

ASS15 reactive form

appcomponent.html

```
<div>

<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"
      [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 &&
getcontrol('firstname')?.touched">
        <small class="text-danger"
*ngIf="getcontrol('firstname')?.errors?.['required']"> first name is
req</small>
        <small class="text-danger"
*ngIf="getcontrol('firstname')?.errors?.['maxlength']||getcontrol('firstname')
?.errors?.['minlength']"> name should be of atleast 3 char</small>
        <small class="text-danger"
*ngIf="getcontrol('firstname')?.errors?.['pattern']">
number notalloud special symbol not alloud mid char should not be no or
special symbol</small>
```



```

        <small class="text-danger"
*ngIf="getcontrol('Address')?.errors?.['maxlength']">Only 32 characters are
allowed.</small>
        <small class="text-danger"
*ngIf="getcontrol('Address')?.errors?.['pattern']">proper format</small>

    </div>

</div>
<br>

<div class="form-group">
    <div class="text-center">
        <button class=" btn btn-primary" [disabled]="MarvellousForm.status ===
'INVALID'">Submit</button>
    </div>
</div>

</form>

```

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',
    styleUrls: ['./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}$/;
addresspattern=/^(\d+)?([A-Za-z](?=\s)?(.*?)([^\s]+)?((?<=)\sAPT)?((?<=)\s\d*)?$/;

MarvellousForm=new FormGroup(
  {
    firstname: new
FormControl("",[Validators.required,Validators.maxLength(32),Validators.minLength(3),
  Validators.pattern('[a-zA-z].*')]),
    lname:new
FormControl("",[Validators.required,Validators.maxLength(10),Validators.minLength(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.maxLength(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)
}
}

```