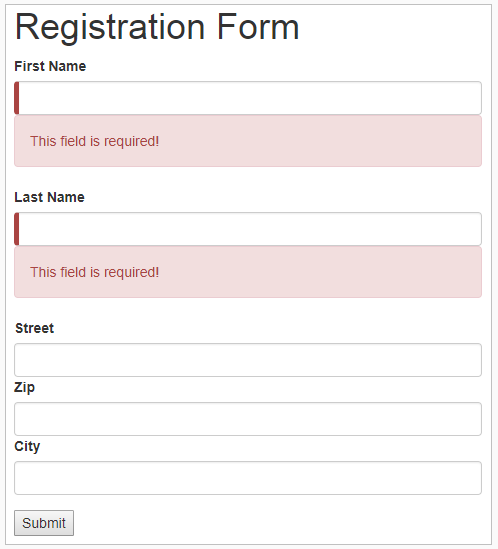
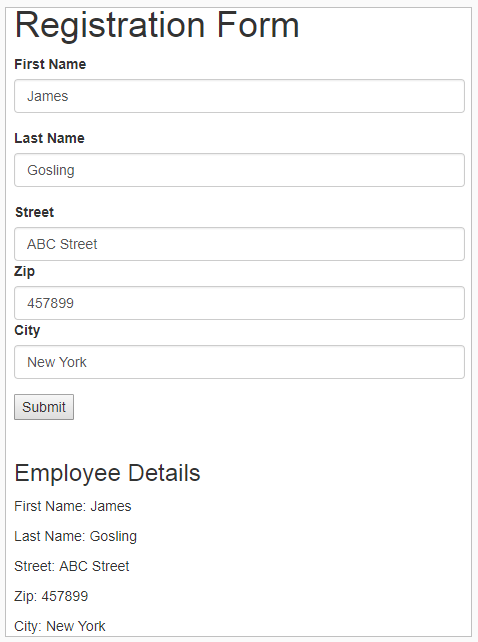
**Demo Steps:**

**Problem Statement:**Creating an employee registration form as a reactive form as shown below



 It renders the values at the bottom after clicking submit button



1. Write the below code in **app.module.ts**
2. import { BrowserModule } from '@angular/platform-browser';
3. import { NgModule } from '@angular/core';
4. import { ReactiveFormsModule } from '@angular/forms';
5. import { AppComponent } from './app.component';
6. import { RegistrationFormComponent } from './registration-form/registration-form.component';
7. @NgModule({
8. declarations: [
9. AppComponent,
10. RegistrationFormComponent
11. ],
12. imports: [
13. BrowserModule,
14. ReactiveFormsModule
15. ],
16. providers: [],
17. bootstrap: [AppComponent]
18. })
19. export class AppModule { }

1. Create a component called **RegistrationForm** using the following CLI command
2. ng generate component RegistrationForm

1. Add the following code in the **registration-form.component.ts** file
2. import { Component, OnInit } from '@angular/core';
3. import { FormBuilder, FormGroup, Validators } from '@angular/forms';
4. @Component({
5. selector: 'app-registration-form',
6. templateUrl: './registration-form.component.html',
7. styleUrls: ['./registration-form.component.css']
8. })
9. export class RegistrationFormComponent implements OnInit {
10. registerForm!: FormGroup;
11. submitted:boolean=false;
12. constructor(private formBuilder: FormBuilder) { }
13. ngOnInit() {
14. this.registerForm = this.formBuilder.group({
15. firstName: ['', Validators.required],
16. lastName: ['', Validators.required],
17. address: this.formBuilder.group({
18. street: [],
19. zip: [],
20. city: []
21. })
22. });
23. }
24. }

1. Write the below code in **registration-form.component.html**
2. <div class="container">
3. <h1>Registration Form</h1>
4. <form [formGroup]="registerForm" (ngSubmit)="submitted=true">
5. <div class="form-group">
6. <label>First Name</label>
7. <input type="text" class="form-control" formControlName="firstName">
8. <p \*ngIf="registerForm.controls['firstName'].errors" class="alert alert-danger">This field is required!</p>
9. </div>
10. <div class="form-group">
11. <label>Last Name</label>
12. <input type="text" class="form-control" formControlName="lastName">
13. <p \*ngIf="registerForm.controls['lastName'].errors" class="alert alert-danger">This field is required!</p>
14. </div>
15. <div class="form-group">
16. <fieldset formGroupName="address">
17. <label>Street</label>
18. <input type="text" class="form-control" formControlName="street">
19. <label>Zip</label>
20. <input type="text" class="form-control" formControlName="zip">
21. <label>City</label>
22. <input type="text" class="form-control" formControlName="city">
23. </fieldset>
24. </div>
25. <button type="submit">Submit</button>
26. </form>
27. <br/>
28. </div>

**Line 3:** formGroup is a directive that binds HTML form with the FormGroup property created inside a component class. We have created a FormGroup in component with name registerForm. Here form tag is binded with FormGroup registerForm

**Line 6, 11:** Two input fields for first name and last name are binded with the form controls created in the component using the **formControlName**directive

**Line 24:** When the submit button is clicked, the form is submitted using the ngSubmit directive, which initializes *submitted*property value to *true.*

**Line 27:** div tag will be hidden if the form is not submitted

**Line 29-33:** Using get() method of FormGroup, we are fetching values of each FormControl and rendering it

1. Write the below code in **registration-form.component.css**
2. .ng-valid[required] {
3. border-left: 5px solid *#42A948; /\* green \*/*
4. }
5. .ng-invalid:not(form) {
6. border-left: 5px solid *#a94442; /\* red \*/*
7. }

**Line 1-3:** ng-valid CSS class changes left border of the input field to green if form control has valid input

**Line 5-7:** ng-invalid CSS class changes left border of the input field to red if form control has invalid data

1. Write the below code in **app.component.html**
2. <app-registration-form></app-registration-form>

**Line 1:** Loads RegistrationFormComponent in the root component

1. Save and run the files to check the output in the browser