**App Module.ts**

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

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

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

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

import {HttpClientModule} 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 { ProjectComponent } from './project/project.component';

import { UserService } from './shared/user.service';

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

import { cityfilterPipe, domainfilterPipe } from './project/show/filterPipe';

import { ShowComponent } from './project/show/show.component';

import { CreateComponent } from './project/create/create.component';

import { AdminComponent } from './project/admin/admin.component';

import { AdminCreateComponent } from './project/admin/admin-create/admin-create.component';

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

import { HomePageComponent } from './project/home-page/home-page.component';

@NgModule({

declarations: [

AppComponent,

ProjectComponent,

AdminComponent,

AdminCreateComponent,

AdminShowComponent,

CreateComponent,

ShowComponent,

cityfilterPipe,

domainfilterPipe,

HomePageComponent

],

imports: [

BrowserModule,

AppRoutingModule,

FormsModule,

ReactiveFormsModule,

HttpClientModule,

CommonModule,

BrowserAnimationsModule,

ToastrModule.forRoot(),

RouterModule.forRoot([ { path:'show', component: ShowComponent },

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

{path:'adminlink',component:AdminComponent}

,{path:'home',component:HomePageComponent}

])

],

providers: [UserService],

bootstrap: [AppComponent]

})

export class AppModule { }

**App-Component.html**

<app-project></app-project>

**Project-Component.html**

<div class="Jumbotron ">

<nav class="navbar navbar-inverse ">

<div class="row">

<div class="col-sm-2">

<a routerLink="home"><img src="/assets/image/cognizant.jpg" style="height:40px;width:110px;border-radius:10px;padding-top:5px" alt=""/></a>

</div>

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

<li style="text-align:left"><a routerLink="Register"><img src="/assets/image/reg.jpg" style="width:120px;height:40px;border-radius:5px;padding-top:3px"> </a></li>

</div>

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

<li style="text-align:left;"><a routerLink="show"><img src="/assets/image/detail.jpg" style="width:100px;height:35px;border-radius:5px;padding-top:7px"></a></li>

</div>

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

<li style="text-align:right;padding-right:15px"><a routerLink="adminlink"><img src="/assets/image/adminlogo.jpg" style="width:55px;height:45px;border-radius:10px;padding-top:3px"></a></li>

</div>

</div>

</nav>

<router-outlet></router-outlet>

</div>

**Homepage.html**

<div class="container">

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

<!-- Indicators -->

<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>

<!-- Wrapper for slides -->

<div class="carousel-inner">

<div class="item active">

<img src="/assets/image/z.jpg" alt="Los Angeles" style="width:100%;height: 480px;">

</div>

<div class="item">

<img src="/assets/image/z2.jpg" alt="Chicago" style="width:100%;height: 480px;">

</div>

<div class="item">

<img src="/assets/image/z3.jpg" alt="New York" style="width:100%;height: 480px;">

</div>

</div>

<!-- Left and right controls -->

<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>

**Create.html**

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

<div class="container register">

<div class="row">

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

<img src="/assets/image/cognizant.jpg" style="height:60px;width:150px;border-radius:20px" alt=""/>

<h3>Welcome</h3>

<p> Complete your Profile! </p>

</div>

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

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

<div class="row register-form">

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

<div class="form-group">

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

<div

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

</div>

<div class="form-group">

<input type="date" class="form-control" placeholder="Date of Birth " name="dob" [(ngModel)]="this.service.formData.dob" #dob="ngModel" required />

<div

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

</div>

<div class="form-group">

<input type="text" minlength="10" maxlength="10" class="form-control" placeholder="Your Phone \*" name="Contact" [(ngModel)]="this.service.formData.Contact" #Contact="ngModel" required />

<div

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

</div>

<div class="form-group">

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

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

</div>

</div>

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

<div class="form-group">

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

<div

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

</div>

<div class="form-group">

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

<div

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

</div>

<div class="form-group">

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

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

</div>

<div class="form-group">

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

<div class="red-icon" \*ngIf="BU.invalid && BU.touched">Business Unit is Required</div>

</div>

</div>

<div class="row">

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

<div class="form-group" style="margin-left:310px">

<button type="submit" class="btn btn-md btn-success" [disabled]="form.invalid"> Register </button>

</div>

</div>

</div>

</div>

</div>

</div>

</div>

</form>

**Create.ts**

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

import { UserService } from '../../Shared/user.service';

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

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

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

import { User } from '../../shared/User.model';

import { ToastrService } from 'ngx-toastr';

@Component({

selector: 'app-create',

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

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

})

export class CreateComponent implements OnInit {

public empids:number[];

public result:any;

user\_id:number;

use:User;

constructor(private service:UserService,private toastr:ToastrService)

{

this.service.formData={

Userid:null,

Username:null,

Emailid:null,

dob:null,

Contact:null,

City:null,

BU:null,

Domain:null,

Image:null,

}

this.result=this.service.list;

}

ngOnInit() {

this.resetForm();

}

OnSubmit(form:NgForm)

{ {debugger}

this.user\_id=form.value.Userid;

this.use=this.service.list.find(e=>e.Userid==this.user\_id);

console.log(this.user\_id);

if(this.use==null)

{

this.InsertRecord(form);

}

else

{

this.UpdateRecord(form);

}

}

InsertRecord(form:NgForm)

{

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

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

this.resetForm(form)

this.service.refreshList();

});

}

UpdateRecord(form:NgForm)

{

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

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

this.resetForm(form)

this.service.refreshList();

});

}

resetForm(form?:NgForm)

{

if(form!=null)

{

form.resetForm();

this.service.formData={

Userid:null,

Username:null,

Emailid:null,

dob:null,

Contact:null,

City:null,

BU:null,

Domain:null,

Image:null,

}

}

}

}

**Show.html**

<div class="container">

<div class="row">

<div class="col-md-2">

<label style="font-size:20px">Enter the City:</label>

</div>

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

<input type="text" class="form-control" id=empCity [(ngModel)]="empCity" placeholder=" City ">

</div>

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

<label style="font-size:20px">Enter the Domain:</label>

</div>

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

<input type="text" class="form-control" id=empDomain [(ngModel)]="empDomain" placeholder=" Domain ">

</div>

</div>

<br/>

<br/>

<div class="row">

<div class="card" \*ngFor="let emp of service.list|cityfilterPipe:empCity|domainfilterPipe:empDomain" >

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

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

<h4><b>{{emp.Username}}</b></h4>

<p>{{emp.Emailid}}</p>

<p><b>{{emp.BU}}</b>&nbsp;&nbsp;&nbsp;&nbsp;{{emp.Domain}}</p>

<p><b>{{emp.City}}</b></p>

<p>{{emp.Contact}}</p>

</div>

</div>

</div>

</div>

**Show.ts**

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

import { UserService } from '../../Shared/user.service';

import {User} from '../../shared/User.model';

import { ToastrService } from 'ngx-toastr';

@Component({

selector: 'app-show',

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

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

})

export class ShowComponent implements OnInit {

constructor(public service:UserService,public toastr:ToastrService) {

}

ngOnInit() {

this.service.refreshList();

}

Populateform(use:User)

{

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

}

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 Details");

});

}

}

}

**Admin.html**

<div class="container">

<h1 class="dipslay-4.text-center"> </h1>

<hr>

<div class="row">

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

<app-admin-create></app-admin-create>

</div>

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

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

</div>

</div>

</div>

**AdminCreate.html**

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

<div style="margin-left:20px;margin-top:20px;background-color:aliceblue">

<div class="form-group">

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

</div>

<div class="form-group">

<label> Full Name:</label>

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

</div>

<div class="form-group">

<label> Email-id:</label>

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

</div>

<div class="form-group">

<label> Contact:</label>

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

</div>

<div class="form-group">

<label> Date Of Birth:</label>

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

</div>

<div class="form-group">

<label> City:</label>

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

</div>

<div class="form-group">

<label> Domain:</label>

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

</div>

<div class="form-group">

<label> Image:</label>

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

</div>

<div class="form-group">

<label> BU:</label>

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

</div>

<div class="form-group">

<button type="submit" class="btn btn-lg btn-block"> SUBMIT </button>

</div>

</div>

</form>

**AdminCreate.ts**

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

import { UserService } from '../../../Shared/user.service';

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

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

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

import { User } from '../../../shared/User.model';

import { ToastrService } from 'ngx-toastr';

@Component({

selector: 'app-admin-create',

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

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

})

export class AdminCreateComponent implements OnInit {

public empids:number[];

public result:any;

id:number;

use:User;

constructor(private service:UserService,private toastr:ToastrService)

{

this.service.formData={

Userid:null,

Username:null,

Emailid:null,

dob:null,

Contact:null,

City:null,

BU:null,

Domain:null,

Image:null,

}

this.result=this.service.list;

}

ngOnInit() {

this.resetForm();

}

OnSubmit(form:NgForm)

{ {debugger}

this.id=form.value.Userid;

this.use=this.service.list.find(e=>e.Userid==this.id);

console.log(this.id);

if(this.use==null)

{

this.InsertRecord(form);

}

else

{

this.UpdateRecord(form);

}

}

InsertRecord(form:NgForm)

{

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

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

this.resetForm(form)

this.service.refreshList();

});

}

UpdateRecord(form:NgForm)

{

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

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

this.resetForm(form)

this.service.refreshList();

});

}

resetForm(form?:NgForm)

{

if(form!=null)

{

form.resetForm();

this.service.formData={

Userid:null,

Username:null,

Emailid:null,

dob:null,

Contact:null,

City:null,

BU:null,

Domain:null,

Image:null,

}

}

}

}

**Adminshow.html**

<div class="container">

<div class="card" \*ngFor="let emp of service.list" >

<div class="row">

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

<img src="{{'.'+emp.Image}}" (click)=Populateform(emp) style="width:250px;height:250px;border-radius:10px ;" >

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

<p (click)=Populateform(emp)>{{emp.Emailid}}</p>

<p (click)=Populateform(emp)><b>{{emp.BU}}</b>&nbsp;&nbsp;&nbsp;&nbsp;{{emp.Domain}}</p>

<p (click)=Populateform(emp)><b>{{emp.City}}</b></p>

<p (click)=Populateform(emp)>{{emp.Contact}}</p>

</div>

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

<button (click)="onDelete(emp.Userid)" class="btn btn-sm btn-danger">X</button>

</div>

</div>

</div>

</div>

**AdminShow.ts**

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

import { UserService } from '../../../Shared/user.service';

import {User} from '../../../Shared/User.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(public service:UserService,public toastr:ToastrService) {

}

name:string;

ngOnInit() {

this.service.refreshList();

}

Populateform(use:User)

{

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

}

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.warning("Deleted Successfully","Employee Details");

});

}

}

}

**User.Model.ts**

export class User {

Userid:number;

Username:string;

Emailid:string;

dob:Date;

Contact:string;

City:string;

BU:string;

Domain:string;

Image:string;

}

**User.Service.ts**

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

import { User } from '../Shared/User.model';

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

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

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

@Injectable({

providedIn: 'root'

})

export class UserService {

formData:User;

list:User[];

readonly rootURL="http://localhost:59395/api"

constructor(private http:HttpClient) { }

PostEmployeeData(formData:User)

{

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

}

refreshList():void

{

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

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

}

PutEmployeeData(formData:User)

{

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

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

{

return this.http.put<User>(this.rootURL+'/EmployeeDatas/'+formData.Userid,formData,{headers,params});

}

}

deleteEmployee(id:number)

{

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

}

}

**FilterPipe.ts**

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

import { User } from '../Shared/User.model';

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

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

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

@Injectable({

providedIn: 'root'

})

export class UserService {

formData:User;

list:User[];

readonly rootURL="http://localhost:59395/api"

constructor(private http:HttpClient) { }

PostEmployeeData(formData:User)

{

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

}

refreshList():void

{

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

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

}

PutEmployeeData(formData:User)

{

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

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

{

return this.http.put<User>(this.rootURL+'/EmployeeDatas/'+formData.Userid,formData,{headers,params});

}

}

deleteEmployee(id:number)

{

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

}

}