appcomponent.html

```
<div>
<img src="assets/laptop.jpg" style="float:right;">
<h1 style="text-align: center;color: blueviolet;">Contact our </h1>
<h1 style="text-align: center;"> Customer support center </h1>
</div>
<br>
<br>
<div style="text-align:center;">
 <br>
  <h4>Contact information </h4>
Please fill out your contact informations below
</div>
<form [formGroup] = "MarvellousForm" (ngSubmit)="MarvellousForm.status !==
'INVALID' && registerFn()">
Name*
<div class="form-group">
    <div class="row">
      <div class="col">
          <input type="text" class="form-control" formcontrolName="firstname"</pre>
          [class.is-invalid] = "MarvellousForm.get('firstname')?.invalid &&
MarvellousForm.get('firstname')?.touched"
           type="text" class="form-control" formControlName="firstname"
          [ngClass]=" { errors : getcontrol('firstname')?.errors &&
getcontrol('firstname')?.touched }">first name
          <div *ngIf="getcontrol('firstname')?.errors &&</pre>
getcontrol('firstname')?.touched">
            <small class="text-danger"</pre>
*ngIf="getcontrol('firstname')?.errors?.['required']"> first name is
req</small>
    <small class="text-danger"</pre>
*ngIf="getcontrol('firstname')?.errors?.['maxlength']||getcontrol('firstname')
?.errors?.['minlength']"> name should be of atleast 3 char/small>
    <small class="text-danger"</pre>
*ngIf="getcontrol('firstname')?.errors?.['pattern']">
number notalloud special symbol not alloud mid char should not be no or
special symbol</small>
```

```
</div>
        </div>
          <div class="col">
         <!--- <input type="text" class="form-control"
formcontrolName="lname">Lastname-->
          <input [class.is-invalid] = "MarvellousForm.get('lname')?.invalid &&</pre>
MarvellousForm.get('lname')?.touched" type="text" class="form-control"
formControlName="lname">Last name
    <small [class.d-none]="MarvellousForm.get('lname')?.valid ||</pre>
MarvellousForm.get('lname')?.untouched" class="text-danger">lname is
required</small>
   <small [ngClass]=" { errors : getcontrol('lname')?.errors &&</pre>
getcontrol('lname')?.touched }"></small>
    <small class="text-danger"</pre>
*ngIf="getcontrol('lname')?.errors?.['maxlength']"> only 10 char
alloud</small>
    <small class="text-danger"</pre>
*ngIf="getcontrol('lname')?.errors?.['minlength']"> atleast 4 char
alloud</small>
  </div>
    </div>
</div>
<hr>
<div class="form-group">
        <label>Email *</label>
         <input [class.is-invalid] = "MarvellousForm.get('email')?.invalid &&</pre>
MarvellousForm.get('email')?.touched"
         type="email" class="form-control" formControlName="email"
         [ngClass]=" { errors : getcontrol('email')?.errors &&
getcontrol('email')?.touched }">email
         <div *ngIf="getcontrol('email')?.errors &&</pre>
getcontrol('email')?.touched">
           <small class="text-danger"</pre>
*ngIf="getcontrol('email')?.errors?.['required']"> email is req</small>
   <small class="text-danger"</pre>
*ngIf="getcontrol('email')?.errors?.['maxlength']"> only 32 char
alloud</small>
   <small class="text-danger"</pre>
*ngIf="getcontrol('email')?.errors?.['pattern']"> must be in proper
format</small>
         </div>
</div>
```

```
<br>
<div class="form-group">
 <label>Phone *</label>
 <!---<input type="number" class="form-control" formcontrolName="phone">-->
 <input type="text" class="form-control" id="mobile" placeholder="Phone"</pre>
formControlName="mobile"
 [class.is-invalid] = "MarvellousForm.get('mobile')?.invalid &&
MarvellousForm.get('mobile')?.touched"
 [ngClass]="{error : getcontrol('mobile')?.errors &&
getcontrol('mobile')?.touched}">
      <div *ngIf="getcontrol('mobile')?.errors &&</pre>
getcontrol('mobile')?.touched">
        <small class="text-danger"</pre>
*ngIf="getcontrol('mobile')?.errors?.['required']">Mobile field is
requird</small>
        <small class="text-danger"</pre>
*ngIf="getcontrol('mobile')?.errors?.['minlength'] ||
getcontrol('mobile')?.errors?.['maxlength'] ||
getcontrol('mobile')?.errors?.['pattern']">Please enter a valid mobile
no.</small>
      </div>
</div>
<br>
<div class="form-group">
  <label>Address *</label>
  <!--<label for="lname" class="form-label">Last Name</label>-->
      <input type="text" class="form-control" placeholder="adress"</pre>
formControlName="Address"
      [ngClass]="{error : getcontrol('Address')?.errors &&
getcontrol('Address')?.touched}"
      [class.is-invalid] = "MarvellousForm.get('Address')?.invalid &&
MarvellousForm.get('Address')?.touched" >
      <div *ngIf="getcontrol('Address')?.errors &&</pre>
getcontrol('Address')?.touched">
        <small class="text-danger"</pre>
*ngIf="getcontrol('Address')?.errors?.['required']">Address field is
requird</small>
```

Appcomponent.ts

```
import { Component } from '@angular/core';
import { CommonModule } from '@angular/common';
import { RouterOutlet } from '@angular/router';
import {AbstractControl, FormControl, FormGroup, ReactiveFormsModule} from
'@angular/forms';
import {FormBuilder, Validators} from '@angular/forms'
@Component({
  selector: 'app-root',
  standalone: true,
  imports: [CommonModule, RouterOutlet,ReactiveFormsModule],
  templateUrl: './app.component.html',
  styleUrl: './app.component.css',
  providers:[FormBuilder, Validators],
})
export class AppComponent {
 title = 'reactiveformAss15';
//// Inject FormBuilder service
```

```
emailpattern=/^[a-zA-Z0-9._-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,4}$/;
addresspattern=/^(\d+) ?([A-Za-z](?= ))? (.*?) ([^ ]+?) ?((?<= )APT)? ?((?<=
)\d*)?$/;
MarvellousForm=new FormGroup(
    firstname: new
FormControl("",[Validators.required,Validators.maxLength(32),Validators.minLen
gth(3),
      Validators.pattern('[a-zA-z].*')]),
      lname:new
FormControl("",[Validators.required,Validators.maxLength(10),Validators.minLen
gth(4),
        Validators.pattern('[a-zA-z].*')]),
        email :new
FormControl("",[Validators.required,Validators.maxLength(32),
          Validators.pattern(this.emailpattern)]),
         mobile :new
FormControl("",[Validators.required,Validators.minLength(10),Validators.maxLen
gth(10)]),
          Address
:new FormControl("",[Validators.required,Validators.maxLength(60),
            Validators.pattern(this.addresspattern)]),
getcontrol(name:any):AbstractControl|null
  return this.MarvellousForm.get(name);
registerFn()
  console.log(this.MarvellousForm.value)
```