

# SOURCE CODE

## app.module.ts

```
import { BrowserModule } from '@angular/platform-browser';
import { NgModule } from '@angular/core';
import { RouterModule, Routes } from '@angular/router'
import { AppRoutingModule } from './app-routing.module';
import { AppComponent } from './app.component';
import { ReactiveFormsModule } from '@angular/forms'
import { SignupComponent } from './signup/signup.component';

const routes: Routes = [{
  path: '',
  component: AppComponent
},
{
  path: 'signup',
  component: SignupComponent
},
]

@NgModule({
  declarations: [
    AppComponent,
    SignupComponent ],
  imports: [
    BrowserModule,
    AppRoutingModule,
    ReactiveFormsModule,
    RouterModule.forRoot(routes)
  ],
  providers: [],
  bootstrap: [AppComponent]
})
export class AppModule { }
```

## app.component.ts

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

```

import { FormBuilder } from '@angular/forms';

@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.css']
})
export class AppComponent implements OnInit {
  constructor(private formBuilder: FormBuilder) {}

  title = 'angCare';
  status = "You haven't signed up yet";
  name = '';
  showRouterOutlet = false;

  ngOnInit() {
    // Hide the router-outlet initially
    this.showRouterOutlet = false;
  }

  onTyping(event: Event) {
    this.name = (event.target as HTMLInputElement).value;
  }

  signup() {
    this.status = '';
    // Show the router-outlet after the button is clicked
    this.showRouterOutlet = true;
  }
}

```

## app.component.html

```

<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <title>Page Title</title>
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css"
">
</head>
<body style="padding: 40px">

```

```

<p style="text-align: center; font-size: 20px">
  Welcome to the <b>ANGULAR!</b><br><br>
  Type Your Name <input type="text" placeholder="Your name"
(input)="onTyping($event)" />
</p>
<p style="text-align: center; font-size: 20px">
  Hello {{name}}
</p>
<div style="text-align: center; padding: 20px">
  <p>
    Click to Register
    <button (click)="signup()" class="btn btn-success"
[routerLink]='"/signup"'>Sign up with {{title}}</button> <br>
    {{status}}
  </p>
</div>
<router-outlet *ngIf="showRouterOutlet"></router-outlet>
</body>
</html>

```

## signup.component.ts

```

import { Component, OnInit } from '@angular/core';
import { FormBuilder, FormGroup, Validators } from '@angular/forms';
@Component({
  selector: 'app-signup',
  templateUrl: './signup.component.html',
  styleUrls: ['./signup.component.css']
})
export class SignupComponent implements OnInit {
  submitted = false;
  registerForm!: FormGroup; // Add '!' to indicate it will be initialized
  loading = false; // Add the 'loading' property
  constructor(private formBuilder: FormBuilder) { }
  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')
}
}

```

## signup.component.html

```

<div class="container">
  <div class="row">
    <div class="col-md-3"></div>
    <div class="col-md-6">
      <div class="jumbotron">

        <h2>REGISTRATION FORM</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>
        </form>
      </div>
    </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-
success">REGISTER</button>
        </div>
    </form>
</div>
</div>
</div>
</div>

```