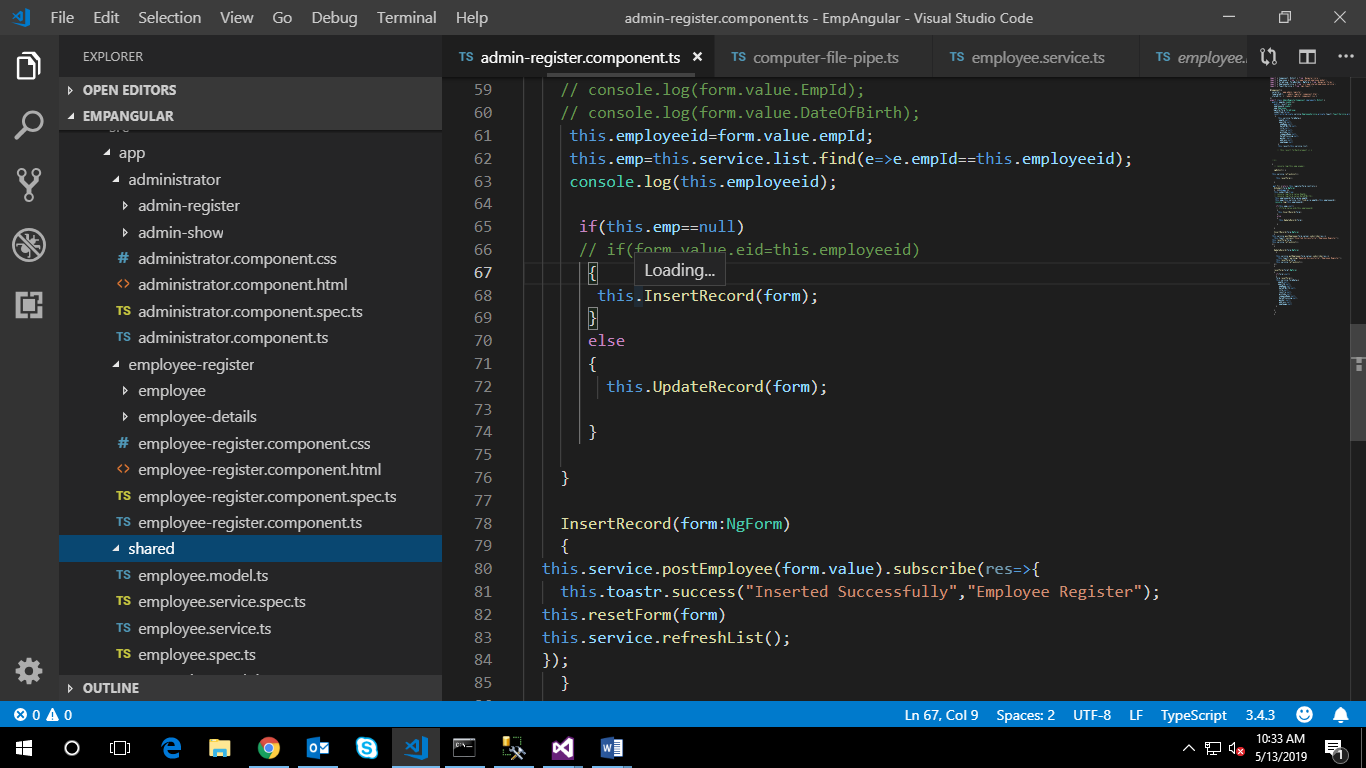
**Employee Management System**

**(Angular 7)**



Img1.Component Folder Details

Shared Folder:

Employee.Model.ts:-

export class Employee {

empId:number;

empName: string;

dateOfBirth: Date;

emailId: string;

gender: string;

address: string;

pinCode: string;

country: string;

city :string

companyName:string;

empImage:string;

DateOfJoining:Date;

}

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

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

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

import { Employee } from '../../app/shared/employee.model';

@Injectable({

providedIn: 'root'

})

export class EmployeeService {

formData:Employee;

list:Employee[];

readonly rootURL="http://localhost:50055/api"

constructor(private http:HttpClient) { }

postEmployee(formData:Employee)

{

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

}

refreshList():void

{

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

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

}

putEmployee(formData:Employee)

{

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

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

{

return this.http.put<Employee>(this.rootURL+'/Employees/'+formData.empId,formData,{headers,params});

}

}

deleteEmployee(id:number)

{

return this.http.delete(this.rootURL+'/Employees/'+id);

}

}

**Employee.Service.ts**

**Administrator Component**

* **Admin-register.**

1. **Admin-register.component.css**
2. **Admin-register.component.html**
3. **Admin-register.component.ts.**

* **Admin-show.**

1. **Admin-show.component.css**
2. **Admin-show.component.html**
3. **Admin-show.component.ts**

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

import { EmployeeService } from '../shared/employee.service';

import { ToastrService } from 'ngx-toastr';

import { Employee } from '../shared/employee.model';

@Component({

selector: 'app-administrator',

templateUrl: './administrator.component.html',

styleUrls: ['./administrator.component.css']

})

export class AdministratorComponent implements OnInit {

constructor(public service:EmployeeService,public toastr:ToastrService) { }

ngOnInit() {

this.service.refreshList();

}

Populateform(emp:Employee)

{

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

}

onDelete(id:number)

{

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

console.log(id);

this.service.deleteEmployee(id).subscribe(res=>

{

this.service.refreshList();

this.toastr.info("Deleted Successfully","Employee Register");

});

}

}

}

**Administrator.component.ts**

<div class="Jumbotron">

<div class="row">

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

<app-admin-register></app-admin-register>

</div>

<div class="col-md-8">

<app-admin-show></app-admin-show>

</div>

</div>

</div>

**Administrator.component.html**

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

import { Employee } from 'src/app/shared/employee.model';

import { FormGroup, FormBuilder, NgForm } from '@angular/forms';

import { EmployeeService } from 'src/app/shared/employee.service';

import { ToastrService } from 'ngx-toastr';

@Component({

selector: 'app-admin-register',

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

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

})

export class AdminRegisterComponent implements OnInit {

public empids:number[];

public result:any;

employeeid:number;

emp:Employee;

registerForm:FormGroup;

submitted=false;

constructor(private service:EmployeeService,private toastr:ToastrService,private formBuilder:FormBuilder)

{

this.service.formData={

empId:null,

emailId:null,

empName:null,

dateOfBirth:null,

city:null,

country:null,

pinCode:null,

companyName:null,

DateOfJoining:null,

gender:null,

address:null,

empImage:null

}

this.result=this.service.list;

// this.result.forEach(element => {

//});

}

// console.log(this.emp.ename);

ngOnInit() {

this.service.refreshList();

this.resetForm();

}

get f() {return this.registerForm.controls;}

OnSubmit(form:NgForm)

{ {{debugger}}

this.submitted=true;

// console.log(form.value.EmpId);

// console.log(form.value.DateOfBirth);

this.employeeid=form.value.empId;

this.emp=this.service.list.find(e=>e.empId==this.employeeid);

console.log(this.employeeid);

if(this.emp==null)

// if(form.value.eid=this.employeeid)

{

this.InsertRecord(form);

}

else

{

this.UpdateRecord(form);

}

}

InsertRecord(form:NgForm)

{

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

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

this.resetForm(form)

this.service.refreshList();

});

}

UpdateRecord(form:NgForm)

{

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

this.toastr.warning("Updated Successfully","Employee Register");

this.resetForm(form)

this.service.refreshList();

});

}

resetForm(form?:NgForm)

{

if(form!=null)

{

form.resetForm();

this.service.formData={

empId:null,

emailId:null,

empName:null,

dateOfBirth:null,

city:null,

country:null,

pinCode:null,

companyName:null,

DateOfJoining:null,

gender:null,

address:null,

empImage:null

}

}

}

}

**Admin-register.component.ts**

<script type="text/script">

$("#myel").on("input",function(){

if($(this).val().length>0){

$(this).addClass("full");

}

else{

$(this).removeClass("full");

}

});

</script>

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

<div class="container register-form">

<div class="form">

<div class="note">

<p>This is a Admin Page for Changes</p>

{{this.service.formData.Email}}

{{this.service.formData.City}}

</div>

<div class="form-content">

<div class="row">

<div class="col-md-6">

<div class="form-group">

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

</div>

<div class="form-group">

<input type="text" class="form-control" name="empName" placeholder="Name" required maxlength="15" minlength="3" [(ngModel)]="this.service.formData.empName" #empName="ngModel" id="EmpName">

</div>

<div class="form-group">

<input type="text" name="dateOfBirth" class="form-control datepicker" required [(ngModel)]="this.service.formData.dateOfBirth" #dateOfBirth="ngModel" onfocus="(this.type='date')" onblur="(this.type='text')" id="dob" placeholder=" Date of Birth">

</div>

<div class="form-group">

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

<option class="hidden" selected disabled>Please select City</option>

<option>Chennai</option>

<option>Bangalore</option>

<option>Pune</option>

<option>Gurgaon</option>

</select>

</div>

<div class="form-group">

<input type="number" class="form-control" name="pinCode" required maxlength="6" minlength="6" [(ngModel)]="this.service.formData.pinCode" #pinCode="ngModel" placeholder="Enter Pincode" >

</div>

<div class="form-group">

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

<option class="hidden" selected disabled>Company Name</option>

<option>Cognizant</option>

<option>Tcs</option>

<option>Infosys</option>

</select>

</div>

<div class="form-group">

<div class="maxl">

<label class="radio inline">

<input type="radio" name="Gender" value="male" [(ngModel)]="this.service.formData.Gender" #Gender="ngModel" checked>

<span> Male </span>

</label>

<label class="radio inline">

<input type="radio" name="Gender" [(ngModel)]="this.service.formData.Gender" #Gender="ngModel" value="Female">

<span>Female </span>

</label>

</div>

</div>

</div>

<div class="col-md-6">

<div class="form-group">

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

</div>

<div class="form-group">

<input type="email" class="form-control" name="emailId" placeholder="Your Email " [(ngModel)]="this.service.formData.emailId" #emailId="ngModel" >

</div>

<div class="form-group">

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

<option class="hidden" selected disabled>Please select Country</option>

<option>India</option>

</select>

</div>

<div class="form-group">

<input type="text" class="form-control" name="address" required placeholder="Enter your Address " [(ngModel)]="this.service.formData.address" #address="ngModel" >

</div>

<div class="form-group">

<input type="text" name="DateOfJoining" class="form-control datepicker" onfocus="(this.type='date')" onblur="(this.type='text')" id="dob" placeholder=" Date of Joining" required [(ngModel)]="this.service.formData.DateOfJoining" #DateOfJoining="ngModel" >

</div>

<div class="form-group">

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

</div>

</div>

</div>

<button type="submit" class="btnRegister">Submit </button>

</div>

</div>

</div>

</form>

**Admin-register.component.html**

.note

{

text-align: center;

height: 80px;

background: -webkit-linear-gradient(left, #0072ff, #8811c5);

color: #fff;

font-weight: bold;

line-height: 80px;

}

.form-content

{

padding: 5%;

border: 1px solid #ced4da;

margin-bottom: 2%;

}

.form-control{

border-radius:1.5rem;

}

.btnSubmit

{

border:none;

border-radius:1.5rem;

padding: 1%;

width: 20%;

cursor: pointer;

background: #0062cc;

color: #fff;

}

**Admin-register.component.css**

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

import { EmployeeService } from 'src/app/shared/employee.service';

import { ToastrService } from 'ngx-toastr';

import { Employee } from 'src/app/shared/employee.model';

@Component({

selector: 'app-admin-show',

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

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

})

export class AdminShowComponent implements OnInit {

constructor(public service:EmployeeService,public toastr:ToastrService) { }

ngOnInit() {

this.service.refreshList();

}

Populateform(emp:Employee)

{

{{debugger}}

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

console.log(emp.empName);

console.log(emp.address);

console.log(emp.city);

}

onDelete(id:number)

{

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

console.log(id);

this.service.deleteEmployee(id).subscribe(res=>

{

this.service.refreshList();

this.toastr.info("Deleted Successfully","Employee Register");

});

}

}

}

**Admin-show.component.ts**

<div class="row">

<div class="card" style="background-color:darkgray" \*ngFor="let emp of service.list"(click)=Populateform(emp) >

<div class="col-md-12" >

<img src="{{emp.empImage}}" style="width:250px;height:250px;border-radius:10px ;" />

<table class="table table-striped">

<tr>

<th>Name</th>

<td>{{emp.empName}}</td>

</tr>

<tr>

<th>Company Name</th>

<td>{{emp.companyName}}</td>

</tr>

<tr>

<th>Email-id</th>

<td>{{emp.emailId}}</td>

</tr>

<tr>

<th>Date Of Birth</th>

<td>{{emp.dateOfBirth|date:'yyyy/MM/dd'}}</td>

</tr>

<tr>

<th>Date Of Joining</th>

<td>{{emp.DateOfJoining|date:'yyyy/mm/dd'}}</td>

</tr>

<tr>

<th>Location</th>

<td>{{emp.city}} </td>

</tr>

<tr>

<th>Country</th>

<td>{{emp.country}} </td>

</tr>

<tr>

<th>Parmanent Address</th>

<td>{{emp.address}} </td>

</tr>

<tr>

<tfoot>

<tr>

<button (click)="onDelete(emp.empId)" class="btn btn-sm btn-outline-danger">Delete</button>

</tr>

</tfoot>

</table>

</div>

</div>

</div>

**Admin-show.component.html**

**Employee-register(Main Component)**

* **Employee**

1. **Employee.component.ts**
2. **Employee.component.html**
3. **Employee.component.css**

* **Employee-details**

1. **Employee-details.component.ts**
2. **Employee-details.html**
3. **Employee-details.component.css**

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

@Component({

selector: 'app-employee-register',

templateUrl: './employee-register.component.html',

styleUrls: ['./employee-register.component.css']

})

export class EmployeeRegisterComponent implements OnInit {

constructor() { }

ngOnInit() {

}

}

**Employee-register.component.ts**

<nav class="navbar navbar-light container-fluid"style="background-color:rgb(20, 60, 134)">

<form class="form-inline">

<button class="btn btn-outline-success " type="button" routerLink="Employee">Employee Registration</button>

<button class="btn btn-lg btn-outline-secondary" type="button" routerLink="Details">Employee Details</button>

<button class="btn btn-lg btn-outline-secondary" type="button" routerLink="Administrator">Admin</button>

</form>

</nav>

<router-outlet></router-outlet>

**Employee-register.component.html**

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

import { NgForm, NgFormSelectorWarning, FormGroup, Validators } from '@angular/forms';

import{FormBuilder,Validator} from '@angular/forms'

import { MockNgModuleResolver } from '@angular/compiler/testing';

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

import { ToastrService } from 'ngx-toastr';

import { EmployeeService } from '../../shared/employee.service';

import { Employee } from '../../shared/employee.model';

@Component({

selector: 'app-employee',

templateUrl: './employee.component.html',

styleUrls: ['./employee.component.css']

})

export class EmployeeComponent implements OnInit {

public empids:number[];

public result:any;

employeeid:number;

emp:Employee;

constructor(private service:EmployeeService,private toastr:ToastrService,private formBuilder:FormBuilder)

{

this.service.formData={

empId:null,

emailId:null,

empName:null,

dateOfBirth:null,

city:null,

country:null,

pinCode:null,

companyName:null,

DateOfJoining:null,

gender:null,

address:null,

empImage:null

}

this.result=this.service.list;

}

ngOnInit() {

this.resetForm();

this.service.refreshList();

}

OnSubmit(form:NgForm)

{ {{debugger}}

console.log(form.value.EmpId);

console.log(form.value.DateOfBirth);

this.employeeid=form.value.empId;

this.emp=this.service.list.find(e=>e.empId==this.employeeid);

console.log(this.employeeid);

if(this.emp==null)

{

this.InsertRecord(form);

}

else

{

this.UpdateRecord(form);

}

}

InsertRecord(form:NgForm)

{

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

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

this.resetForm(form)

this.service.refreshList();

});

}

UpdateRecord(form:NgForm)

{

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

this.toastr.warning("Updated Successfully","Employee Register");

this.resetForm(form)

this.service.refreshList();

});

}

resetForm(form?:NgForm)

{

if(form!=null)

{

form.resetForm();

this.service.formData={

empId:null,

emailId:null,

empName:null,

dateOfBirth:null,

city:null,

country:null,

pinCode:null,

companyName:null,

DateOfJoining:null,

gender:null,

address:null,

empImage:null

}

}

}

}

**Employee.component.ts**

<script type="text/script">

$("#myel").on("input",function(){

if($(this).val().length>0){

$(this).addClass("full");

}

else{

$(this).removeClass("full");

}

});

</script>

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

<div class="container-fluid register inputGroupContainer">

<div class="row">

<!-- <img src="/assets/Register.jpg"> -->

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

<img src="/assets/Register.jpg" class="img-circle" alt="Cinque Terre" height="150" style="border-radius:50%;width:60%">

<h3>Welcome</h3>

<p>Please Enter all the details correctly for Registration</p>

</div>

<div class="col-md-9 register-right">

<div class="tab-content" id="myTabContent">

<div class="tab-pane fade show active" id="home" role="tabpanel" aria-labelledby="home-tab">

<h3 class="register-heading">Register as a Employee</h3>

<div class="row register-form">

<div class="col-md-6">

<div class="form-group">

<input type="hidden" hidden class="form-control" name="EmpId" placeholder="id" required [(ngModel)]="this.service.formData.EmpId" #EmpId="ngModel" value="" />

</div>

<div class="form-group input-group">

<span class="input-group-addon"><i class="glyphicon glyphicon-user"></i></span>

<i [class.green-icon]="empName.valid" [class.red-icon]="empName.invalid && empName.touched"></i>

<input type="text" class="form-control" name="empName" placeholder="Full Name" required maxlength="15" minlength="3" [(ngModel)]="this.service.formData.empName" #empName="ngModel" value="" />

<div class="red-icon" \*ngIf="empName.invalid && empName.touched">Name is Required</div>

</div>

<div class="form-group">

<input type="date" name="dateOfBirth" class="form-control datepicker" required [(ngModel)]="this.service.formData.dateOfBirth" #dateOfBirth="ngModel" onfocus="(this.type='date')" onblur="(this.type='text')" id="dob" placeholder=" Date of Birth" value="">

<div class="red-icon" \*ngIf="dateOfBirth.invalid && dateOfBirth.touched">Date is Required</div>

</div>

<div class="form-group">

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

<option class="hidden" selected disabled>Please select City</option>

<option>Chennai</option>

<option>Bangalore</option>

<option>Pune</option>

<option>Gurgaon</option>

</select>

</div>

<div class="form-group">

<input type="number" class="form-control" name="pinCode" required maxlength="6" [(ngModel)]="this.service.formData.pinCode" #pinCode="ngModel" placeholder="Enter Pincode" value="" />

<div class="red-icon" \*ngIf="pinCode.invalid && pinCode.touched">Pincode is Required</div>

</div>

<div class="form-group">

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

<option class="hidden" selected disabled>Company Name</option>

<option>Cognizant</option>

<option>Tcs</option>

<option>Infosys</option>

</select>

</div>

<div class="form-group">

<div class="maxl">

<label class="radio inline">

<input type="radio" name="gender" value="male" [(ngModel)]="this.service.formData.gender" #gender="ngModel" checked>

<span> Male </span>

</label>

<label class="radio inline">

<input type="radio" name="gender" [(ngModel)]="this.service.formData.gender" #gender="ngModel" value="female">

<span>Female </span>

</label>

</div>

</div>

</div>

<div class="col-md-6">

<div class="form-group">

<input type="text" hidden class="form-control" name="EmpId" placeholder="id" required [(ngModel)]="this.service.formData.EmpId" #EmpId="ngModel" value="" />

</div>

<div class="form-group">

<input type="email" class="form-control" name="emailId" required placeholder="Your Email " [(ngModel)]="this.service.formData.emailId" #emailId="ngModel" value="" />

</div>

<div class="form-group">

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

<option class="hidden" selected disabled>Please select Country</option>

<option>India</option>

<option>United State</option>

</select>

</div>

<div class="form-group">

<input type="text" class="form-control" name="address" required placeholder="Parmenent Address " [(ngModel)]="this.service.formData.address" #address="ngModel" value="" />

</div>

<div class="form-group">

<input type="date" name="DateOfJoining" class="form-control datepicker" onfocus="(this.type='date')" onblur="(this.type='text')" id="dob" placeholder=" Date of Joining" required [(ngModel)]="this.service.formData.DateOfJoining" #DateOfJoining="ngModel" value="">

</div>

<div class="form-group">

<input type="text" name="empImage" class="form-control" placeholder="Image " required [(ngModel)]="this.service.formData.empImage" #empImage="ngModel" value="" />

</div>

<button type="submit" class="btnRegister" > Register </button>

</div>

</div>

</div>

</div>

</div>

</div>

</div>

</form>

<router-outlet></router-outlet>

**Employee.component.html**

.register{

background: -webkit-linear-gradient(left, #15156b, #1f5cbe);

margin-top: 3%;

padding: 3%;

}

.register-left{

text-align: center;

color: #fff;

margin-top: 4%;

}

.register-left input{

border: none;

border-radius: 1.5rem;

padding: 2%;

width: 60%;

background: #f8f9fa;

font-weight: bold;

color: #383d41;

margin-top: 30%;

margin-bottom: 3%;

cursor: pointer;

}

.register-right{

background: #f8f9fa;

border-top-left-radius: 10% 50%;

border-bottom-left-radius: 10% 50%;

}

.register-left img{

margin-top: 15%;

margin-bottom: 5%;

width: 25%;

-webkit-animation: mover 2s infinite alternate;

animation: mover 1s infinite alternate;

}

@-webkit-keyframes mover {

0% { transform: translateY(0); }

100% { transform: translateY(-20px); }

}

@keyframes mover {

0% { transform: translateY(0); }

100% { transform: translateY(-20px); }

}

.register-left p{

font-weight: lighter;

padding: 12%;

margin-top: -9%;

}

.register .register-form{

padding: 10%;

margin-top: 10%;

}

.btnRegister{

float: right;

margin-top: 10%;

border: none;

border-radius: 1.5rem;

padding: 2%;

background: #1f1e1e;

color: #fff;

font-weight: 600;

width: 50%;

cursor: pointer;

}

.register .nav-tabs{

margin-top: 3%;

border: none;

background: #cc0099;

border-radius: 1.5rem;

width: 28%;

float: right;

}

.register .nav-tabs .nav-link{

padding: 2%;

height: 34px;

font-weight: 600;

color: #fff;

border-top-right-radius: 1.5rem;

border-bottom-right-radius: 1.5rem;

}

.register .nav-tabs .nav-link:hover{

border: none;

}

.register .nav-tabs .nav-link.active{

width: 100px;

color: #cc0099;

border: 2px solid #cc0099;

border-top-left-radius: 1.5rem;

border-bottom-left-radius: 1.5rem;

}

.register-heading{

text-align: center;

margin-top: 8%;

margin-bottom: -15%;

color: #495057;

}

input[type=date] {

text-align: right;

}

input[type="date"]:before {

color: lightgrey;

content: attr(placeholder) !important;

margin-right: 0.5em;

}

**Employee.component.css**

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

import { EmployeeService } from '../../shared/employee.service';

import { ToastrService } from 'ngx-toastr';

import { Employee } from '../../shared/employee.model';

@Component({

selector: 'app-employee-details',

templateUrl: './employee-details.component.html',

styleUrls: ['./employee-details.component.css']

})

export class EmployeeDetailsComponent implements OnInit {

constructor(public service:EmployeeService,public toastr:ToastrService) { }

ngOnInit() {

this.service.refreshList();

}

Populateform(emp:Employee)

{

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

}

onDelete(id:number)

{

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

console.log(id);

this.service.deleteEmployee(id).subscribe(res=>

{

this.service.refreshList();

this.toastr.info("Deleted Successfully","Employee Register");

});

}

}

}

// @Pipe({

// name:'companyfilterPipe'

// })

// export class companyfilterPipe implements PipeTransform{

// transform( list:Employee[],companyname:string)

// {

// if(!list||!companyname)

// {

// return list;

// }

// return list.filter((x=>x.companyName.toLowerCase().indexOf(companyname.toLowerCase())!==-1));

// }

// }

// @Pipe({

// name:'CountryfilterPipe'

// })

// export class CountryfilterPipe implements PipeTransform{

// transform( list:Employee[],Country:string)

// {

// if(!list|| !Country)

// {

// return list;

// }

// return list.filter((x=>x.country.toLowerCase().indexOf(Country.toLowerCase())!==-1));

// }

// }

// @Pipe({

// name:'CityfilterPipe'

// })

// export class CityfilterPipe implements PipeTransform{

// transform( list:Employee[],City:string)

// {

// if(!list|| !City)

// {

// return list;

// }

// return list.filter((x=>x.city.toLowerCase().indexOf(City.toLowerCase())!==-1));

// }

// }

**Employee-details.component.ts**

<div class="container-fluid" style="background-color:#ca9c9c" >

<div class="row">

<h3>Employee Details</h3>

</div>

<br/>

<div class="row">

<div class="active-pink-3 active-pink-4 mb-4 col-md-4">

<input class="form-control" type="text" name="search" [(ngModel)]="companyname" placeholder="Enter the Company Name" aria-label="Search">

</div>

<div class="active-purple-3 active-purple-4 mb-4 col-md-4">

<input class="form-control" type="text" [(ngModel)]="Country" placeholder="Enter the Country Name" aria-label="Search">

</div>

<div class="active-cyan-3 active-cyan-4 mb-4 col-md-4">

<input class="form-control" type="text" [(ngModel)]="City" placeholder="Enter the City Name" aria-label="Search">

</div>

</div>

<div class="row" >

<div class="col-md-4" \*ngFor="let emp of service.list|companyfilterPipe:companyname|countryfilterPipe:Country|cityfilterPipe:City">

<div class="panel panel-default">

<div class="panel-heading"> </div>

<div class="panel-body">

<div class="box box-info">

<div class="box-body">

<div class="col-sm-6">

<div >

<img class="img-circle img-responsive" style="margin-left: 90px;" alt="User Pic" src={{emp.empImage}} id="profile-image1" >

</div>

<br>

</div>

<div class="col-sm-6">

<h4 (click)=Populateform(emp)>{{emp.empName}}</h4>

<span><h5>{{emp.companyName}}</h5></span>

</div>

<div class="clearfix"></div>

<hr style="margin:5px 0 5px 0;">

<table class="table table-striped">

<tr>

<th>Emailid</th>

<td>{{emp.emailId}} </td>

</tr>

<tr>

<th>Date Of Birth</th>

<td>{{emp.dateOfBirth|date:'yyyy/MM/dd'}}</td>

</tr>

<tr>

<th>Date Of Joining</th>

<td>{{emp.DateOfJoining|date:'yyyy/MM/dd'}}</td>

</tr>

<tr>

<th>Location</th>

<td>{{emp.city}} </td>

</tr>

<tr>

<th>Parmanent Address</th>

<td>{{emp.address}} </td>

</tr>

<tr>

<th>Country</th>

<td>{{emp.country}} </td>

</tr>

</table>

</div>

</div>

</div>

</div>

<div class="col-md-1">

<button (click)="onDelete(emp.empId)" class="btn btn-sm btn-outline-danger">Delete</button>

</div>

</div>

</div>

</div>

**Employee-details.component.html**

input.hidden {

position: absolute;

left: -9999px;

}

#profile-image1 {

width: 200px;

height: 200px;

    border:2px solid #03b1ce ;}

    .tital{ font-size:16px; font-weight:500;}

.bot-border{ border-bottom:1px #f8f8f8 solid; margin:5px 0 5px 0}

h3{

margin-left:40%;color:rgb(20, 60, 134)

}

h4{

color:#234646;

}

h5{

color:#00b10f;

}

**Employee-details.component.css**

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

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

const routes: Routes = [];

@NgModule({

imports: [RouterModule.forRoot(routes)],

exports: [RouterModule]

})

export class AppRoutingModule { }

**app-routing.module.ts**

<app-employee-register></app-employee-register>

**App.component.html**

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

@Component({

selector: 'app-root',

templateUrl: './app.component.html',

styleUrls: ['./app.component.css']

})

export class AppComponent {

title = 'CompanyEmployee';

}

**App.component.ts**

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

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

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

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

import {ToastrModule} from 'ngx-toastr';

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

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

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

import { EmployeeService } from './shared/employee.service';

import { EmployeeRegisterComponent } from './employee-register/employee-register.component';

import { cityfilterPipe, countryfilterPipe, companyfilterPipe } from './employee-register/employee-details/filterPipe';

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

import { AdministratorComponent } from './administrator/administrator.component';

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

import { AdminRegisterComponent } from './administrator/admin-register/admin-register.component';

import { EmployeeDetailsComponent } from './employee-register/employee-details/employee-details.component';

import { EmployeeComponent } from './employee-register/employee/employee.component';

@NgModule({

declarations: [

AppComponent,

EmployeeRegisterComponent,

companyfilterPipe,

countryfilterPipe,

cityfilterPipe,

AdministratorComponent,

EmployeeDetailsComponent,

AdminRegisterComponent,

AdminShowComponent,

EmployeeComponent

],

imports: [

BrowserModule,

AppRoutingModule,

FormsModule,

ReactiveFormsModule,

HttpClientModule,

BrowserAnimationsModule,

ToastrModule.forRoot(),

RouterModule.forRoot([ { path:'Employee',component:EmployeeComponent },

{ path:'Details',component:EmployeeDetailsComponent},

{path: 'Administrator', component:AdministratorComponent}])

],

providers: [HttpClientModule, EmployeeService],

bootstrap: [AppComponent]

})

export class AppModule { }

**app.module.ts**