control-label">Message:</label>

<textarea type="text" class="form-control" formControlName="message" placeholder="Message..." cols="30" rows="10"

[ngClass]="{'is-invalid': contactForm.get('message').errors && contactForm.get('message').touched}"></textarea>

<div class="invalid-feedback"

\*ngIf="contactForm.get('message').hasError('required') && contactForm.get('message').touched">

Message is required</div>

</div>

<button class="btn btn-success" [disabled]="!contactForm.valid" type="submit" id="formSubmit">Submit</button>

</form>

<div id="iconsContainer">

<a href="https://github.com/lenzoburger" class="contactIcon" id="githubLink" target="\_blank">

<fa-icon [icon]="['fab', 'github']"></fa-icon>

</a>

<a href="https://www.linkedin.com/in/lencho-burka-935b93118/" target="\_blank" class="contactIcon" id="linkedInLink">

<fa-icon [icon]="['fab', 'linkedin']"></fa-icon>

</a>

<a href="mailto:lenzoburger@gmail.com" class="contactIcon" id="emailLink">

</a>

</div>

</div>

<!-- <P>Form value: {{contactForm.value | json}}</P>

<P>Form Status: {{contactForm.status | json}}</P> -->

**Contact.component.ts**

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

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

import { AlertifyService } from '../\_services/alertify.service';

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

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

@Component({

selector: 'app-contact',

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

styleUrls: ['./contact.component.scss']

})

export class ContactComponent implements OnInit {

formApi = 'https://getsimpleform.com/messages?';

formApiToken = 'form\_api\_token=eac98a47b2c53f674a58b358e3c4b3f4';

contactForm: FormGroup;

@HostListener('window:beforeunload', ['$event'])

unloadNotification($event: any) {

if (this.contactForm.dirty) {

$event.returnValue = true;

}

}

constructor(private fb: FormBuilder, private alertify: AlertifyService, private http: HttpClient, private router: Router) { }

ngOnInit() {

this.createEmailForm();

}

createEmailForm() {

this.contactForm = this.fb.group(

{

name: ['', [Validators.required, Validators.pattern('[ a-zA-Z-\_\.\'’‘]\*')]],

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

message: ['', Validators.required]

}

);

}

emailForm() {

const formData = new FormData();

formData.append('name', this.contactForm.get('name').value);

formData.append('email', this.contactForm.get('email').value);

formData.append('message', this.contactForm.get('message').value);

formData.append('redirect\_to', 'https://jsonplaceholder.typicode.com/todos/1');

this.http.post<any>(this.formApi + '' + this.formApiToken, formData)

.subscribe(

(res) => {

this.alertify.success('Your Submission Has Been Sent!');

console.log('Sucess: ' + res);

this.contactForm.reset();

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

},

(err) => {

console.log('Error: ' + err);

this.alertify.error('Something went wrong, please try again later.');

}

);

}

}

**Contact.component.css**

$formLabelWidth: 90px;

$btnSuccesColor: #28a745;

:host {

display: grid;

grid-template-rows: minmax(60px, auto) auto;

place-items: center;

font-family: "Montserrat", sans-serif;

font-weight: normal;

grid-row-gap: 30px;

}

#contactsHeading {

margin-top: 30px;

margin-left: $formLabelWidth;

grid-area: 1 / 1;

}

#contactsContainer {

grid-area: 2 / 1;

display: grid;

grid-template-areas:

" contactForm "

" iconsContainer ";

place-items: center;

grid-row-gap: 30px;

form {

grid-area: contactForm;

display: grid;

place-items: center;

grid-row-gap: 10px;

.form-group{

width: 100%;

display: grid;

grid-template-columns: $formLabelWidth auto;

label {

grid-area: 1 / 1;

}

input, textarea {

grid-area: 1 / 2;

}

.invalid-feedback {

font-weight: bold;

grid-area: 2 / 2

}

}

#formSubmit{

margin-top: 10px;

margin-left: $formLabelWidth;

}

#formSubmit:disabled{

background-color: lighten($btnSuccesColor, 25%);

}

}

}

#iconsContainer {

width: 100%;

grid-area: iconsContainer;

margin-left: $formLabelWidth;

margin-top: 20px;

margin-bottom: 20px;

}

**Home.component.html**

<div id="homeContainer">

<h1 id="myName">Shubham Vivek</h1>

<p id="myRole">Full Stack Engineer</p>

<a class="button btn btn-primary slideInLeft" id="projectBtn" [routerLink]="['/projects']">View Projects</a>

<a class="button btn btn-primary slideInRight" id="resumeBtn" href="assets/resume/LenchoBurka\_CV\_July\_2020\_Portfolio\_Copy.pdf" target="\_blank">View

Resume</a>

<div id="iconsContainer">

<a href="https://github.com/skv0702" class="contactIcon bounceInDown" id="githubLink" target="\_blank">

<fa-icon [icon]="['fab', 'github']"></fa-icon>

</a>

<a href="" target="\_blank" class="contactIcon bounceInDown"

id="https://www.linkedin.com/in/shubham-kumar-vivek-853016b1/">

<fa-icon [icon]="['fab', 'linkedin']"></fa-icon>

</a>

<a href="mailto:shubham.sk.kumar007@gmail.com" class="contactIcon bounceInDown" id="emailLink">

</a>

</div>

</div>

**Home.component.ts**

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

@Component({

selector: 'app-home',

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

styleUrls: ['./home.component.scss']

})

export class HomeComponent implements OnInit {

title = 'Home';

isCollapsed = true;

constructor() { }

ngOnInit() {

}

}

**Home.component.css**

:host {

display: grid;

place-items: center;

font-family: "Montserrat", sans-serif;

font-weight: normal;

}

#homeContainer {

display: grid;

grid-template-areas:

" myName myName "

" myRole myRole "

" projects resume "

" contact contact ";

column-gap: 5px;

place-items: center;

text-align: center;

}

#myName {

grid-area: myName;

font-size: 3.5em;

}

#myRole {

grid-area: myRole;

font-size: 2.5em;

margin-top: 20px;

margin-bottom: 40px;

}

#projectBtn {

grid-area: projects;

font-size: 1.3em;

}

#resumeBtn {

grid-area: resume;

font-size: 1.3em;

}

#iconsContainer {

margin-top: 40px;

grid-area: contact;

width: 100%;

}

#loginBtn{

grid-area: projects;

font-size: 1.3em;

}

**Login.component.html**

<div>

<div class="wrapper fadeInDown">

<div id="formContent">

<div class="fadeIn first">

<h2 class='sign'>Sign In</h2>

<p class="error">{{error}}</p>

</div>

<form>

<input type="text" id="login" class="fadeIn second" name="email"

placeholder="Email"

[(ngModel)]="user.uemail"/>

<br>

<br>

<input type="password" [(ngModel)]="user.upassword"

id="password" class="fadeIn third" name="password"

placeholder="Password"/>

<br>

<br>

<input type="submit" (click)="login()"

class="fadeIn fourth" value="Log In" />

</form>

</div>

</div>

</div>

**Login.component.ts**

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

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

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

import { LoginService } from "../\_services/login.service";

import { User } from "../\_models/User";

@Component({

selector: "app-login",

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

styleUrls: ["./login.component.scss"],

})

export class LoginComponent implements OnInit {

email: string;

password: string;

constructor() {}

ngOnInit(): void {}

home() {

if (

this.email == "rahamthulla556@gmail.com" &&

this.password == "123456789"

) {

}

}

}

**Nav.component.html**

<!--\*\*\* NAVIGATION \*\*\*-->

<nav class="navbar navbar-expand-md navbar-dark bg-dark">

<a class="navbar-brand" [routerLink]="['/home']">

<img src="favicon.ico" width="32" height="32" class="d-inline-block align-top"

alt="">

SHUBHAM VIVEK

</a>

<button class="navbar-toggler" type="button" (click)="isCollapsed = !isCollapsed" (click)="clickedInside($event)"

aria-controls="navbarNav" [attr.aria-expanded]="!isCollapsed"

aria-label="Toggle navigation">

<span class="navbar-toggler-icon"></span>

</button>

<div class="navbar-collapse" id="navbarNav" [collapse]="isCollapsed">

<ul class="navbar-nav ml-auto">

<li class="nav-item" routerLinkActive="active">

<a class="nav-link" [routerLink]="['/home']">Home</a>

</li>

<li class="nav-item" routerLinkActive="active">

<a class="nav-link" [routerLink]="['/login']">Login</a>

</li>

<li class="nav-item" routerLinkActive="active">

<a class="nav-link" [routerLink]="['/signup']">Singup</a>

</li>

<li class="nav-item" routerLinkActive="active">

<a class="nav-link" [routerLink]="['/projects']">Projects</a>

</li>

<li class="nav-item" routerLinkActive="active">

<a class="nav-link" [routerLink]="['/contact']">Contact</a>

</li>

<li class="nav-item">

<a class="nav-link" href="" target="\_blank">Resume</a>

</li>

</ul>

</div>

</nav>

**Nav.component.ts**

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

@Component({

selector: 'app-nav',

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

styleUrls: ['./nav.component.scss']

})

export class NavComponent implements OnInit {

isCollapsed = true;

@HostListener('document:click', ['$event']) clickedOutside($event) {

this. collapseNavbar(); // Hide navigation menu on outside click

// console.log('CLICKED OUTSIDE');

}

constructor() { }

ngOnInit() {

}

collapseNavbar() {

if (this.isCollapsed === false) {

this.isCollapsed = true;

}

}

clickedInside($event: Event) {

$event.preventDefault();

$event.stopPropagation(); // <- stop propagation on lower layers

// console.log('CLICKED INSIDE, MENU WON\'T HIDE');

}

}

**Nav.component.css**

.navbar {

padding-left: 25px;

padding-right: 25px;

}

.navbar-nav {

float: right;

}

.show > .navbar-nav {

float: none;

}

.nav-item:hover {

transform: scale(1.2, 1.2);

transition-duration: 500ms;

transition-timing-function: ease-out;

}

.nav-item {

transform: scale(1, 1);

transition-duration: 500ms;

transition-timing-function: ease-out;

}

.navbar-brand {

text-transform: uppercase;

letter-spacing: 5px;

font-size: 1.2em;

text-decoration: none;

color: var(--offWhite);

transform: scale(1, 1);

transition-duration: 500ms;

transition-timing-function: ease-out;

}

.show {

position: absolute;

background-color: #343a40!important;

top: 62px;

right: 5px;

text-align: center;

width: 110px !important;

z-index: 1;

}

.navbar-toggler:hover {

border: 2px solid #007bff;

}

.navbar-brand:hover {

transform: scale(1.1, 1.1);

transition-duration: 500ms;

transition-timing-function: ease-out;

}

**Signup.component.html**

<div>

<div class="wrapper fadeInDown">

<div id="formContent">

<div class="fadeIn first">

<h2 class='sign'>Sign In</h2>

<p class="error">{{error}}</p>

</div>

<form>

<input type="text" id="login" class="fadeIn second" name="email"

placeholder="Email"

[(ngModel)]="user.uemail"/>

<br>

<br>

<input tupe="number" id="mobile" name="Moblie Number" placeholder="ph no"/>

<br>

<br>

<input type="password" [(ngModel)]="user.upassword"

id="password" class="fadeIn third" name="password"

placeholder="Password"/>

<br>

<br>

<input type="password" id="login2" class="fadeIn foutn" name="confirm password" placeholder="confirm password"/>

<br>

<br>

<input type="submit" (click)="login()"

class="fadeIn fourth" value="Log In" />

</form>

</div>

</div>

</div>

**Signup.comopenent.ts**

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

@Component({

selector: 'app-signup',

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

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

})

export class SignupComponent implements OnInit {

constructor() { }

ngOnInit() {

}

}

**Thankypou.component.html**

<div class="thankYou">

<h2>Thank you for your submission!</h2>

<a [routerLink]="['/home']" class="button btn">Back to Site</a>

</div>

**Thankyou.component.ts**

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

import { AlertifyService } from '../\_services/alertify.service';

@Component({

selector: 'app-thank-you',

templateUrl: './thank-you.component.html',

styleUrls: ['./thank-you.component.scss']

})

export class ThankYouComponent implements OnInit {

constructor(private alertify: AlertifyService) { }

ngOnInit() {

// this.alertify.success('Your Submission Has Been Sent!');

}

}

**Thankyou.component.css**

:host {

display: grid;

place-items: center;

div{

text-align: center;

a {

margin-top: 20px;

}

}

}

**App.component.html**

<app-nav></app-nav>

<main id="background">

<router-outlet></router-outlet>

</main>

<footer></footer>

**App.component.ts**

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

@Component({

selector: 'app-root',

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

styleUrls: ['./app.component.scss']

})

export class AppComponent {

}

**App.moudle.ts**

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

import { BrowserAnimationsModule } from "@angular/platform-browser/animations";

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

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

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

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

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

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

import { AppRoutes } from "./app.routing";

import { NavComponent } from "./nav/nav.component";

import { FontAwesomeModule } from "@fortawesome/angular-fontawesome";

import { library } from "@fortawesome/fontawesome-svg-core";

import { faGithub, faLinkedin } from "@fortawesome/free-brands-svg-icons";

import { faEnvelope } from "@fortawesome/free-regular-svg-icons";

import { CollapseModule } from "ngx-bootstrap/collapse";

import { ContactComponent } from "./contact/contact.component";

import { ProjectService } from "./\_services/project.service";

import { AlertifyService } from "./\_services/alertify.service";

import { ThankYouComponent } from "./thank-you/thank-you.component";

import { PreventUnsavedChanges } from "./\_gaurds/prevent-unsaved-changes.guard";

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

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

@NgModule({

declarations: [

AppComponent,

HomeComponent,

NavComponent,

ContactComponent,

ThankYouComponent,

LoginComponent,

SignupComponent,

],

imports: [

BrowserModule,

AppRoutingModule,

AppRoutes,

HttpClientModule,

FormsModule,

ReactiveFormsModule,

FontAwesomeModule,

BrowserAnimationsModule,

CollapseModule.forRoot(),

],

providers: [AlertifyService, ProjectService, PreventUnsavedChanges],

bootstrap: [AppComponent],

})

export class AppModule {

constructor() {

library.add(faGithub, faLinkedin, faEnvelope);

}

}

**App.routing.ts**

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

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

import { ContactComponent } from "./contact/contact.component";

import { ThankYouComponent } from "./thank-you/thank-you.component";

import { PreventUnsavedChanges } from "./\_gaurds/prevent-unsaved-changes.guard";

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

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

const routes: Routes = [

{

path: "",

redirectTo: "login",

pathMatch: "full",

},

{

path: "login",

component: LoginComponent,

},

{

path: "signup",

component: SignupComponent,

},

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

{

path: "contact",

component: ContactComponent,

canDeactivate: [PreventUnsavedChanges],

},

{ path: "thankYou", component: ThankYouComponent },

{ path: "\*\*", redirectTo: "/home", pathMatch: "full" },

];

export const AppRoutes = RouterModule.forChild(routes);

**AppRouting.module**

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

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

const routes: Routes = [];

@NgModule({

imports: [RouterModule.forRoot(routes)],

exports: [RouterModule]

})

export class AppRoutingModule { }