**Form Validations**

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<title>Page Title</title>

</head>

<body style="padding:40px ">

<p style="text-align: center;font-size: 30px"> Hello <input type="text" placeholder="Your name" (input)="ontyping($event)"/>, Welcome to the <b>angCare!</b></p>

<p style="text-align: center;font-size: 20px">Hello {{name}}<br>I am Joe, your personal assistant! I will guide you further...</p>

<div style="text-align:left;padding: 20px">

<p> Click on Sign up to create your account with angCare: <button (click)="signup()" class="btn btn-primary">Sign up with {{title}}</button> : {{status}}</p>

<p> Click on Sign up to create your account with angCare: <button>Sign In </button></p>

</div>

<div >

<div class="jumbotron">

<div class="container">

<div class="row">

<div class="col-md-6 offset-md-3">

<h2>Angular 6 Reactive Form Validation</h2>

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

<div class="form-group">

<label>First Name</label>

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

<div \*ngIf="submitted && f.firstName.errors" class="invalid-feedback">

<div \*ngIf="f.firstName.errors.required">First Name is required</div>

</div>

</div>

<div class="form-group">

<label>Last Name</label>

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

<div \*ngIf="submitted && f.lastName.errors" class="invalid-feedback">

<div \*ngIf="f.lastName.errors.required">Last Name is required</div>

</div>

</div>

<div class="form-group">

<label>Email</label>

<input type="text" formControlName="email" class="form-control" [ngClass]="{ 'is-invalid': submitted && f.email.errors }" />

<div \*ngIf="submitted && f.email.errors" class="invalid-feedback">

<div \*ngIf="f.email.errors.required">Email is required</div>

<div \*ngIf="f.email.errors.email">Email must be a valid email address</div>

</div>

</div>

<div class="form-group">

<label>Password</label>

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

<div \*ngIf="submitted && f.password.errors" class="invalid-feedback">

<div \*ngIf="f.password.errors.required">Password is required</div>

<div \*ngIf="f.password.errors.minlength">Password must be at least 6 characters</div>

</div>

</div>

<div class="form-group">

<button [disabled]="loading" class="btn btn-primary">Register</button>

</div>

</form>

</div>

</div>

</div>

</div>

</div>

</body>

</html>

<router-outlet></router-outlet>

Step 3.6.3: Completing the functionality by implementing validators

Open the app.component.ts file and replace only the source code available in AppComponent class with the following source code:

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

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

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

@Component({

selector: 'app-root',

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

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

})

export class AppComponent {

constructor(private formBuilder: FormBuilder) { }

title = 'angCare';

status = 'You haven\'t signed up yet';

name = '';

submitted = false;

registerForm: FormGroup;

ontyping(event:Event) {

this.name = (<HTMLInputElement>event.target).value;

}

signup() {

this.status = 'Oops! We are working on it!';

}

ngOnInit() {

this.registerForm = this.formBuilder.group({

firstName: ['', Validators.required],

lastName: ['', Validators.required],

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

password: ['', [Validators.required, Validators.minLength(6)]]

});

}

get f() { return this.registerForm.controls; }

onSubmit() {

this.submitted = true;

// stop here if form is invalid

if (this.registerForm.invalid) {

return;

}

alert('Your request has been submitted for approval')

}

}

Open the app.module.ts file and add the source code mentioned below:

In import section of the file, add

import { ReactiveFormsModule} from '@angular/forms'

In imports: section of the file, add

ReactiveFormsModule

Use ctrl+s to save the changes and the Angular application will automatically load. You should be able to see the form available in your Angular application.

