

Marvellous Infosystems: Angular Assignment 14

Name: Shrirang Jagdish Nikam

Enrollment No: 396AM_Shrirang

1. We have to design above form as Reactive form.

We have to perform input validation for below form controls as

- First Name : It should be compulsory filed. It should contains only alphabets (A-Z / a- z).
- Phone : It should be compulsory filed. It should contains only digits. Length of field should be 10. (0-9)
- Email : It should be compulsory filed. Email filed contains any alphanumeric contents before @ symbol. After @ symbol there should be minimum 1 letter before . and after . there should be minimum 2 letters.
- City : It should be compulsory field which contains minimum 4 alphabets.
- State : It should contains three options as Maharashtra, Gujarat, Uttar Pradesh
- ZIP code : It should contains all digits and length of zip code should be 5.
- Comments : It should contains minimum 30 characters.
- Submit Form : Button should be disabled till all the form fields are filled with valid data.

Note : All the form controls should be in one form group.

Use FormBuilder service.

Use Visual Feedback as well as Error message.

Answer:

Form.component.html:

```
<div class="container">
  <h2>Reactive Form</h2>
  <form [formGroup]="myForm" (ngSubmit)="submitForm()">
    <div class="form-group">
      <label for="firstName">First Name:</label>
      <input type="text" class="form-control" formControlName="firstName">
      <div *ngIf="isFieldInvalid('firstName')" class="text-danger">First name
is required and should contain only alphabets.</div>
    </div>

    <div class="form-group">
      <label for="phone">Phone:</label>
      <input type="text" class="form-control" formControlName="phone">
      <div *ngIf="isFieldInvalid('phone')" class="text-danger">Phone is
required and should contain 10 digits.</div>
    </div>

    <div class="form-group">
      <label for="email">Email:</label>
      <input type="text" class="form-control" formControlName="email">
      <div *ngIf="isFieldInvalid('email')" class="text-danger">Email is
required and should be in the format 'username@example.com'.</div>
    </div>

    <div class="form-group">
      <label for="city">City:</label>
      <input type="text" class="form-control" formControlName="city">
      <div *ngIf="isFieldInvalid('city')" class="text-danger">City is required
and should contain at least 4 alphabets.</div>
    </div>

    <div class="form-group">
      <label for="state">State:</label>
      <select class="form-control" formControlName="state">
        <option value="">Select a state</option>
        <option value="Maharashtra">Maharashtra</option>
        <option value="Gujarat">Gujarat</option>
        <option value="Uttar Pradesh">Uttar Pradesh</option>
      </select>
      <div *ngIf="isFieldInvalid('state')" class="text-danger">Please select a
state.</div>
  </form>
</div>
```

```

    </div>

    <div class="form-group">
      <label for="zipCode">ZIP code:</label>
      <input type="text" class="form-control" formControlName="zipCode">
      <div *ngIf="isFieldInvalid('zipCode')" class="text-danger">ZIP code is
required and should contain 5 digits.</div>
    </div>

    <div class="form-group">
      <label for="comments">Comments:</label>
      <textarea class="form-control" formControlName="comments"></textarea>
      <div *ngIf="isFieldInvalid('comments')" class="text-danger">Comments are
required and should contain at least 30 characters.</div>
    </div>

    <button type="submit" class="btn btn-primary"
[disabled]="myForm.invalid">Submit Form</button>
  </form>
</div>

```

Form.component.ts:

```

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

@Component({
  selector: 'app-form',
  templateUrl: './form.component.html',
  styleUrls: ['./form.component.css']
})
export class FormComponent implements OnInit {
  myForm: FormGroup;

  constructor(private formBuilder: FormBuilder) { }

  ngOnInit(): void {
    this.myForm = this.formBuilder.group({
      firstName: ['', [Validators.required, Validators.pattern('^[a-zA-Z]+$')]],
      phone: ['', [Validators.required, Validators.pattern('^([0-9]){10}$')]],
      email: ['', [Validators.required, Validators.pattern('[a-zA-Z0-9]+@[a-zA-Z]+\\.[a-zA-Z]{2,}$')]],
      city: ['', [Validators.required, Validators.minLength(4)]]],

```

```

    state: ['', Validators.required],
    zipCode: ['', [Validators.required, Validators.pattern('^[0-9]{5}$')]],
    comments: ['', [Validators.required, Validators.minLength(30)]]
  });
}

isFieldInvalid(field: string) {
  const control = this.myForm.get(field);
  return control.invalid && control.touched;
}

submitForm() {
  if (this.myForm.valid) {
    console.log(this.myForm.value);
  } else {
    this.markFieldsAsTouched();
  }
}

markFieldsAsTouched() {
  Object.keys(this.myForm.controls).forEach(field => {
    const control = this.myForm.get(field);
    control.markAsTouched({ onlySelf: true });
  });
}
}

```

Form.component.css:

```

.container {
  max-width: 400px;
  margin: 50px auto;
}

.text-danger {
  color: red;
}

```

Form-builder.service.ts:

```

import { Injectable } from '@angular/core';

```

```

@Injectable({
  providedIn: 'root'
})
export class FormBuilderService {

  constructor() { }
}

```

App.module.ts:

```

import { NgModule } from '@angular/core';
import { BrowserModule } from '@angular/platform-browser';

import { AppComponent } from './app.component';
import { FormComponent } from './form/form.component';
import { FormsModule, ReactiveFormsModule } from '@angular/forms';

@NgModule({
  declarations: [
    AppComponent,
    FormComponent
  ],
  imports: [
    BrowserModule,
    ReactiveFormsModule,
    FormsModule
  ],
  providers: [],
  bootstrap: [AppComponent]
})
export class AppModule { }

```

angular.json:

```

{
  "$schema": "./node_modules/@angular/cli/lib/config/schema.json",
  "version": 1,
  "newProjectRoot": "projects",
  "projects": {
    "angular-form-validation": {

```

```
"projectType": "application",
"schematics": {},
"root": "",
"sourceRoot": "src",
"prefix": "app",
"architect": {
  "build": {
    "builder": "@angular-devkit/build-angular:browser",
    "options": {
      "outputPath": "dist/angular-form-validation",
      "index": "src/index.html",
      "main": "src/main.ts",
      "polyfills": [
        "zone.js"
      ],
      "tsConfig": "tsconfig.app.json",
      "assets": [
        "src/favicon.ico",
        "src/assets"
      ],
      "styles": [
        "node_modules/bootstrap/dist/css/bootstrap.min.css",
        "src/styles.css"
      ],
      "scripts": []
    },
    "configurations": {
      "production": {
        "budgets": [
          {
            "type": "initial",
            "maximumWarning": "500kb",
            "maximumError": "1mb"
          },
          {
            "type": "anyComponentStyle",
            "maximumWarning": "2kb",
            "maximumError": "4kb"
          }
        ],
        "outputHashing": "all"
      },
      "development": {
        "buildOptimizer": false,
        "optimization": false,
```

```
        "vendorChunk": true,  
        "extractLicenses": false,  
        "sourceMap": true,  
        "namedChunks": true  
    }  
},  
"defaultConfiguration": "production"  
},  
"serve": {  
    "builder": "@angular-devkit/build-angular:dev-server",  
    "configurations": {  
        "production": {  
            "browserTarget": "angular-form-validation:build:production"  
        },  
        "development": {  
            "browserTarget": "angular-form-validation:build:development"  
        }  
    },  
    "defaultConfiguration": "development"  
},  
"extract-i18n": {  
    "builder": "@angular-devkit/build-angular:extract-i18n",  
    "options": {  
        "browserTarget": "angular-form-validation:build"  
    }  
},  
"test": {  
    "builder": "@angular-devkit/build-angular:karma",  
    "options": {  
        "polyfills": [  
            "zone.js",  
            "zone.js/testing"  
        ],  
        "tsConfig": "tsconfig.spec.json",  
        "assets": [  
            "src/favicon.ico",  
            "src/assets"  
        ],  
        "styles": [  
            "src/styles.css"  
        ],  
        "scripts": []  
    }  
}  
}
```


```
}  
},  
"cli": {  
  "analytics": "fba138f1-8626-48da-940a-070ce02c1135"  
}  
}
```

App.component.html:

```
<app-form></app-form>
```

Output:

Contact Our
Customer Support Center



Contact Information

Please fill out your contact information below.

Name*

First NameLast Name

Email*

Phone*

Address*

