5.6 Reactive Forms Validation



This section will guide you to:

* Create a sign-up page
* Write a back-end logic to validate user input
* Respond to the successful user input

This guide has four subsections, namely:

5.6.1 Configuring the Angular application

5.6.2 Building a front-end form to sign up

5.6.3 Completing the functionality by implementing validators

5.6.4 Pushing the code to Github repositories

**Step 5.6.1:** Configuring the Angular application

* Create new project to this exercise ng angular-forms
  + Routing 🡪yes
  + Styling 🡪CSS
* Open the IDE and import the application by following the steps mentioned below:
* Click on **File**  **Open Folder…** and provide the path where you have installed your Angular application.
* Click **View** 🡪 **Terminal** to open the terminal of the IDE.
* Use **ng serve** in the terminal to start the local server to run at port 4200.

**Step 5.6.2:** Building a front-end form to sign up

* Open the **app.component.html** available in the **app** folder of the Angular application. Replace all the code available in the file to the source code mentioned below:

<div class="jumbotron">

  <div class="container">

      <div class="row">

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

              <h2>Angular 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 class="btn btn-primary">Register</button>

                  </div>

              </form>

          </div>

      </div>

  </div>

</div>

**Step 5.6.3:** 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

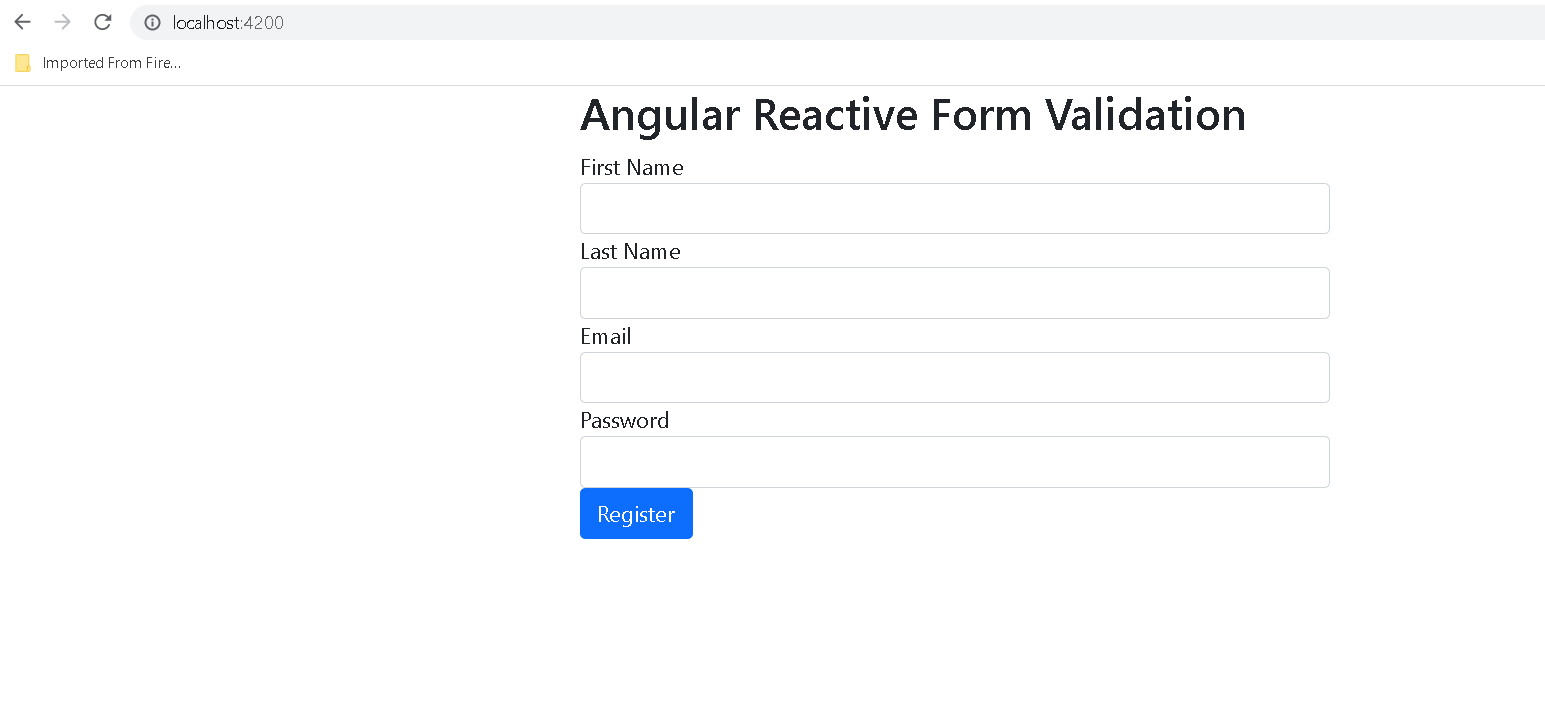
import { ReactiveFormsModule} from '@angular/forms'

* In **imports:** sectionof the file, add

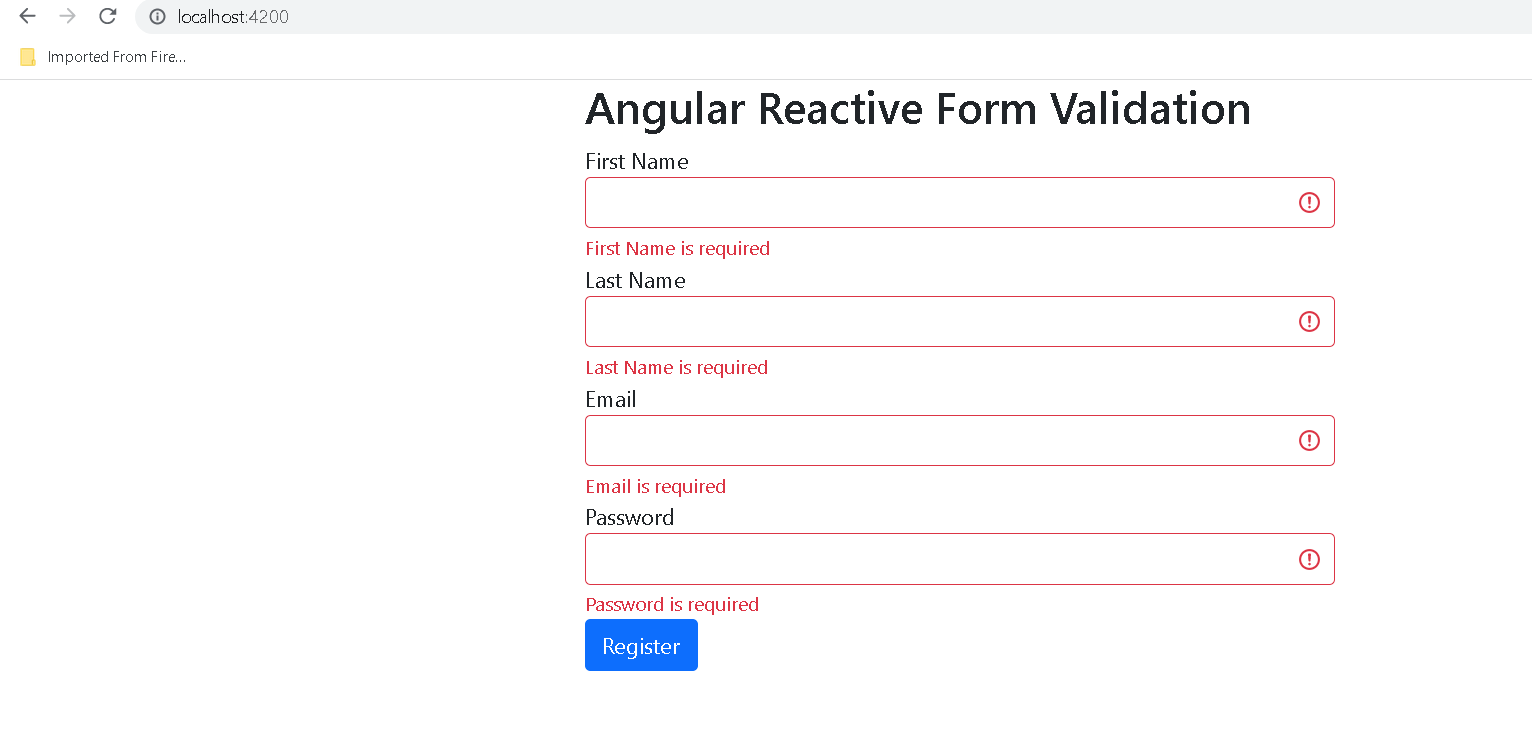
ReactiveFormsModule

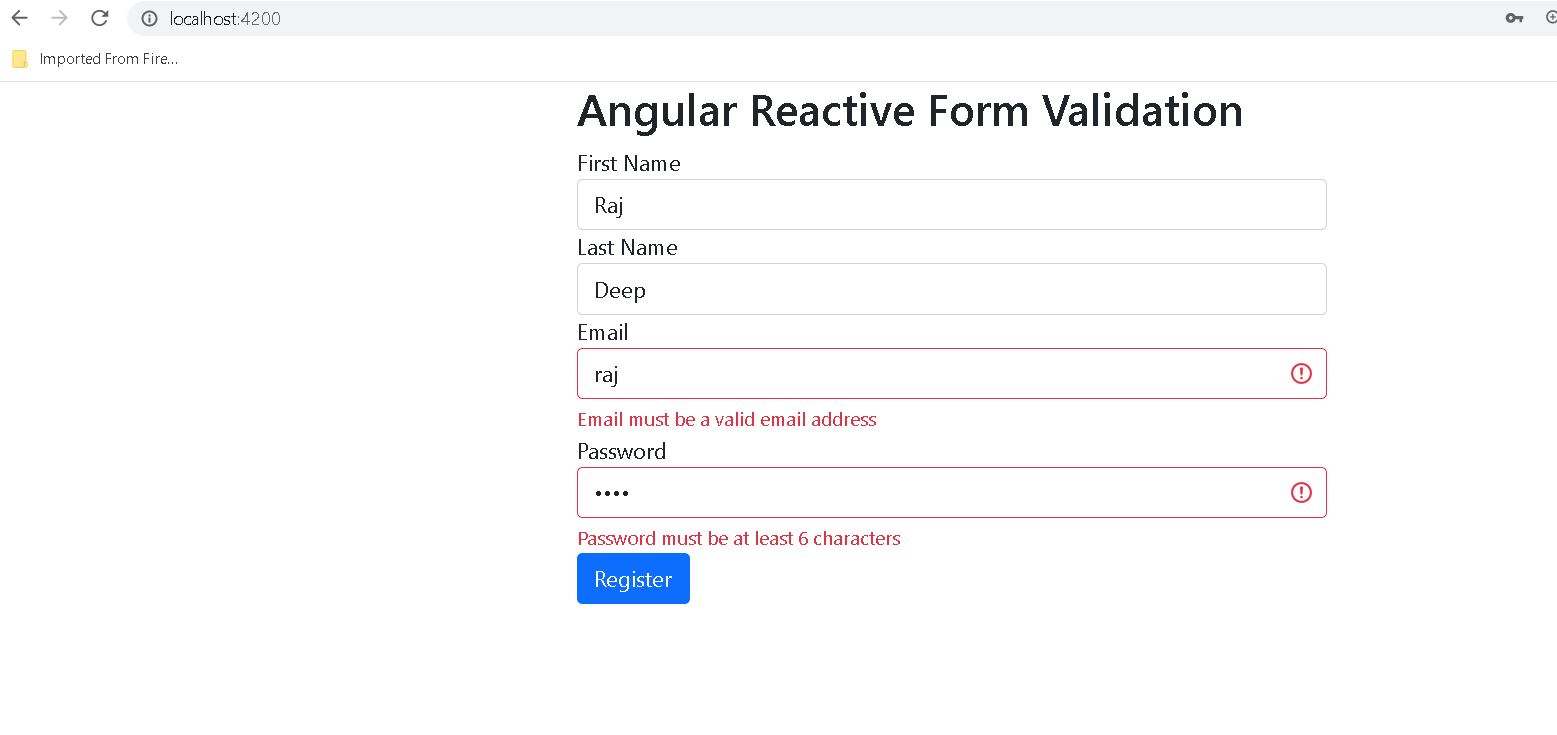
* Use **ctrl+s** to save the changes and the Angular application will automatically load. You should be able to see the form available in your Angular application.

Output as

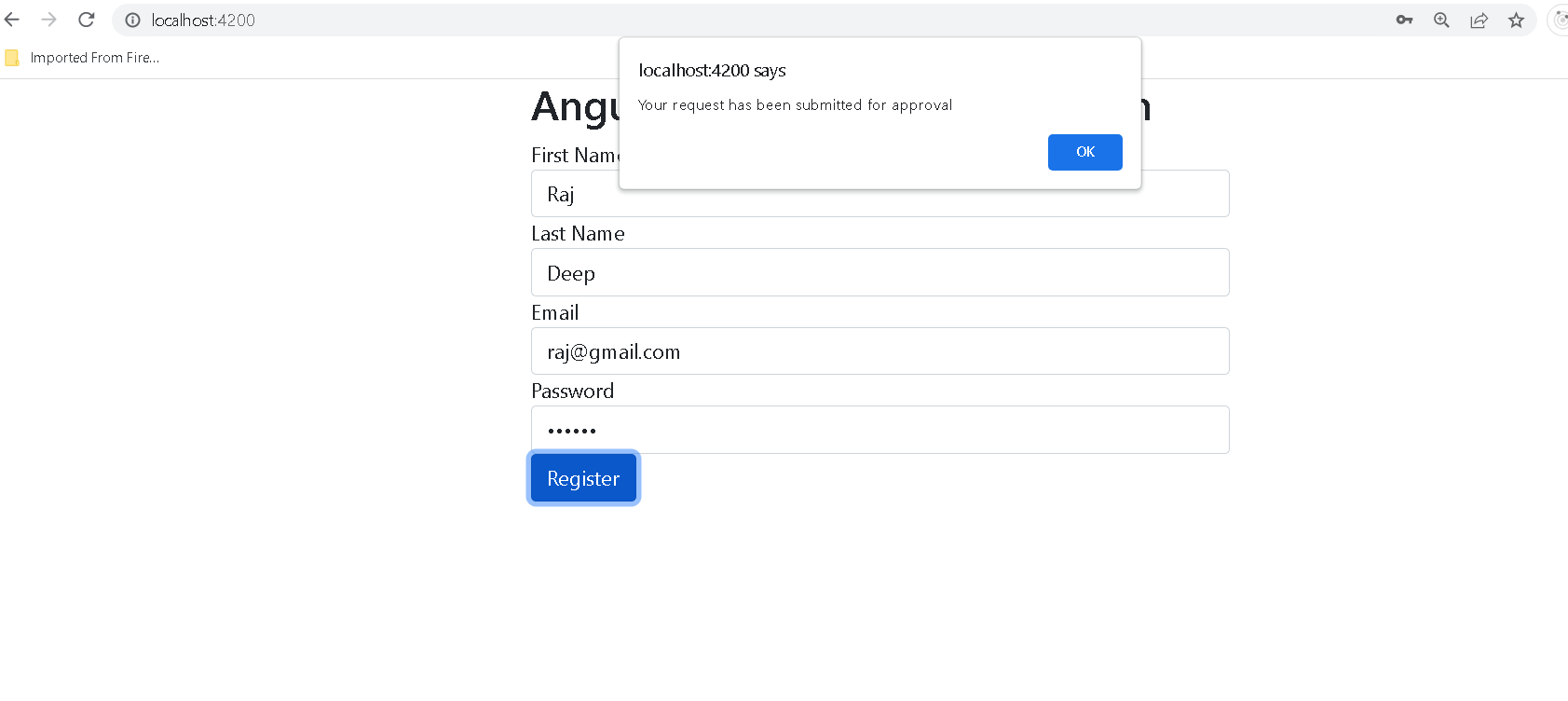


Without writing if we click on Register button





After filled all your details



**Step 5.6.4:** Pushing the code to your GitHub repositories

Open your command prompt and navigate to the folder where you have created your files

cd <folder path>

Initialize your repository using the following command:

git init

Add all the files to your git repository using the following command:

git add . 

Commit the changes using the following command:

git commit . -m “Changes have been committed.”

Push the files to the folder you initially created using the following command:

git push -u origin master