**Name**-Abdul Nadeem

**Id**-755143

**Component-**

1-admin-comp

\*admin-reg

\*admin-show

2-homepage

3-Shared

\*voter.model.ts

\* voter.service.ts

4-voter-comp

\*Voter-reg

\*Voter-show

App module .ts

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

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

import {ToastrModule} from 'ngx-toastr';

import {FormsModule} from '@angular/forms';

import {BrowserAnimationsModule} from '@angular/platform-browser/animations';

import { AppRoutingModule } from './app-routing.module';

import { AppComponent } from './app.component';

import { VoterCompComponent } from './voter-comp/voter-comp.component';

import { VoterRegComponent } from 'src/app/voter-comp/voter-reg/voter-reg.component';

import { VoterShowComponent } from 'src/app/voter-comp/voter-show/voter-show.component';

import { VoterService } from './Shared/voter.service';

import {HttpClientModule} from '@angular/common/http';

import {RouterModule, Router} from '@angular/router';

import { AdminCompComponent } from './admin-comp/admin-comp.component';

import { AdminRegComponent } from './admin-comp/admin-reg/admin-reg.component';

import { AdminShowComponent } from './admin-comp/admin-show/admin-show.component';

import { HomepageComponent } from './homepage/homepage.component';

import {voterNamefilterPipe,voterAadharfilterPipe, dateFormatPipe} from '../app/voter-comp/voter-show/filterPipe'

@NgModule({

declarations: [

AppComponent,

VoterCompComponent,

VoterRegComponent,

VoterShowComponent,

AdminCompComponent,

AdminRegComponent,

AdminShowComponent,

HomepageComponent,

voterNamefilterPipe,

voterAadharfilterPipe,

dateFormatPipe

],

imports: [

BrowserModule,

FormsModule,

AppRoutingModule,

HttpClientModule,

BrowserAnimationsModule,

ToastrModule.forRoot(),

RouterModule.forRoot([

{path:'Register',component:VoterRegComponent},

{path:'Show',component:VoterShowComponent},

{path:'Admin',component:AdminCompComponent},

{path:'',component:HomepageComponent}

])

],

providers: [VoterService],

bootstrap: [AppComponent]

})

export class AppModule { }

App component.html

<div class="container-fluid">

<app-voter-comp></app-voter-comp>

</div>

Homepage component.html

<link rel="stylesheet" href="https://www.w3schools.com/w3css/4/w3.css">

<link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.8.1/css/all.css">

<div class="container-fluid" id="containercorousel">

<div id="myCarousel" class="carousel slide" data-ride="carousel" id="corheight">

<ol class="carousel-indicators">

<li data-target="#myCarousel" data-slide-to="0" class="active"></li>

<li data-target="#myCarousel" data-slide-to="1"></li>

<li data-target="#myCarousel" data-slide-to="2"></li>

</ol>

<div class="carousel-inner">

<div class="item active">

<img class="imgCar" src="https://eci.gov.in/uploads/swiperslider\_images/schandra1.jpg.bd87caa0a3add3e482d82c6e9d0df56d.jpg" >

</div>

<div class="item">

<img class="imgCar" src="../../assets/corousel1.jpg" >

</div>

<div class="item">

<img class="imgCar" src="../../assets/corousel2.jfif" >

</div>

</div>

<a class="left carousel-control" href="#myCarousel" data-slide="prev">

<span class="glyphicon glyphicon-chevron-left"></span>

<span class="sr-only">Previous</span>

</a>

<a class="right carousel-control" href="#myCarousel" data-slide="next">

<span class="glyphicon glyphicon-chevron-right"></span>

<span class="sr-only">Next</span>

</a>

</div>

</div>

<div class="container-fluid" id="newscontainer">

<div class=row id="rownews">

<div class="col-md-2" id="news">NEWS<i class="far fa-newspaper" id="glyphnews"></i></div>

<div class="col-md-10" style="padding:0px"> <marquee onmouseover="this.stop();" onmouseout="this.start();"><b>The Election Commission of India Wednesday announced conducting repolling in 13 polling stations in Tamil Nadu, where the Lok Sabha elections were held on April 18. The repolling in Dharmapuri, Tiruvallur, Theni and Erode will be held on May 19|Final Voter turnout of Phase 1, 2, 3 and 4 of the Lok Sabha Elections 2019|The Supreme Court will hear the petition seeking direction to the Election Commission of India (ECI) to prosecute voters who are involved in electoral malpractices.</b></marquee></div></div>

</div>

<div row="container" id="commissionpad">

<div class="row">

<div class="col-md-12" id="commissionrow"><h1>HON'BLE COMMISSION</h1></div>

</div>

</div>

<div class="row" id="cardrow">

<div class="col-md-4">

<div class="w3-card-4" id="imgcontwidth">

<img src="../../assets/Sunil.jpg" alt="Alps" id="imgcard">

<div class="w3-container w3-center" id="imginfo">

<h2>Sh .Sunil Arora</h2>

<h4>Chief Election Commissioner</h4>

</div>

</div></div>

<div class="col-md-4">

<div class="w3-card-4" id="imgcontwidth" >

<img src="../../assets/sushil.jpg" alt="Alps" id="imgcard">

<div class="w3-container w3-center" id="imginfo">

<h2>Sh. Sushil Chandra</h2>

<h4>Election Commissioner</h4>

</div>

</div></div>

<div class="col-md-4">

<div class="w3-card-4" id="imgcontwidth" >

<img src="../../assets/ashok.jpg" alt="Alps" id="imgcard">

<div class="w3-container w3-center" id="imginfo">

<h2>Sh. Ashok Lavasa</h2>

<h4>Election Commissioner</h4>

</div>

</div></div>

</div>

Voter Model.ts

export class Voter{

Id:number;

Name:string;

Address:string;

Gender:string;

District:string;

Image:string;

Mobile:string;

DOB:string;

Aadhar:string;

}

Voter Service.ts

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

import {HttpClient, HttpHeaders, HttpParams} from '@angular/common/http';

import { Observable } from 'rxjs/internal/Observable';

import { getNumberOfCurrencyDigits } from '@angular/common';

import { from } from 'rxjs';

import { Voter } from './voter.model';

@Injectable({

providedIn: 'root'

})

export class VoterService {

formData:Voter;

list:Voter[];

readonly rootURL="http://localhost:57922/api"

constructor(private http:HttpClient) { }

postVoter(formData:Voter)

{

return this.http.post(this.rootURL+'/tblVoters',formData);

}

refreshList():void

{

this.http.get(this.rootURL+'/tblVoters')

.toPromise().then(res=>this.list=res as Voter[]);

}

putVoter(formData:Voter)

{ const headers = new HttpHeaders().set('content-type', 'application/json');

const params = new HttpParams().set('ID', formData.Id.toString());{

return this.http.put<Voter>(this.rootURL+'/tblVoters/'+formData.Id,formData,{headers,params});

}

}

deleteVoter(Id:number)

{

return this.http.delete(this.rootURL+'/tblVoters/'+Id);

}

}

Voter Component.html

<link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.8.1/css/all.css">

<nav class="navbar navbar-default navbar-fixed-top" id="navbarheight">

<div class="navbar-header">

<button type="button" class="navbar-toggle" data-toggle="collapse" data-target="#myNavbar">

<span class="icon-bar"></span>

<span class="icon-bar"></span>

<span class="icon-bar"></span>

</button>

</div>

<div class="collapse navbar-collapse" id="myNavbar">

<ul class="nav navbar-nav navbar-left">

<li><a routerLink=""><img src="../../assets/logo.png" id="navbarlink"></a></li>

<li><a routerLink="Register"><i class="fas fa-user"></i>Register</a></li>

<li><a routerLink="Show"><i class="fas fa-address-card"></i>Show</a></li>

<li><a routerLink="Admin"><i class="fas fa-user-shield"></i>Admin</a></li>

</ul>

</div>

</nav>

<router-outlet></router-outlet>

Voter Reg.html

<div class="form-wrapper" >

<header>

<img src="../../../assets/govtimg.png" id="voterregform">

<h1 style="margin-top:0px">Voter Registration</h1>

</header>

<form #form="ngForm" (submit)=OnSubmit(form) autocomplete="off">

<div class="form-group">

<label>Name:</label>

<input type="text" class="form-control" name="Name" required [(ngModel)]="this.service.formData.Name" #Name=ngModel>

<div class="validation-error" \*ngIf="Name.invalid && Name.touched">Name is required!</div></div>

<div class="form-group">

<label>Address:</label>

<input class="form-control" name="Address" required [(ngModel)]="this.service.formData.Address" #Address="ngModel">

<div class="validation-error" \*ngIf="Address.invalid && Address.touched">Address is required!</div></div>

<div class="form-group">

<label> District:</label>

<input class="form-control" name="District" required [(ngModel)]="this.service.formData.District" #District="ngModel">

<div class="validation-error" \*ngIf="District.invalid && District.touched">District is required!</div></div>

<div class="form-group">

<label> Aadhar Number:</label>

<input class="form-control" name="Aadhar" minlength="12" maxlength="12" required [(ngModel)]="this.service.formData.Aadhar" #Aadhar="ngModel">

<div class="validation-error" \*ngIf="Aadhar.invalid && Aadhar.touched">Aadhar Number is required!</div></div>

<div class="form-group">

<label> Gender:</label>

<select class="form-control" name="Gender" required [(ngModel)]="this.service.formData.Gender" #Gender="ngModel">

<option class="hidden" id="Gender" selected disabled>Gender</option>

<option>Male</option>

<option>Female</option>

</select>

<div class="validation-error" \*ngIf="Gender.invalid && Gender.touched">Gender is required!</div>

</div>

<div class="form-group">

<label> Image:</label>

<input class="form-control" required name="Image" [(ngModel)]="this.service.formData.Image" #Image="ngModel" >

<div class="validation-error" \*ngIf="Image.invalid && Image.touched">Image is required!</div>

</div>

<div class="form-group">

<label>Mobile:</label>

<input class="form-control" required name="Mobile" [(ngModel)]="this.service.formData.Mobile" minlength="10" maxlength="10" #Mobile="ngModel" >

<div class="validation-error" \*ngIf="Mobile.invalid && Mobile.touched">Mobile is required!</div>

</div>

<div class="form-group">

<label> Date Of Birth(YYYY/MM/DD)</label>

<input type="text" class="form-control" required name="DOB" [(ngModel)]="this.service.formData.DOB" #DOB="ngModel" >

<div class="validation-error" \*ngIf="DOB.invalid && DOB.touched">Date of Birth is required!</div>

</div>

<div class="form-group">

<button type="submit" [disabled]="form.invalid" class="btn btn-lg btn-block"> SUBMIT </button>

</div>

</form>

</div>

Voter Reg.ts

import { Component, OnInit } from '@angular/core';

import { VoterService } from 'src/app/Shared/voter.service';

import { ToastrService } from 'ngx-toastr';

import { NgForm } from '@angular/forms';

import { Voter } from 'src/app/Shared/voter.model';

@Component({

selector: 'app-voter-reg',

templateUrl: './voter-reg.component.html',

styleUrls: ['./voter-reg.component.css']

})

export class VoterRegComponent implements OnInit {

public Ids:number[];

public result:any;

VoterId:number;

voter:Voter;

constructor(private service:VoterService,private toastr:ToastrService) {

this.service.formData={

Id:null,

Name:null,

Address:null,

Gender:null,

District:null,

Image:null,

Mobile:null,

DOB:null,

Aadhar:null

}

this.result=this.service.list;

}

ngOnInit() {

this.resetForm();

}

resetForm(form?:NgForm)

{

if(form!=null)

{

form.resetForm();

this.service.formData={

Id:null,

Name:null,

Address:null,

Gender:null,

District:null,

Image:null,

Mobile:null,

DOB:null,

Aadhar:null

}

}

}

OnSubmit(form:NgForm){

this.VoterId=form.value.Id;

this.voter=this.service.list.find(e=>e.Id==this.VoterId);

console.log(this.VoterId);

if(this.voter==null)

{

this.insertRecord(form);

}

else

{

this.updateRecord(form);

}

}

insertRecord(form:NgForm) {

this.service.postVoter(form.value).subscribe(res=>{

this.toastr.success("Inserted Successfully","Voter Registered");

this.resetForm(form);

this.service.refreshList();

});

}

updateRecord(form:NgForm)

{

this.service.putVoter(form.value).subscribe(res=>{

this.toastr.info("Updated Successfully","Voter Details Updated");

this.resetForm(form)

this.service.refreshList();

});

}

}

Voter Show.html

<div class="container" id="votershow">

<div class="row">

<div class="col-md-6"><b> Enter the Name :</b>

<input type="text" [(ngModel)]="voterName" id=txtbrand >

</div>

<div class="col-md-6"><b>Enter the Aadhar Number: </b>

<input type="text" [(ngModel)]="voterAadhar" id=txtpurpose >

</div>

</div></div>

<table id="customers" >

<tr><th>Image</th>

<th>Name</th>

<th>Address</th>

<th>Gender</th>

<th>District</th>

<th>Mobile</th>

<th>DOB</th>

<th>Aadhar</th>

</tr>

<tr \*ngFor="let voter of service.list |voterNamefilterPipe:voterName|voterAadharfilterPipe:voterAadhar">

<td (click)=populateForm(voter)><img src="{{voter.Image}}" id="imgsize"></td>

<td (click)=populateForm(voter)>{{voter.Name}}</td>

<td (click)=populateForm(voter)>{{voter.Address}}</td>

<td (click)=populateForm(voter)>{{voter.Gender}}</td>

<td (click)=populateForm(voter)>{{voter.District}}</td>

<td (click)=populateForm(voter)>{{voter.Mobile}}</td>

<td (click)=populateForm(voter)>{{voter.DOB|date:'yyyy-MM-dd'}}</td>

<td (click)=populateForm(voter)>{{voter.Aadhar}}</td>

</tr>

</table>

Voter Show.ts

import { Component, OnInit } from '@angular/core';

import { VoterService } from 'src/app/Shared/voter.service';

import { Voter } from 'src/app/Shared/voter.model';

import { ToastrService } from 'ngx-toastr';

@Component({

selector: 'app-voter-show',

templateUrl: './voter-show.component.html',

styleUrls: ['./voter-show.component.css']

})

export class VoterShowComponent implements OnInit {

constructor(private service:VoterService,private toastr:ToastrService) { }

ngOnInit() {

this.service.refreshList();

}

populateForm(voter:Voter)

{

this.service.formData=Object.assign({},voter);

}

onDelete(Id:number)

{

if(confirm("Are you sure to Delete this record?")){

console.log(Id);

this.service.deleteVoter(Id).subscribe(res=>

{

this.service.refreshList();

this.toastr.warning("Deleted Successfully","Voter Deleted");

});

}

}

}

Filterpipe.ts

import {Pipe,PipeTransform} from '@angular/core';

import {Voter} from '../../Shared/voter.model'

import { DatePipe } from '@angular/common';

@Pipe({

name:'voterNamefilterPipe'

})

export class voterNamefilterPipe implements PipeTransform{

transform(Voterval:Voter[],voterName:string)

{

if(!Voterval||!voterName)

{

return Voterval;

}

return Voterval.filter((x=>x.Name.toLowerCase().indexOf(voterName.toLowerCase())!==-1));

}

}

@Pipe({

name:'voterAadharfilterPipe'

})

export class voterAadharfilterPipe implements PipeTransform{

transform(Voterval:Voter[],voterAadhar:string)

{

if(!Voterval|| !voterAadhar)

{

return Voterval;

}

return Voterval.filter((x=>x.Aadhar.toLowerCase().indexOf(voterAadhar.toLowerCase())!==-1));

}

}

@Pipe({

name: 'dateFormatPipe',

})

export class dateFormatPipe implements PipeTransform {

transform(value: string) {

{{debugger}}

console.log(value);

var datePipe = new DatePipe("en-US");

value = datePipe.transform(value, 'dd/MM/yyyy');

return value;

}

}

Admin Comp.html

<div class="container" id="admincompcont">

<div row="container" id="admincomppadding">

<div class="row" >

<div class="col-md-12" id="adminregcomp" ><h1>Voter Details</h1></div>

</div>

</div>

<div class="row">

<div class="col-md-3">

<app-admin-reg></app-admin-reg></div>

<div class="col-md-9"><app-admin-show></app-admin-show></div>

</div>

</div>

Admin Reg.html

<form #form="ngForm" (submit)=OnSubmit(form) autocomplete="off">

<label>Id:</label>

<input type="text" class="form-control" name="Id" [(ngModel)]="this.service.formData.Id" #Id=ngModel>

<div class="form-group">

<label>Name:</label>

<input type="text" class="form-control" name="Name" required [(ngModel)]="this.service.formData.Name" #Name=ngModel>

</div>

<div class="form-group">

<label>Address:</label>

<input class="form-control" name="Address" required [(ngModel)]="this.service.formData.Address" #Address="ngModel">

</div>

<div class="form-group">

<label> District:</label>

<input class="form-control" name="District" required [(ngModel)]="this.service.formData.District" #District="ngModel">

</div>

<div class="form-group">

<label> Aadhar Number:</label>

<input class="form-control" name="Aadhar" minlength="12" maxlength="12" required [(ngModel)]="this.service.formData.Aadhar" #Aadhar="ngModel">

</div>

<div class="form-group">

<label> Gender:</label>

<select class="form-control" name="Gender" required [(ngModel)]="this.service.formData.Gender" #Gender="ngModel">

<option class="hidden" id="Gender" selected disabled>Gender</option>

<option>Male</option>

<option>Female</option>

</select>

</div>

<div class="form-group">

<label> Image:</label>

<input class="form-control" required name="Image" [(ngModel)]="this.service.formData.Image" #Image="ngModel" >

</div>

<div class="form-group">

<label>Mobile:</label>

<input class="form-control" required name="Mobile" minlength="10" maxlength="10" [(ngModel)]="this.service.formData.Mobile" #Mobile="ngModel" >

</div>

<div class="form-group">

<label> Date Of Birth(YYYY/MM/DD)</label>

<input type="text" class="form-control" required name="DOB" [(ngModel)]="this.service.formData.DOB" #DOB="ngModel" >

</div>

<div class="form-group">

<button type="submit" [disabled]="form.invalid" class="btn btn-lg btn-block"> SUBMIT </button>

</div>

</form>

Admin Reg.ts

import { Component, OnInit } from '@angular/core';

import { VoterService } from 'src/app/Shared/voter.service';

import { ToastrService } from 'ngx-toastr';

import { NgForm } from '@angular/forms';

import { Voter } from 'src/app/Shared/voter.model';

import { dateFormatPipe } from 'src/app/voter-comp/voter-show/filterPipe';

@Component({

selector: 'app-admin-reg',

templateUrl: './admin-reg.component.html',

styleUrls: ['./admin-reg.component.css']

})

export class AdminRegComponent implements OnInit {

public Ids:number[];

dob1:any;

public result:any;

VoterId:number;

voter:Voter;

constructor(private service:VoterService,private toastr:ToastrService) {

this.service.formData={

Id:null,

Name:null,

Address:null,

Gender:null,

District:null,

Image:null,

Mobile:null,

DOB:null,

Aadhar:null

}

this.result=this.service.list;

}

ngOnInit() {

this.resetForm();

}

resetForm(form?:NgForm)

{

if(form!=null)

{

form.resetForm();

this.service.formData={

Id:null,

Name:null,

Address:null,

Gender:null,

District:null,

Image:null,

Mobile:null,

DOB:null,

Aadhar:null

}

}

}

OnSubmit(form:NgForm){

this.VoterId=form.value.Id;

this.voter=this.service.list.find(e=>e.Id==this.VoterId);

console.log(this.VoterId);

if(this.voter==null)

{

this.insertRecord(form);

}

else

{

this.updateRecord(form);

}

}

insertRecord(form:NgForm) {

this.service.postVoter(form.value).subscribe(res=>{

this.toastr.success("Inserted Successfully","Voter Register");

this.resetForm(form);

this.service.refreshList();

});

}

updateRecord(form:NgForm)

{

this.service.putVoter(form.value).subscribe(res=>{

this.toastr.info("Updated Successfully","Voter Details Updated");

this.resetForm(form)

this.service.refreshList();

});

}

}

Admin Show.html

<table id="customers" style="margin-top:5px;margin-left: 17px" >

<tr><th>Image</th>

<th>Name</th>

<th>Address</th>

<th>Gender</th>

<th>District</th>

<th>Mobile</th>

<th>DOB</th>

<th>Aadhar</th>

<th style="background-color:red">Delete</th>

</tr>

<tr \*ngFor="let voter of service.list">

<td (click)=populateForm(voter)><img src={{voter.Image}} style="height:140px;width:138px"></td>

<td (click)=populateForm(voter)>{{voter.Name}}</td>

<td (click)=populateForm(voter)>{{voter.Address}}</td>

<td (click)=populateForm(voter)>{{voter.Gender}}</td>

<td (click)=populateForm(voter)>{{voter.District}}</td>

<td (click)=populateForm(voter)>{{voter.Mobile}}</td>

<td (click)=populateForm(voter)>{{voter.DOB|date:'yyyy-MM-dd'}}</td>

<td (click)=populateForm(voter)>{{voter.Aadhar}}</td>

<td><button (click)="onDelete(voter.Id)"class=".btn-danger">X</button>

</tr>

</table>

Admin Show.ts

import { Component, OnInit } from '@angular/core';

import { VoterService } from 'src/app/Shared/voter.service';

import { Voter } from 'src/app/Shared/voter.model';

import { ToastrService } from 'ngx-toastr';

@Component({

selector: 'app-admin-show',

templateUrl: './admin-show.component.html',

styleUrls: ['./admin-show.component.css']

})

export class AdminShowComponent implements OnInit {

constructor(private service:VoterService,private toastr:ToastrService) { }

ngOnInit() {

this.service.refreshList();

}

populateForm(voter:Voter)

{

this.service.formData=Object.assign({},voter);

this.service.formData.DOB=voter.DOB.toString().split('T')[0];

}

onDelete(Id:number)

{

if(confirm("Are you sure to Delete this record?")){

console.log(Id);

this.service.deleteVoter(Id).subscribe(res=>

{

this.service.refreshList();

this.toastr.warning("Deleted Successfully","Voter Deleted");

});

}

}

}

Style.css

/\* You can add global styles to this file, and also import other style files \*/

@import '~ngx-toastr/toastr.css';

button[type="submit"]{

background-color: #67dfd4;

}

.fontcolor{

background-color:red;

}

div.validation-error{

width:100%;

font-size: 80%;

color:red;

}

#navbarheight{height:50px}

#votershow{margin-top:100px}

#newscontainer{padding-top: 10px}

#admincomppadding{padding-top:10px}

#admincompcont{margin-top:80px}

#corheight{height:340px}

#containercorousel{margin-top:60px}

#navbarlink{height:40px;width:178px;padding-bottom:4px}

#adminregcomp{background-color:orange;text-align: center;color: white}

#voterregform{height:80px;width: 57px;margin-top:13px}

#cardrow{padding-top:10px;margin-left: 5px}

#imgsize{width:334px}

#imginfo{background-color:gainsboro}

#imgcard{height:263px;width:100%}

#imgcontwidth{width:337px}

#news{font-size: 35px;background-color:#083e54;color: white}

marquee{background-color:#d9d9ff;height:52px;padding-top:14px}

#rownews{margin-left:0px;margin-right:0px}

#commissionpad{padding-top: 10px}

#glyphnews{padding-left:25px}

#commissionrow{background-color:orange;text-align: center}

#imgsize{height:140px;width:138px}

#customers {

font-family: "Trebuchet MS", Arial, Helvetica, sans-serif;

border-collapse: collapse;

margin-top: 70px;

width: 100%;

}

#customers td, #customers th {

border: 1px solid #ddd;

padding: 8px;

}

#customers tr:nth-child(even){background-color: #f2f2f2;}

#customers tr:hover {background-color: #ddd;}

#customers th {

padding-top: 12px;

padding-bottom: 12px;

text-align: left;

background-color: #4CAF50;

color: white;

}

VoterReg.css

form {

position: relative;

z-index: 2;

padding: 11.5em 1em 2em 1em;

margin: 1em;

border: 1px solid rgba(211, 211, 211, .5);

border-radius: 1em;

}

h1 {

font-size: 2.5em;

font-weight: 500;

color: #fff;

margin-bottom: .35em;

}

div.form-wrapper {

position: relative;

width: 40em;

padding: 1em;

margin: 5em auto;

border-radius: 1.5em;

display: table;

text-align: center;

box-shadow: 15px 21px 62px -9px rgba(145,145,145,1);

}

header {

position: absolute;

width: 100%;

height: 11.5em;

padding: 1.25em 2em 1em 2em;

border-radius: 1em 1em 0 0;

top: 0; left: 0;

background: url('https://s3-us-west-2.amazonaws.com/s.cdpn.io/1239626/camera-1600x1000.jpg');

background-size: 100%;

background-position: bottom center;

}

input {

display: block;

width: 100%;

padding: .5em 1em;

margin: .5em auto;

border-radius: 2em;

font-weight: 300;

outline: none;

}

.btn, input.btn {

max-width: 15em;

display: inline-block;

border-radius: 9em;

padding: .5em 1em;

}

Homepage.css

.imgCar{

width: 100% !important;

height: 340px !important;

}