

Angular Form:

Let's understand Angular form with Bootstrap.

1. Install bootstrap in our application. For doing so, fire a command. (Visit official web site for documentation) → <https://getbootstrap.com/>
npm install [bootstrap@5.2.3](#)

or we can also use CDN

```
<!-- CSS only -->
```

```
<link
```

```
href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/  
/dist/css/bootstrap.min.css" rel="stylesheet"
```

```
integrity="sha384-
```

```
rbsA2VBKQhggwzxH7pPCaAqO46MgnOM80zW1RW  
uH61DGLwZJEdK2Kadq2F9CUG65"
```

```
crossorigin="anonymous">
```

```
<!-- JavaScript Bundle with Popper -->
```

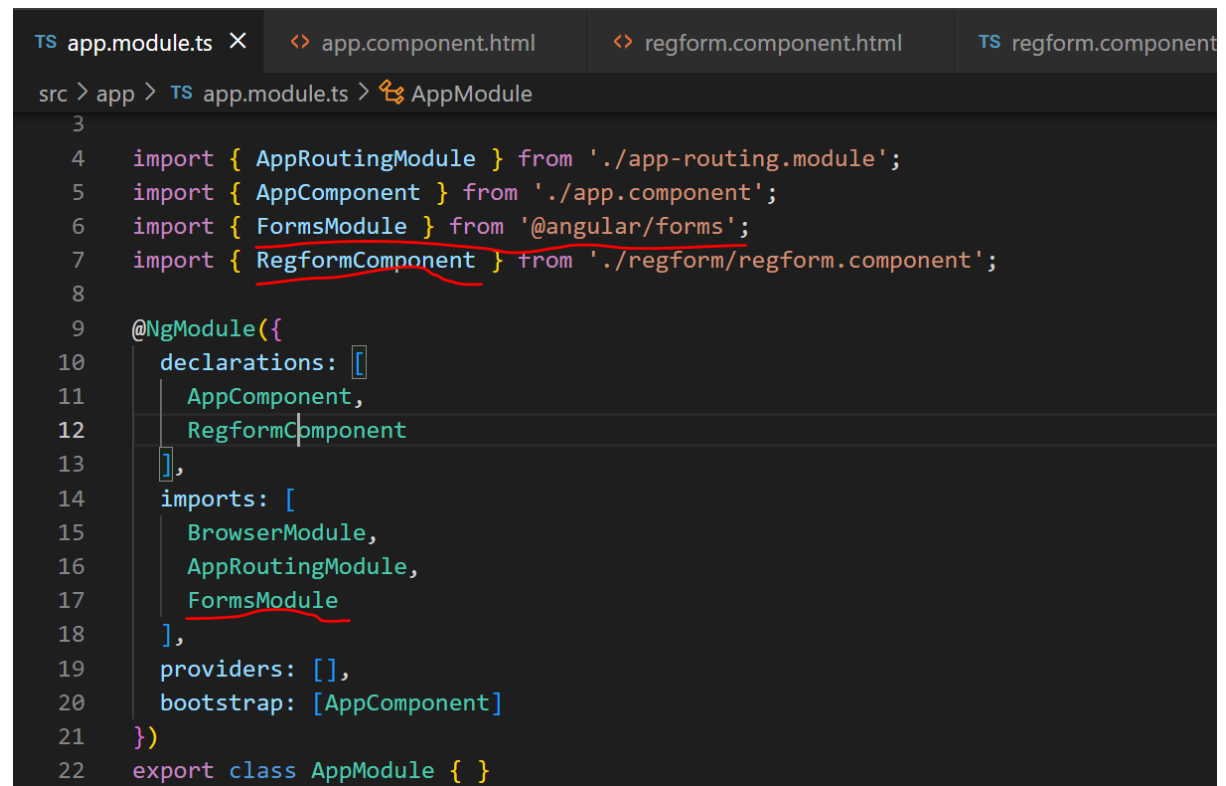
```
<script
```

```
src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/  
dist/js/bootstrap.bundle.min.js" integrity="sha384-  
kenU1KFdBle4zVF0s0G1M5b4hcpyD9F7jL+jjXkk+Q2  
h455rYXK/7HAuoJl+0I4"
```

```
crossorigin="anonymous"></script>
```

Step 2. Add one more component regform.

Step 3. Import formmodule in **app.module.ts**



```
TS app.module.ts X  <> app.component.html  <> regform.component.html  TS regform.component
src > app > TS app.module.ts > AppModule
3
4  import { AppRoutingModule } from './app-routing.module';
5  import { AppComponent } from './app.component';
6  import { FormsModule } from '@angular/forms';
7  import { RegformComponent } from './regform/regform.component';
8
9  @NgModule({
10   declarations: [
11     AppComponent,
12     RegformComponent
13   ],
14   imports: [
15     BrowserModule,
16     AppRoutingModule,
17     FormsModule
18   ],
19   providers: [],
20   bootstrap: [AppComponent]
21 })
22 export class AppModule { }
```

Step 4. Open getbootstrap.com. Click on docs on top. Scroll down and copy paste bootstrap template.

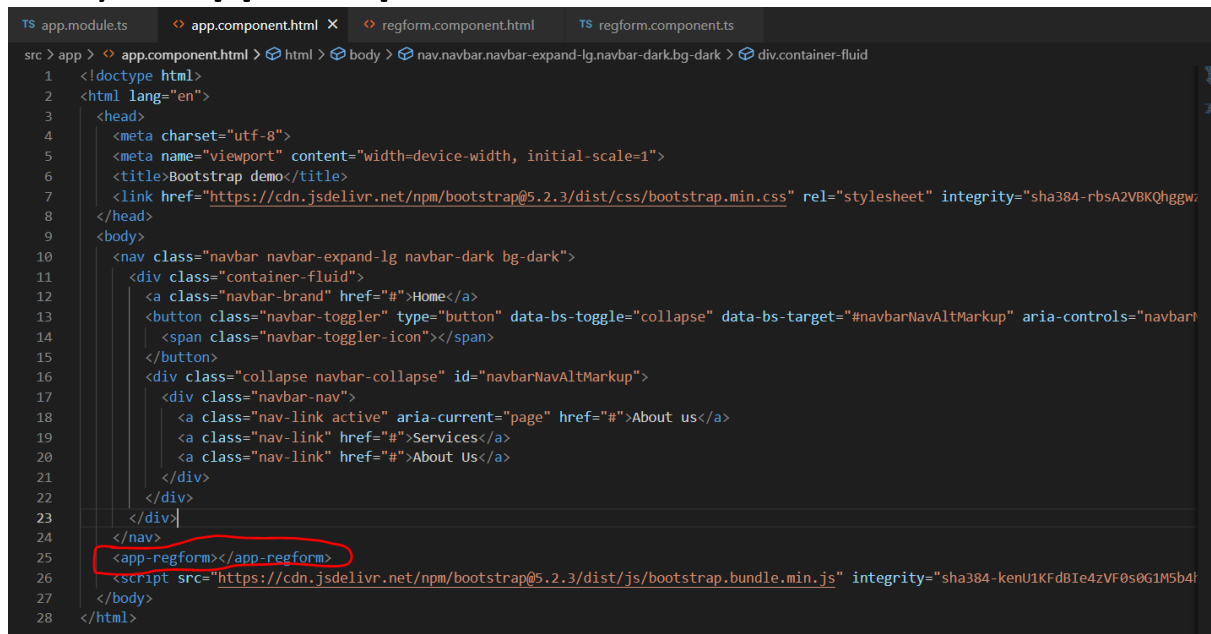
```
<!doctype html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Bootstrap demo</title>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-
rbsA2VBKQhggwzxH7pPCaAqO46MgnOM80zW1RWuH61DGLwZJEdK2Kadq2F9CUG65"
crossorigin="anonymous">
  </head>
  <body>
    <h1>Hello, world!</h1>
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/js/bootstrap.bundle.min.js"
integrity="sha384-
kenU1KFdBLE4zVF0s0G1M5b4hcxpyD9F7jL+jjXkk+Q2h455rYXK/7HAuoJl+0I4"
crossorigin="anonymous"></script>
  </body>
</html>
```

Remove `<h1>Hello, world!</h1>` and make template empty.

Step 5. Copy any required navbar and paste in app.component.html

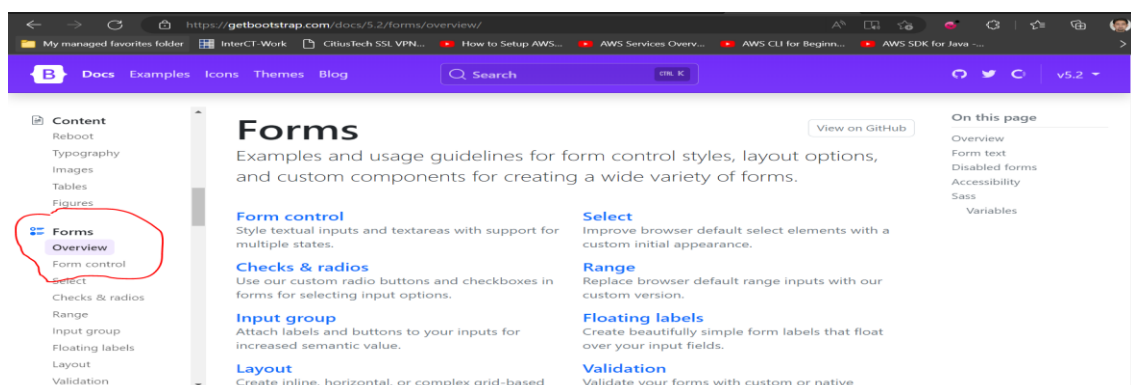
```
<nav class="navbar navbar-dark bg-dark">
  <div class="container-fluid">
    <a class="navbar-brand">Navbar</a>
    <form class="d-flex" role="search">
      <input class="form-control me-2" type="search" placeholder="Search" aria-label="Search">
      <button class="btn btn-outline-success" type="submit">Search</button>
    </form>
  </div>
</nav>
```

Step 6. Now, paste selector of <app-regform> in body of app.component.html



```
src > app > app.component.html > html > body > nav.navbar.navbar-expand-lg.navbar-dark.bg-dark > div.container-fluid
1 <!doctype html>
2 <html lang="en">
3   <head>
4     <meta charset="utf-8">
5     <meta name="viewport" content="width=device-width, initial-scale=1">
6     <title>Bootstrap demo</title>
7     <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-rbsA2VBKQhggw/
8   </head>
9   <body>
10    <nav class="navbar navbar-expand-lg navbar-dark bg-dark">
11      <div class="container-fluid">
12        <a class="navbar-brand" href="#">Home</a>
13        <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarNavAltMarkup" aria-controls="navbar
14          <span class="navbar-toggler-icon"></span>
15        </button>
16        <div class="collapse navbar-collapse" id="navbarNavAltMarkup">
17          <div class="navbar-nav">
18            <a class="nav-link active" aria-current="page" href="#">About us</a>
19            <a class="nav-link" href="#">Services</a>
20            <a class="nav-link" href="#">About Us</a>
21          </div>
22        </div>
23      </div>
24    </nav>
25    <app-regform></app-regform>
26    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/js/bootstrap.bundle.min.js" integrity="sha384-kenU1KfD8Ie4zVF0s0G1M5b4I
27  </body>
28 </html>
```

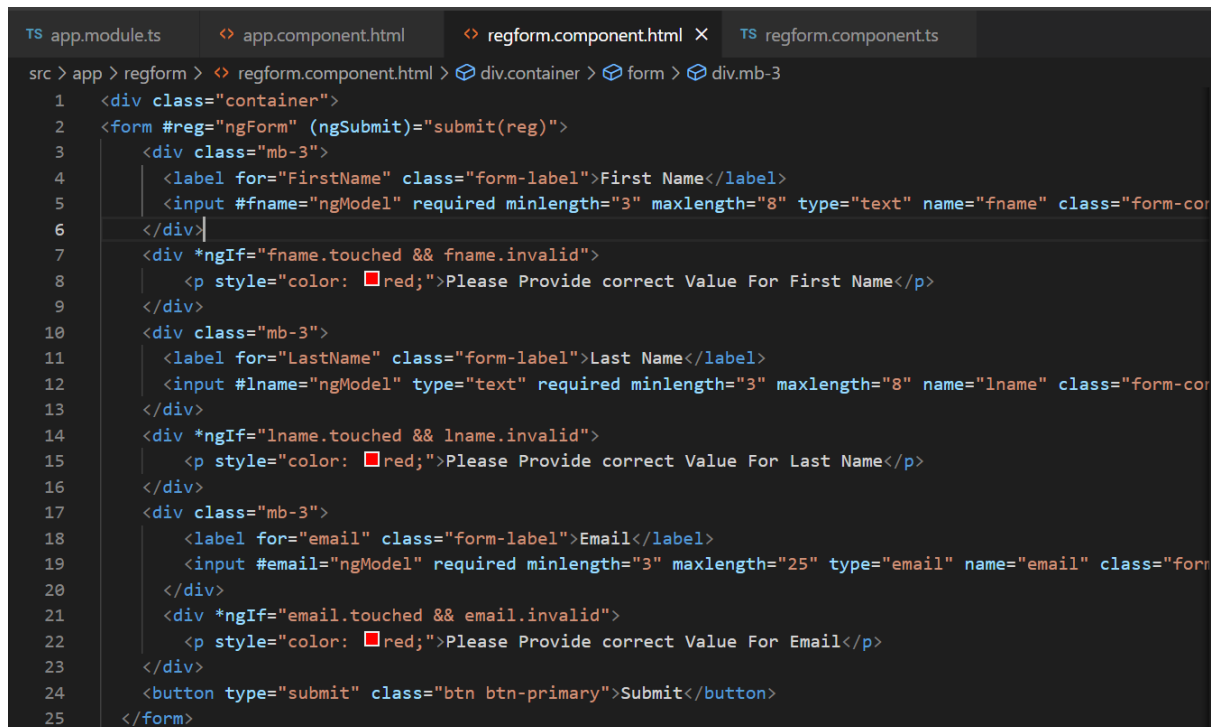
Step 7. Open regform.component.html. Now, open form module of bootstrap.



Scroll down and copy first template of form.

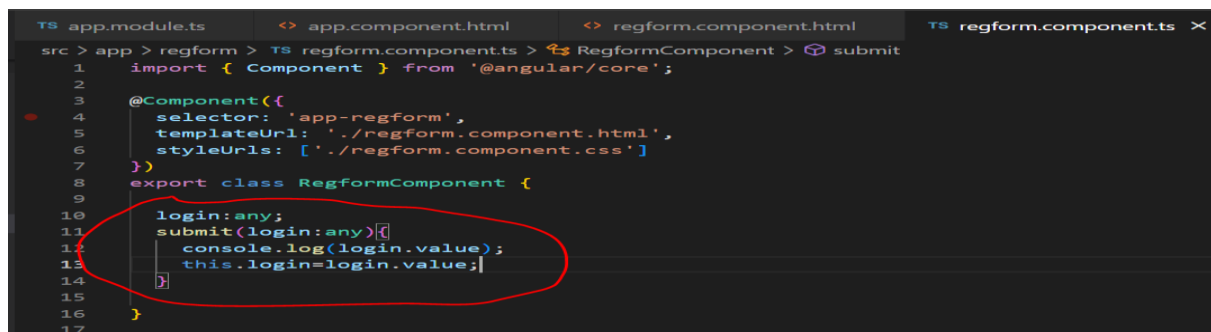
```
<form>
<div class="mb-3">
  <label for="exampleInputEmail1" class="form-label">Email address</label>
  <input type="email" class="form-control" id="exampleInputEmail1" aria-describedby="emailHelp">
  <div id="emailHelp" class="form-text">We'll never share your email with anyone else.</div>
</div>
<div class="mb-3">
  <label for="exampleInputPassword1" class="form-label">Password</label>
  <input type="password" class="form-control" id="exampleInputPassword1">
</div>
<div class="mb-3 form-check">
  <input type="checkbox" class="form-check-input" id="exampleCheck1">
  <label class="form-check-label" for="exampleCheck1">Check me out</label>
</div>
<button type="submit" class="btn btn-primary">Submit</button>
</form>
```

Do given below changes.



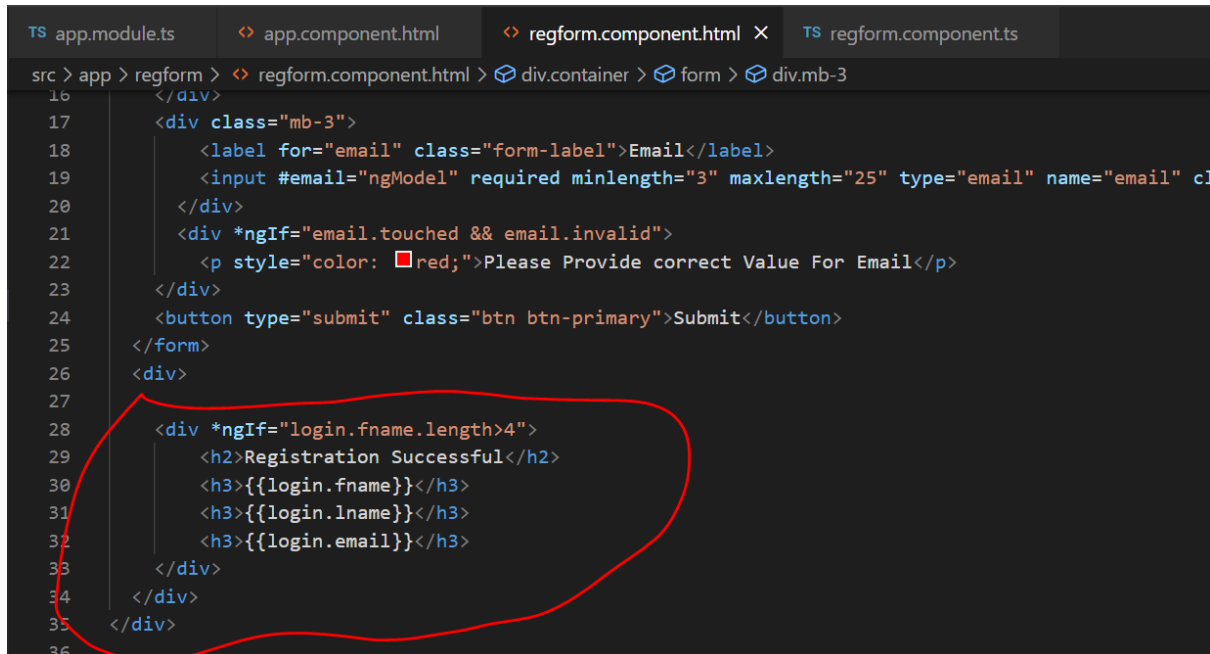
```
src > app > regform > regform.component.html > div.container > form > div.mb-3
1  <div class="container">
2  <form #reg="ngForm" (ngSubmit)="submit(reg)">
3    <div class="mb-3">
4      <label for="FirstName" class="form-label">First Name</label>
5      <input #fname="ngModel" required minlength="3" maxlength="8" type="text" name="fname" class="form-control">
6    </div>
7    <div *ngIf="fname.touched && fname.invalid">
8      <p style="color: red;">Please Provide correct Value For First Name</p>
9    </div>
10   <div class="mb-3">
11     <label for="LastName" class="form-label">Last Name</label>
12     <input #lname="ngModel" type="text" required minlength="3" maxlength="8" name="lname" class="form-control">
13   </div>
14   <div *ngIf="lname.touched && lname.invalid">
15     <p style="color: red;">Please Provide correct Value For Last Name</p>
16   </div>
17   <div class="mb-3">
18     <label for="email" class="form-label">Email</label>
19     <input #email="ngModel" required minlength="3" maxlength="25" type="email" name="email" class="form-control">
20   </div>
21   <div *ngIf="email.touched && email.invalid">
22     <p style="color: red;">Please Provide correct Value For Email</p>
23   </div>
24   <button type="submit" class="btn btn-primary">Submit</button>
25 </form>
```

Step 8. Open regform.component.ts file.



```
src > app > regform > regform.component.ts > RegformComponent > submit
1  import { Component } from '@angular/core';
2
3  @Component({
4    selector: 'app-regform',
5    templateUrl: './regform.component.html',
6    styleUrls: ['./regform.component.css']
7  })
8  export class RegformComponent {
9
10   login:any;
11   submit(login:any){
12     console.log(login.value);
13     this.login=login.value;
14   }
15
16 }
17
```

Step 9. Now, add given below code at the bottom of **regform.component.html** to display Data coming from form.



```
TS app.module.ts  app.component.html  regform.component.html X  TS regform.component.ts
src > app > regform > regform.component.html > div.container > form > div.mb-3
16   </div>
17   <div class="mb-3">
18     <label for="email" class="form-label">Email</label>
19     <input #email="ngModel" required minlength="3" maxlength="25" type="email" name="email" c
20   </div>
21   <div *ngIf="email.touched && email.invalid">
22     <p style="color: red;">Please Provide correct Value For Email</p>
23   </div>
24   <button type="submit" class="btn btn-primary">Submit</button>
25 </form>
26 <div>
27
28   <div *ngIf="login.fname.length>4">
29     <h2>Registration Successful</h2>
30     <h3>{{login.fname}}</h3>
31     <h3>{{login.lname}}</h3>
32     <h3>{{login.email}}</h3>
33   </div>
34 </div>
35 </div>
36
```

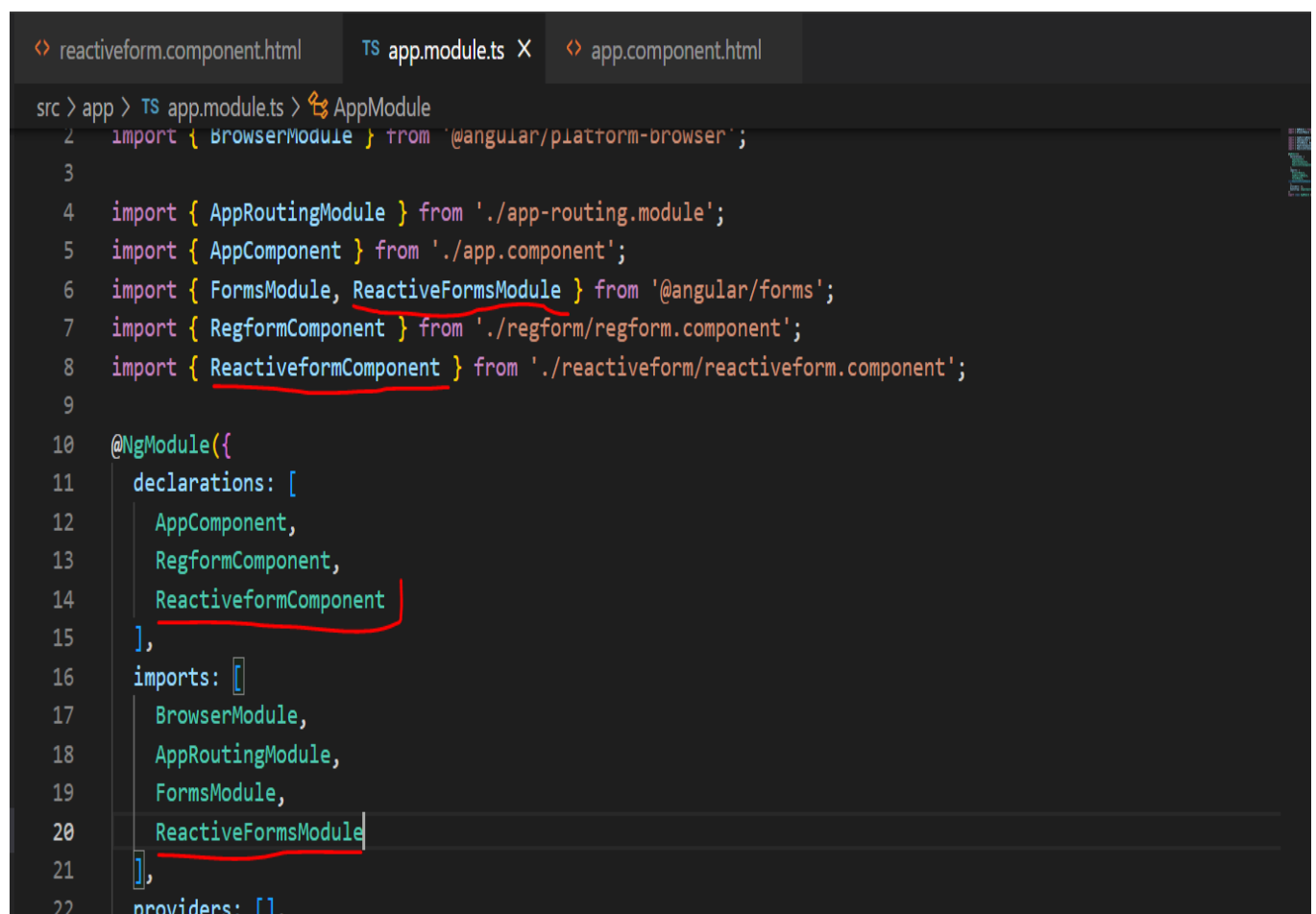
Reactive Form:

Here, we will see how to

1. make a simple form.
2. Import module and form module
3. Connect form control with form
4. Get value from forms
5. Set Default values

Step 1. Create new component **ng g c reactiveform**

Step 2. Open app.module.ts and import ReactiveFormsModule.



```
reactiveform.component.html TS app.module.ts X app.component.html
src > app > TS app.module.ts > AppModule
2 import { BrowserModule } from '@angular/platform-browser';
3
4 import { AppRoutingModule } from './app-routing.module';
5 import { AppComponent } from './app.component';
6 import { FormsModule, ReactiveFormsModule } from '@angular/forms';
7 import { RegformComponent } from './regform/regform.component';
8 import { ReactiveformComponent } from './reactiveform/reactiveform.component';
9
10 @NgModule({
11   declarations: [
12     AppComponent,
13     RegformComponent,
14     ReactiveformComponent
15   ],
16   imports: [
17     BrowserModule,
18     AppRoutingModule,
19     FormsModule,
20     ReactiveFormsModule
21   ],
22   providers: [],
```

Step 3. Open reactiveform.component.html file.

And add given below code into it.

```
<> reactiveform.component.html X TS app.module.ts <> app.component.html
src > app > reactiveform > <> reactiveform.component.html > div.container > form > br
1  <div class="container">
2    <h2>Post Your Comments</h2>
3    <form>
4      <label for="Username">User Name</label>
5      <br/><br/>
6      <input type="text" placeholder="Enter Name" name="user">
7      <br/><br/>
8      <label for="comment">Comment</label>
9      <br/><br/>
10     <input type="text" placeholder="Your Post" name="post">
11     <br/><br/>
12     <button class="btn btn-primary">Post</button>
13   </form>
14
15 </div>
```

Step 4. Open reactiveform.component.ts file and import two classes from @angular/forms

```
<> reactiveform.component.html TS reactiveform.component.ts X TS app.component.ts TS app.module.ts <>
src > app > reactiveform > TS reactiveform.component.ts > ReactiveFormComponent
1  import { Component } from '@angular/core';
2  import { FormGroup, FormControl } from '@angular/forms';
3
4  @Component({
5    selector: 'app-reactiveform',
6    templateUrl: './reactiveform.component.html',
7    styleUrls: ['./reactiveform.component.css']
8  })
9  export class ReactiveFormComponent {
10
11  }
12
13
```

Step 5. Open **reactiveform.component.html** and provide name by using **[formGroup]="postform"**

```
<> reactiveform.component.html X TS reactiveform.component.ts TS app.component.ts
src > app > reactiveform > <> reactiveform.component.html > div.container > form
1  <div class="container">
2    <h2>Post Your Comments</h2>
3    <form [formGroup]="postform">
4      <label for="Username">User Name</label>
5      <br/><br/>
6      <input type="text" placeholder="Enter Name" name="user">
7      <br/><br/>
8      <label for="comment">Comment</label>
9      <br/><br/>
10     <input type="text" placeholder="Your Post" name="post">
11     <br/><br/>
12     <button class="btn btn-primary">Post</button>
13   </form>
14
15 </div>
```

Step 6. Open **reactiveform.component.ts** file.

Now, use same form name and collect two properties of html in ts file.

```
<> reactiveform.component.html X TS reactiveform.component.ts X TS app.component.ts TS app.module.ts
src > app > reactiveform > TS reactiveform.component.ts > ReactiveformComponent > postform > post
1  import { Component } from '@angular/core';
2  import { FormGroup, FormControl } from '@angular/forms';
3
4  @Component({
5    selector: 'app-reactiveform',
6    templateUrl: './reactiveform.component.html',
7    styleUrls: ['./reactiveform.component.css']
8  })
9  export class ReactiveformComponent {
10    postform = new FormGroup({
11      user: new FormControl(''),
12      post: new FormControl('')
13    });
14
15
16  }
17
```

Now, this two elements user and password needs to specify in reactiveform.component.html as in given picture.


```
reactiveform.component.html X TS reactiveform.component.ts TS app.component.ts TS app.module.ts <> app.co

src > app > reactiveform > <> reactiveform.component.html > div.container
1 <div class="container">
2   <h2>Post Your Comments</h2>
3   <form [formGroup]="postform">
4     <label for="Username">User Name</label>
5     <br/><br/>
6     <input type="text" placeholder="Enter Name" name="user" FormControlName="user">
7     <br/><br/>
8     <label for="comment">Comment</label>
9     <br/><br/>
10    <input type="text" placeholder="Your Post" name="post" FormControlName="post">
11    <br/><br/>
12    <button class="btn btn-primary">Post</button>
13  </form>
14
15 </div>
```

Step 7. Add event on form. For this add given below code.

```
reactiveform.component.html X TS reactiveform.component.ts X TS app.component.ts TS app.module.ts <> app.component.ts

src > app > reactiveform > <> reactiveform.component.html > div.container > form
1 <div class="container">
2   <h2>Post Your Comments</h2>
3   <form [formGroup]="postform" (ngSubmit)="postComment()">
4     <label for="Username">User Name</label>
5     <br/><br/>
6     <input type="text" placeholder="Enter Name" name="user" FormControlName="user">
7     <br/><br/>
8     <label for="comment">Comment</label>
9     <br/><br/>
10    <input type="text" placeholder="Your Post" name="post" FormControlName="post">
11    <br/><br/>
12    <button class="btn btn-primary">Post</button>
13  </form>
14
15 </div>
```

Step 8. Open **reactive.component.ts** and given below code

```
<> reactiveform.component.html X TS reactiveform.component.ts X TS app.component.ts TS app.module.ts

src > app > reactiveform > TS reactiveform.component.ts > ReactiveformComponent > postComment
1 import { Component } from '@angular/core';
2 import { FormGroup, FormControl } from '@angular/forms';
3
4 @Component({
5   selector: 'app-reactiveform',
6   templateUrl: './reactiveform.component.html',
7   styleUrls: ['./reactiveform.component.css']
8 })
9 export class ReactiveFormComponent {
10   postform = new FormGroup({
11     user: new FormControl(''),
12     post: new FormControl('')
13   });
14   postComment(){
15     console.log(this.postform.value)
16   }
17
18 }
19
20
```

We can add default value to element of our reactive form

Final reactiveform.component.html file:

```
reactiveform.component.html X TS reactiveform.component.ts TS app.component.ts TS app.module.ts <> app
src > app > reactiveform > <> reactiveform.component.html > div.container > form
1 <div class="container">
2   <h2>Post Your Comments</h2>
3   <form [formGroup]="postform" (ngSubmit)="postComment()">
4     <label for="Username">User Name</label>
5     <br/><br/>
6     <input type="text" placeholder="Enter Name" name="user" formControlName="user">
7     <br/><br/>
8     <label for="comment">Comment</label>
9     <br/><br/>
10    <input type="text" placeholder="Your Post" name="post" formControlName="post">
11    <br/><br/>
12    <button class="btn btn-primary">Post</button>
13  </form>
14
15 </div>
```

Validation

We have seen how to add validation to our form. In Reactive form we can also add validation.

We will use same form and start implementing validation.

Step 1. Open reactform.component.ts file and import validators class.

```
<> reactiveform.component.html TS reactiveform.component.ts X TS app.component.ts TS app.module.ts <> ap
src > app > reactiveform > TS reactiveform.component.ts > ...
1 import { Component } from '@angular/core';
2 import { FormGroup, FormControl, Validators } from '@angular/forms'
3
4 @Component({
5   selector: 'app-reactiveform',
6   templateUrl: './reactiveform.component.html',
7   styleUrls: ['./reactiveform.component.css']
8 })
9 export class ReactiveformComponent {
10   postform = new FormGroup({
11     user: new FormControl(''),
12     post: new FormControl('')
13   })
14   postComment(){
15     console.log(this.postform.value)
16   }
17
18
19 }
```

Let's consider our requirement that user text box should have some value so we will add validation on it.

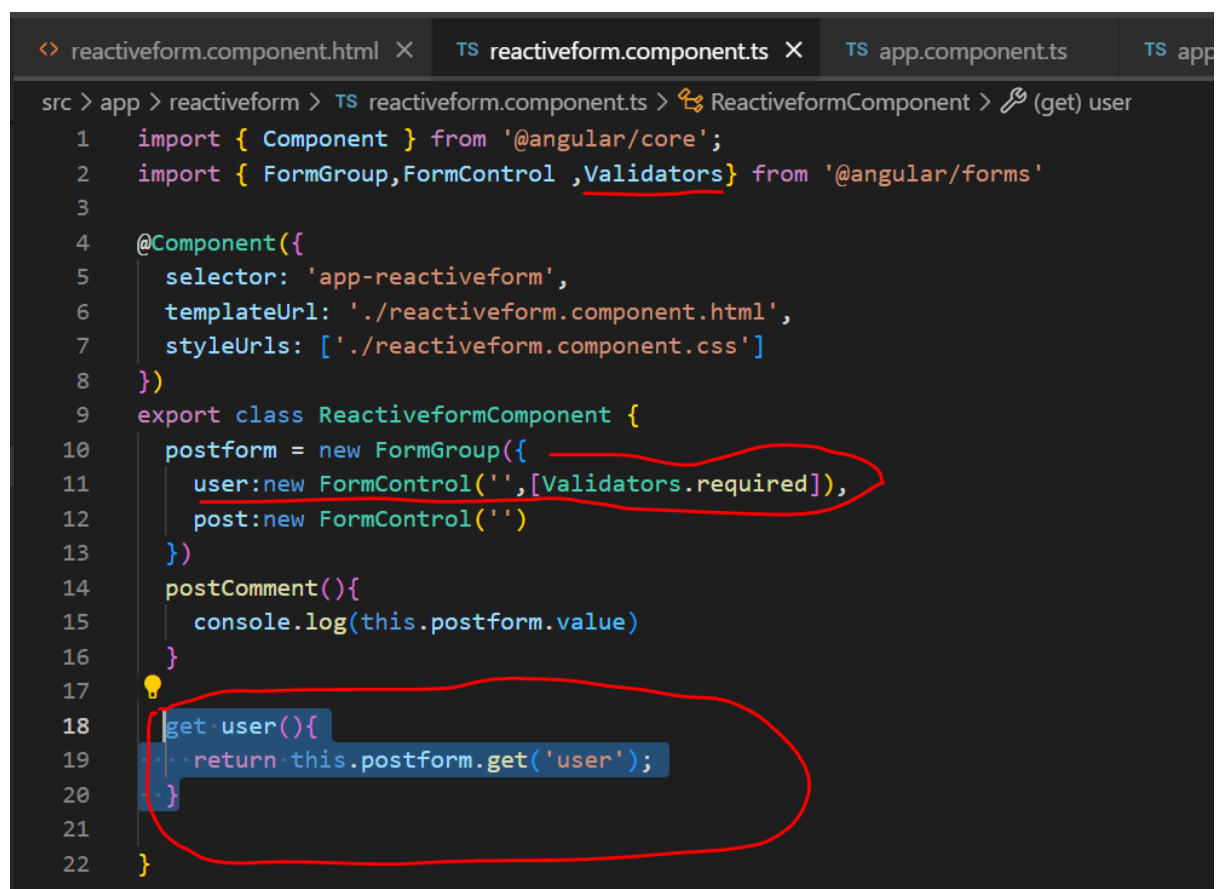
a. Use validators class and apply required property

```
postform = new FormGroup({
  user: new FormControl('', [Validators.required]),
  post: new FormControl('')
})
```

b. Now to get this value in html for displaying message. We will create one getter for user.

```
get user(){
  return this.postform.get('user');
}
```

Finally, our reactive.component.ts will look as given below:



```
<> reactiveform.component.html X TS reactiveform.component.ts X TS app.component.ts TS app
src > app > reactiveform > TS reactiveform.component.ts > ReactiveformComponent > (get) user
1  import { Component } from '@angular/core';
2  import { FormGroup, FormControl, Validators } from '@angular/forms';
3
4  @Component({
5    selector: 'app-reactiveform',
6    templateUrl: './reactiveform.component.html',
7    styleUrls: ['./reactiveform.component.css']
8  })
9  export class ReactiveformComponent {
10    postform = new FormGroup({
11      user: new FormControl('', [Validators.required]),
12      post: new FormControl('')
13    })
14    postComment(){
15      console.log(this.postform.value)
16    }
17
18    get user(){
19      return this.postform.get('user');
20    }
21
22  }
```

Step 2. Now, open reactive.component.ts file and add given below code to display validation.

```
reactiveform.component.html X TS reactiveform.components.ts TS app.components.ts TS app.module.ts app.component.html
src > app > reactiveform > reactiveform.component.html > div.container > form > br
1 <div class="container">
2   <h2>Post Your Comments</h2>
3   <form [formGroup]="postform" (ngSubmit)="postComment()">
4     <label for="Username">User Name</label>
5     <br/><br/>
6     <input type="text" placeholder="Enter Name" name="user" formControlName="user">
7     <br/><br/>
8     <span style="color: red; font-size: large;" *ngIf="user && user.invalid && user.touched">This Field is invalid</span>
9     <br/><br/>
10    <label for="comment">Comment</label>
11    <br/><br/>
12    <input type="text" placeholder="Your Post" name="post" formControlName="post">
13    <br/><br/>
14    <button class="btn btn-primary">Post</button>
15  </form>
16
17 </div>
```

So, here we have seen that how we can implement Reactive form using codes. Therefor reactive forms also called as code driven form.

Additional Info:

If you want to provide some more validation then we can add those multiple validators with Validators class.

Final reactiveform.component.ts

```
reactiveform.component.html X TS reactiveform.components.ts X TS app.components.ts TS app.module.ts app.component.html
src > app > reactiveform > TS reactiveform.components.ts > ReactiveformComponent > (get) post
1 import { Component } from '@angular/core';
2 import { FormGroup, FormControl, Validators } from '@angular/forms';
3
4 @Component({
5   selector: 'app-reactiveform',
6   templateUrl: './reactiveform.component.html',
7   styleUrls: ['./reactiveform.component.css']
8 })
9 export class ReactiveformComponent {
10   postform = new FormGroup({
11     user: new FormControl('', [Validators.required, Validators.email]),
12     post: new FormControl('', [Validators.required, Validators.minLength(3), Validators.maxLength(8)])
13   });
14   postComment(){
15     console.log(this.postform.value)
16   }
17
18   get user(){
19     return this.postform.get('user');
20   }
21
22   get post(){
23     return this.postform.get('post');
24   }
25 }
26
```

Final reactive.component.html

```
reactiveform.component.html x TS reactiveform.component.ts TS app.component.ts TS app.module.ts app.component.html
src > app > reactiveform > reactiveform.component.html > div.container > form > br
1 <div class="container">
2   <h2>Post Your Comments</h2>
3   <form [formGroup]="postform" (ngSubmit)="postComment()">
4     <label for="Username">User Name</label>
5     <br/><br/>
6     <input type="text" placeholder="Enter Name" name="user" formControlName="user">
7     <br/><br/>
8     <span style="color: red; font-size: large;" *ngIf="user && user.invalid && user.touched">This Field is invalid</span>
9     <br/><br/>
10    <label for="comment">Comment</label>
11    <br/><br/>
12    <input type="text" placeholder="Your Post" name="post" formControlName="post">
13    <br/><br/>
14    <span style="color: red; font-size: large;" *ngIf="post && post.invalid && post.touched">This Field is invalid</span>
15    <br/>
16    <button class="btn btn-primary">Post</button>
17  </form>
18 </div>
19
```

Routings in Angular:

As we have seen our any angular application will have multiple components. Each component will have it's own template. So, we need a way to navigate in between templates. In order to achieve this we will use Angular router.

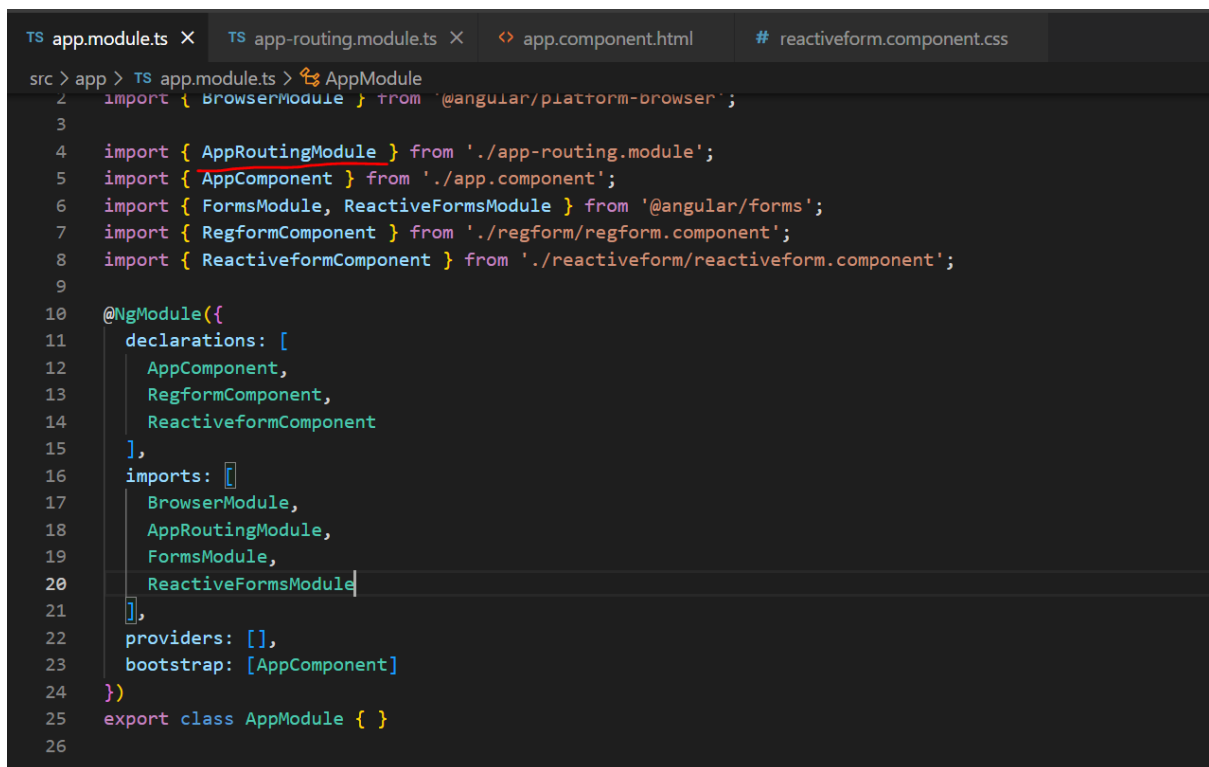
So, here we will see how we can navigate in between components template using router.

As we discussed on very first day about SPA, here router going to help us. We can also use `` but this will load again page load and required data, which we don't want. So here will use concept of routing.

For understanding it better we will use two components which we have created for understanding of forms.

Steps to implement Routes in our application.

Step 1. Check out whether **app.module.ts** file has imported **AppRoutingModule**



```
src > app > TS app.module.ts > AppModule
2 import { BrowserModule } from '@angular/platform-browser';
3
4 import { AppRoutingModule } from './app-routing.module';
5 import { AppComponent } from './app.component';
6 import { FormsModule, ReactiveFormsModule } from '@angular/forms';
7 import { RegformComponent } from './regform/regform.component';
8 import { ReactiveformComponent } from './reactiveform/reactiveform.component';
9
10 @NgModule({
11   declarations: [
12     AppComponent,
13     RegformComponent,
14     ReactiveformComponent
15   ],
16   imports: [
17     BrowserModule,
18     AppRoutingModule,
19     FormsModule,
20     ReactiveFormsModule
21   ],
22   providers: [],
23   bootstrap: [AppComponent]
24 })
25 export class AppModule { }
26
```

Step 2. Open app-routing.module.ts file and here in array of route provide the components for which we have to provide navigation.

```
TS app.module.ts TS app-routing.module.ts X app.component.html # reactiveform.component.css
src > app > TS app-routing.module.ts > routes > component
1 import { NgModule } from '@angular/core';
2 import { RouterModule, Routes } from '@angular/router';
3 import { ReactiveformComponent } from '../reactiveform/reactiveform.component';
4 import { RegformComponent } from '../regform/regform.component';
5
6 const routes: Routes = [
7   {path: 'regform' ,component:RegformComponent},
8   {path: 'reactiveform', component:ReactiveformComponent}
9 ];
10
11 @NgModule({
12   imports: [RouterModule.forRoot(routes)],
13   exports: [RouterModule]
14 })
15 export class AppRoutingModule { }
16
```

Step 3: Now, open our app.component.html file. Delete selectors for both components. And at the end add one directive <router-outlet></router-outlet>

```
TS app.module.ts X TS app-routing.module.ts X app.component.html X # reactiveform.component.css
src > app > app.component.html > html > body > nav.navbar.navbar-expand-lg.navbar-dark.bg-dark > div.container-fluid > div#navbar
1 <!doctype html>
2 <html lang="en">
3 <head>
4   <meta charset="utf-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1">
6   <title>Bootstrap demo</title>
7   <link href="https://cdn.jsdelivrivr.net/npm/bootstrap@5.2.3/dist/css/bootstrap.min.css" rel="stylesheet" i
8 </head>
9 <body>
10   <nav class="navbar navbar-expand-lg navbar-dark bg-dark">
11     <div class="container-fluid">
12       <a class="navbar-brand" href="#">Home</a>
13       <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarNavAl
14         <span class="navbar-toggler-icon"></span>
15     </button>
16     <div class="collapse navbar-collapse" id="navbarNavAltMarkup">
17       <div class="navbar-nav">
18         <a class="nav-link active" aria-current="page" routerLink="/regform">Register</a>
19         <a class="nav-link" routerLink="/reactiveform">Post Your Comment</a>
20         <a class="nav-link" href="#">About Us</a>
21       </div>
22     </div>
23   </nav>
24   <h1 style="text-align: center; color: tomato">Welcome To Citiustech</h1>
25   <script src="https://cdn.jsdelivrivr.net/npm/bootstrap@5.2.3/dist/js/bootstrap.bundle.min.js" integrity="s
26 </body>
27 </html>
28 <router-outlet></router-outlet>
```

Step 4.Now, open our browser and change url. Test by adding /regform and /reactiveform

Step 5. Now, we will have to add this effect on button click.

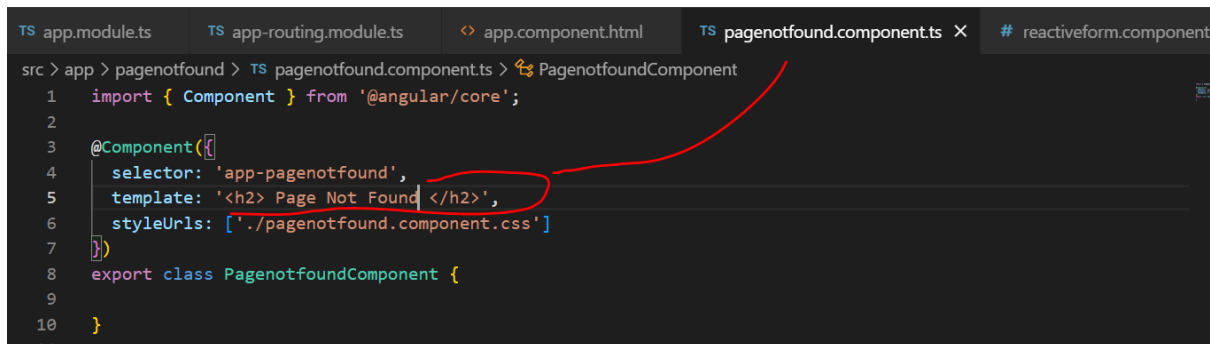
Open our app.component.html file.

```
src > app > app.component.html > html > body > nav.navbar.navbar-expand-lg.navbar-dark.bg-dark > div.container-fluid > div#navbarN
1 <!doctype html>
2 <html lang="en">
3 <head>
4   <meta charset="utf-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1">
6   <title>Bootstrap demo</title>
7   <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/css/bootstrap.min.css" rel="stylesheet" i
8 </head>
9 <body>
10   <nav class="navbar navbar-expand-lg navbar-dark bg-dark">
11     <div class="container-fluid">
12       <a class="navbar-brand" href="#">Home</a>
13       <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarNavAl
14         <span class="navbar-toggler-icon"></span>
15       </button>
16       <div class="collapse navbar-collapse" id="navbarNavAltMarkup">
17         <div class="navbar-nav">
18           <a class="nav-link active" aria-current="page" routerLink="/regform">Register</a>
19           <a class="nav-link" routerLink="/reactiveform">Post Your Comment</a>
20           <a class="nav-link" href="#">About Us</a>
21         </div>
22       </div>
23     </div>
24   </nav>
25   <h1 style="text-align: center; color: tomato">Welcome To Citiustech</h1>
26   <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/js/bootstrap.bundle.min.js" integrity="s
27 </body>
28 </html>
29 <router-outlet></router-outlet>
```

Step 6. Wildcart routes.--> Let's consider a scenario, where this might be possible that sometime we will use some url in our application which is not available so our application will throws error. So, in order to handle such scenarios we can use wild cart routes.

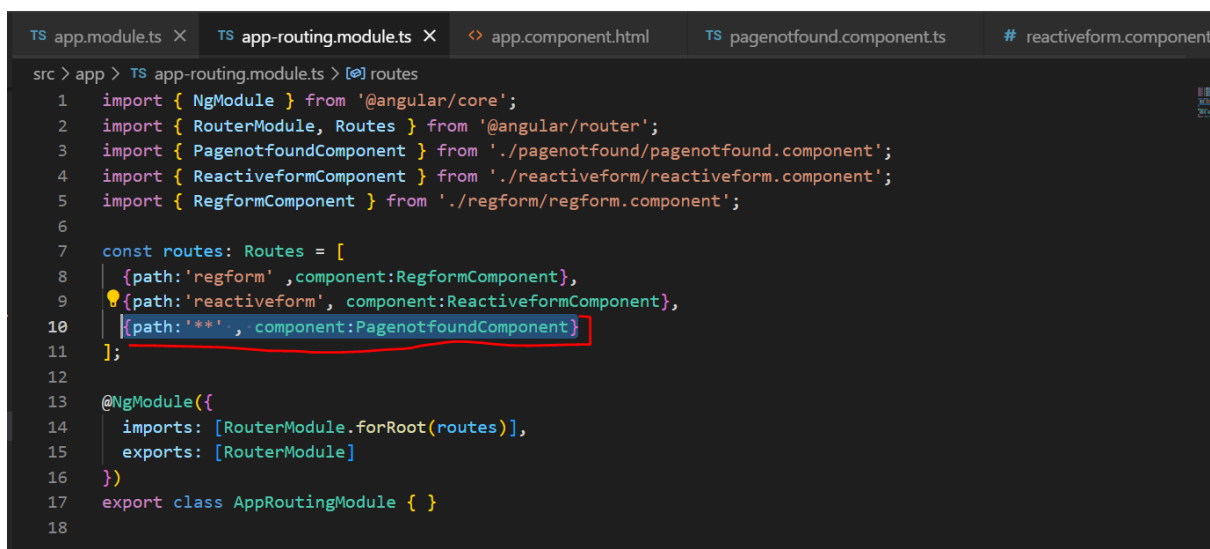
In order to achieve this we will create new component. **ng g c pagenotfound**

Step 7. Open pagenotfound.component.ts file. And just add a simple line in component decorator.



```
TS app.module.ts TS app-routing.module.ts app.component.html TS pagenotfound.component.ts X # reactiveform.component
src > app > pagenotfound > TS pagenotfound.component.ts > PagenotfoundComponent
1 import { Component } from '@angular/core';
2
3 @Component({
4   selector: 'app-pagenotfound',
5   template: '<h2> Page Not Found </h2>',
6   styleUrls: ['./pagenotfound.component.css']
7 })
8 export class PagenotfoundComponent {
9
10 }
11
```

Step 8. Now open app-routing.module.ts



```
TS app.module.ts X TS app-routing.module.ts X app.component.html TS pagenotfound.component.ts # reactiveform.component
src > app > TS app-routing.module.ts > routes
1 import { NgModule } from '@angular/core';
2 import { RouterModule, Routes } from '@angular/router';
3 import { PagenotfoundComponent } from '../pagenotfound/pagenotfound.component';
4 import { ReactiveformComponent } from '../reactiveform/reactiveform.component';
5 import { RegformComponent } from '../regform/regform.component';
6
7 const routes: Routes = [
8   {path: 'regform' , component:RegformComponent},
9   {path: 'reactiveform', component:ReactiveformComponent},
10  {path: '**' , component:PagenotfoundComponent}
11 ];
12
13 @NgModule({
14   imports: [RouterModule.forRoot(routes)],
15   exports: [RouterModule]
16 })
17 export class AppRoutingModule { }
18
```

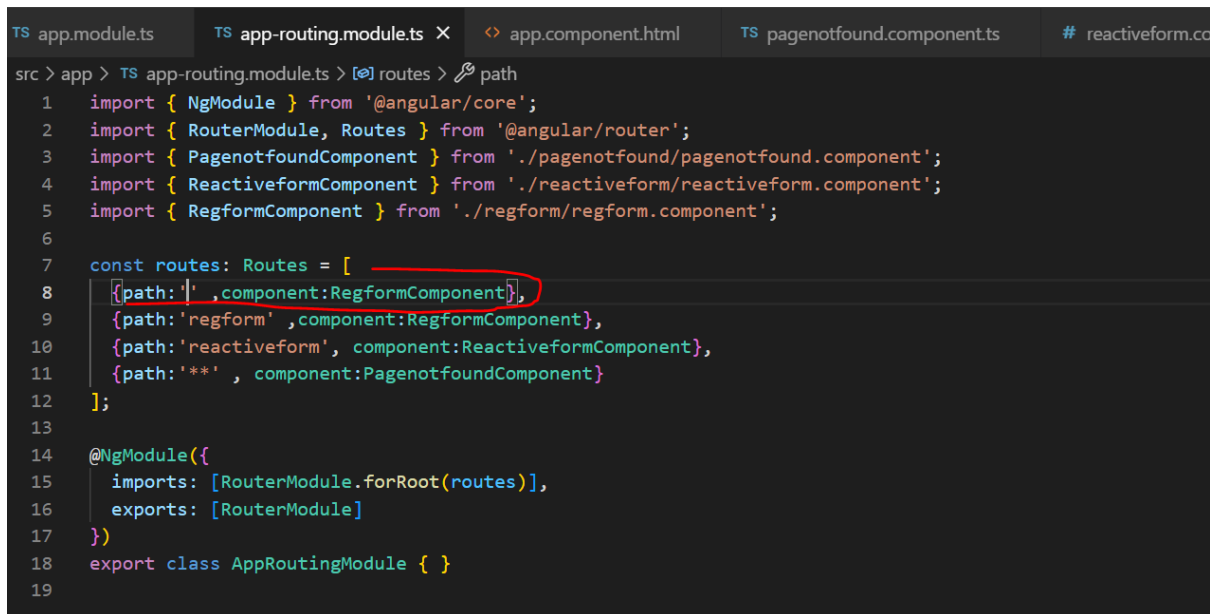
Note:- Add wildcard route at the end of array of routes. Because if we will add wildcard route on top then every time it will consider same route so we will get output, page not found only.

But now, here is one problem that we are observing here is Page not found displaying on Home page too.

The reason is there is no any matching route.

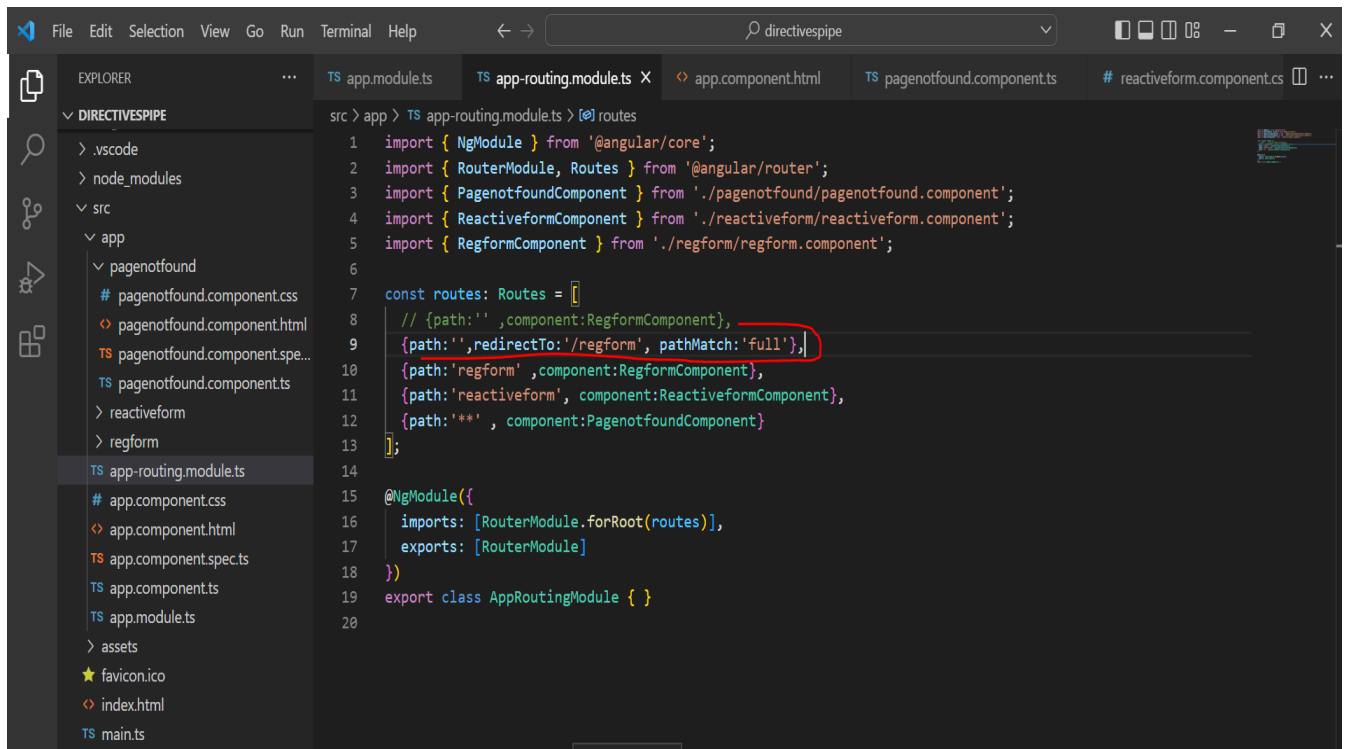
In order to avoid this problem we will have to define one more element in array of routes for component which we will have to display as landing page.

Add given line of code to fix our issue.



```
src > app > TS app-routing.module.ts > routes > path
1  import { NgModule } from '@angular/core';
2  import { RouterModule, Routes } from '@angular/router';
3  import { PagenotfoundComponent } from '../pagenotfound/pagenotfound.component';
4  import { ReactiveformComponent } from '../reactiveform/reactiveform.component';
5  import { RegformComponent } from '../regform/regform.component';
6
7  const routes: Routes = [
8    {path: '', component:RegformComponent},
9    {path:'regform' ,component:RegformComponent},
10   {path:'reactiveform', component:ReactiveformComponent},
11   {path:'**' , component:PagenotfoundComponent}
12 ];
13
14 @NgModule({
15   imports: [RouterModule.forRoot(routes)],
16   exports: [RouterModule]
17 })
18 export class AppRoutingModule { }
19
```

We can also replace line number 8 by



```
src > app > TS app-routing.module.ts > routes
1  import { NgModule } from '@angular/core';
2  import { RouterModule, Routes } from '@angular/router';
3  import { PagenotfoundComponent } from '../pagenotfound/pagenotfound.component';
4  import { ReactiveformComponent } from '../reactiveform/reactiveform.component';
5  import { RegformComponent } from '../regform/regform.component';
6
7  const routes: Routes = [
8    // {path: '', component:RegformComponent},
9    {path:'',redirectTo:'/regform', pathMatch:'full'},
10   {path:'regform' ,component:RegformComponent},
11   {path:'reactiveform', component:ReactiveformComponent},
12   {path:'**' , component:PagenotfoundComponent}
13 ];
14
15 @NgModule({
16   imports: [RouterModule.forRoot(routes)],
17   exports: [RouterModule]
18 })
19 export class AppRoutingModule { }
20
```

