Computer Management Store

(Visual Studio Code)

Admin.component.css

.bgi{

background-image: url("../../../assets/lightbg.jpg");

/\*background-repeat: no-repeat;\*/

background-position: center;

width: 100%;

}

Admin.component.html

<div class="container-fluid bgi">

<div class="row">

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

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

</div>

</div>

<div class="row">

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

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

</div>

</div>

</div>

Admin.component.ts

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

@Component({

selector: 'app-admin',

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

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

})

export class AdminComponent implements OnInit {

constructor() { }

ngOnInit() {

}

}

Admin-create.component.css

body{

background-color: #525252;

}

.centered-form{

    margin-top: 60px;

}

.centered-form .panel{

    background: rgba(255, 255, 255, 0.8);

    box-shadow: rgba(0, 0, 0, 0.3) 20px 20px 20px;

}

.bgi{

background-image: url("../../../assets/comphd.jpg");

background-repeat: no-repeat;

background-position: center;

    }

    .control-col{

        color: aqua;

    }

    .panel-heading{

        color:seagreen;

    }

Admin-create.component.html

<div class="container-fluid bgi">

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

<div class="container">

<div class="row centered-form">

<div class="col-xs-12 col-sm-8 col-md-4 col-sm-offset-2 col-md-offset-4">

<div class="panel panel-default">

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

<h3 class="panel-title">Please Enter the Computer Details</h3>

</div>

<div class="panel-body" style="background-color: powderblue">

<!-- <form role="form"> -->

<div class="row">

<div class="form-group">

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

<div class="validation-error" \*ngIf="Computer\_Id.invalid && Computer\_Id.touched">This field is required</div>

</div>

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

<div class="form-group">

<input name="Company\_Name" id="Company\_Name" [(ngModel)]="this.service.formData.Company\_Name" class="form-control" #Company\_Name="ngModel" placeholder="Company Name" required>

<div class="validation-error" \*ngIf="Company\_Name.invalid && Company\_Name.touched">This field is required</div>

</div>

</div>

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

<div class="form-group">

<input name="Comp\_RAM" id="Comp\_RAM" [(ngModel)]="this.service.formData.Comp\_RAM" class="form-control" placeholder="RAM" #Comp\_RAM="ngModel" required>

<div class="validation-error" \*ngIf="Comp\_RAM.invalid && Comp\_RAM.touched">This field is required</div>

</div>

</div>

</div>

<div class="form-group">

<input name="Comp\_Processor" id="Comp\_Processor" [(ngModel)]="this.service.formData.Comp\_Processor" #Comp\_Processor="ngModel" class="form-control" placeholder="Processor" required>

<div class="validation-error" \*ngIf="Comp\_Processor.invalid && Comp\_Processor.touched">This field is required</div>

</div>

<div class="row">

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

<div class="form-group">

<input name="Comp\_Price" id="Comp\_Price" [(ngModel)]="this.service.formData.Comp\_Price" class="form-control" placeholder="Price" #Comp\_Price="ngModel" required>

<div class="validation-error" \*ngIf="Comp\_Price.invalid && Comp\_Price.touched">This field is required</div>

</div>

</div>

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

<div class="form-group">

<input name="Comp\_Images" id="Comp\_Images " [(ngModel)]="this.service.formData.Comp\_Images" class="form-control" placeholder="Image Path" #Comp\_Images="ngModel" required>

</div>

</div>

</div>

<div class="form-group">

<input type="submit" [disabled]="form.invalid"class="btn btn-lg btn-block" value="Submit">

</div>

<!-- </form> -->

</div>

</div>

</div>

</div>

</div>

</form>

</div>

Admin-create.component.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={

Computer\_Id:null,

Company\_Name:null,

Comp\_RAM:null,

Comp\_Processor:null,

Comp\_Price:null,

Comp\_Images:null,

}

this.result=this.service.list;

}

ngOnInit() {

this.resetForm();

// this.refreshList();

}

OnSubmit(form:NgForm)

{

// {debugger}

// console.log(form.value.Computer\_Id);

// console.log(form.value.Company\_Name);

// console.log(form.value.Comp\_RAM);

// console.log(form.value.Comp\_Processor);

// console.log(form.value.Comp\_Price);

// console.log(form.value.Comp\_Images);

// this.InsertRecord(form);

{debugger}

this.id=form.value.Computer\_Id;

this.use=this.service.list.find(e=>e.Computer\_Id==this.id);

console.log(this.id);

if(this.use==null)

{

this.InsertRecord(form);

}

else

{

this.UpdateRecord(form);

}

}

InsertRecord(form:NgForm)

{

// alert("here");

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

this.toastr.success("Inserted Successfully");

this.resetForm(form)

this.service.refreshList();

});

}

UpdateRecord(form:NgForm)

{

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

this.toastr.info("Updated Successfully");

this.resetForm(form)

this.service.refreshList();

});

}

resetForm(form?:NgForm)

{

if(form!=null)

form.resetForm();

this.service.formData={

Computer\_Id:null,

Company\_Name:null,

Comp\_RAM:null,

Comp\_Processor:null,

Comp\_Price:null,

Comp\_Images:null,

}

}

}

Admin-show.component.css

.th{

padding-left: 460px;

}

.thead{

color:blue;

background-color: lightgreen

}

Admin-show.component.html

<div class="container-fluid">

<table class="table table-bordered">

<thead class="thead">

<tr>

<th class="th">Image</th>

<th>Name</th>

<th>RAM</th>

<th>Processor</th>

<th>Price</th>

<th>Option</th>

</tr>

</thead>

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

<img src ="{{use.Comp\_Images}}" style="height:250px;width:50%;margin-left:100px">

<td (click)=PopulateForm(use)>{{use.Company\_Name}}</td>

<td (click)=PopulateForm(use)>{{use.Comp\_RAM}}</td>

<td (click)=PopulateForm(use)>{{use.Comp\_Processor}}</td>

<td (click)=PopulateForm(use)>{{use.Comp\_Price}}</td>

<td><button (click)="onDelete(use.Computer\_Id)" class="btn btn-danger">Delete</button></td>

</tr>

</table>

</div>

Admin-show.component.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)

{

}

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.deleteComputer(id).subscribe(res=>

{

this.service.refreshList();

this.toastr.info("Deleted Successfully");

});

}

}

}

Show

pipeFilter.ts

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

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

@Pipe({

name:'companyNamefilterPipe'

})

export class companyNamefilterPipe implements PipeTransform{

transform(ComputerDetails:User[],Company\_Name:string)

{

if(!ComputerDetails||!Company\_Name)

{

return ComputerDetails;

}

return ComputerDetails.filter((x=>x.Company\_Name.toLowerCase().indexOf(Company\_Name.toLowerCase())!==-1));

}

}

@Pipe({

name:'RAMfilterPipe'

})

export class RAMfilterPipe implements PipeTransform{

transform(ComputerDetails:User[],Comp\_RAM:string)

{

if(!ComputerDetails||!Comp\_RAM)

{

return ComputerDetails;

}

return ComputerDetails.filter((x=>x.Comp\_RAM.toLowerCase().indexOf(Comp\_RAM.toLowerCase())!==-1));

}

}

Show.component.css

.bgis{

background-image: url("../../../assets/lightbg.jpg");

/\*background-repeat: no-repeat;\*/

background-position: center;

width: 100%;

}

/\* .bold {

font-weight:bold;

} \*/

.form-control{

width:25%;

}

.thead{

color:blue;

background-color: lightgreen

}

.img{

height:250px;

width:50%;

margin-left:100px

}

.th{

padding-left: 460px;

}

Show.component.html

<div class="container-fluid bgis">

<div class="row">

<br>

<div>

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

<label>

Enter the Company Name of the Computer

</label>

</div>

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

<input class="form-control form-control" placeholder="Company Name" [(ngModel)]="Company\_Name" id=Company\_Name>

</div>

</div>

</div>

<div class="row">

<br>

<div>

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

<label>

Enter the RAM of the Computer

</label>

</div>

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

<input class="form-control form-control" placeholder="RAM" [(ngModel)]="Comp\_RAM" id=Comp\_RAM >

</div>

</div>

</div>

<br>

<table class="table table-bordered">

<thead class="thead">

<tr>

<!-- <div class="row"> -->

<th class="th">Image</th>

<th>Name</th>

<th>RAM</th>

<th>Processor</th>

<th>Price</th>

<th>Option</th>

<!-- </div> -->

</tr>

</thead>

<tbody>

<tr \*ngFor="let use of service.list|companyNamefilterPipe:Company\_Name|RAMfilterPipe:Comp\_RAM">

<!-- <div class="row"> -->

<td><img src ="{{use.Comp\_Images}}" style="height:250px;width:50%;margin-left:100px"></td>

<td (click)=PopulateForm(use)>{{use.Company\_Name}}</td>

<td (click)=PopulateForm(use)>{{use.Comp\_RAM}}</td>

<td (click)=PopulateForm(use)>{{use.Comp\_Processor}}</td>

<td (click)=PopulateForm(use)>{{use.Comp\_Price}}</td>

<td><button (click)="onDelete(use.Computer\_Id)" class="btn btn-danger">Delete</button></td>

<!-- </div> -->

</tr>

</tbody>

</table>

</div>

Show.component.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.deleteComputer(id).subscribe(res=>

{

this.service.refreshList();

this.toastr.info("Deleted Successfully");

});

}

}

}

Project.component.css

.navStyle{

text-decoration:none;

color:brown;

font-size:25px;

list-style: none;

padding : 3px 10% 0;

}

.navbar{

margin-bottom: 0;

}

.router{

width:41%;

height:73px;

border-radius:10px;

}

Project.component.html

<div>

<!-- <h1 class="dipslay-4.text-center" style="color:black;text-align: center">Admin</h1>

<hr>

<div class="row">

<div class="col-md-5">

<app-create></app-create>

</div>

<div class="col-md-7">

<app-show></app-show>

</div>

</div> -->

<nav class="navbar navbar-inverse">

<div class="row">

<div class="col-xs-4 navStyle">

<li><a routerLink="admin" ><img src="/assets/AdminIm.jpg" style="width:41%;

height:73px;

border-radius:10px;"></a></li>

</div>

<div class="col-xs-4" style="font-size:30px;color:white;padding-left:70px">

Computer <br>Management

</div>

<div class="col-xs-4 navStyle">

<li style="float: right"><a routerLink="Show"><img src="/assets/User.jpg" style="width:41%;

height:73px;

border-radius:10px;"></a></li>

</div>

</div>

</nav>

<router-outlet></router-outlet>

</div>

Project.component.ts

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

@Component({

selector: 'app-project',

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

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

})

export class ProjectComponent implements OnInit {

constructor() { }

ngOnInit() {

}

}

Shared

User.model.ts

export class User {

Computer\_Id:number;

Company\_Name:string;

Comp\_RAM:string;

Comp\_Processor:string;

Comp\_Price:number;

Comp\_Images: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 {

// constructor() { }

formData:User;

list:User[];

readonly rootURL="http://localhost:55058/api"

constructor(private http:HttpClient) { }

postComputer(formData:User)

{

return this.http.post(this.rootURL+'/Computer\_Details',formData);

}

refreshList():void

{

this.http.get(this.rootURL+'/Computer\_Details')

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

}

putComputer(formData:User)

{

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

const params = new HttpParams().set('ID', formData.Computer\_Id.toString());

{

return this.http.put<User>(this.rootURL+'/Computer\_Details/'+formData.Computer\_Id,formData,{headers,params});

}

}

deleteComputer(id:number)

{

return this.http.delete(this.rootURL+'/Computer\_Details/'+id);

}

}

app-routing.module.ts

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

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

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

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

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

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

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

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

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

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

// const routes: Routes = [

// { path: 'Create', component: CreateComponent },

// {path:'Show',component:ShowComponent},

// {path:'admin',component:AdminComponent}

// ];

// @NgModule({

// declarations: [

// CreateComponent,ShowComponent,AdminCreateComponent,AdminShowComponent,AdminComponent],

// imports: [BrowserModule,

// CommonModule,FormsModule,

// RouterModule.forRoot(routes)],

// exports: [RouterModule]

// })

// export class AppRoutingModule { }

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.component.html

<!--The content below is only a placeholder and can be replaced.-->

<!-- <div style="text-align:center">

<h1>

Welcome to {{ title }}!

</h1>

<img width="300" alt="Angular Logo" src="">

</div>

<h2>Here are some links to help you start: </h2>

<ul>

<li>

<h2><a target="\_blank" rel="noopener" href="https://angular.io/tutorial">Tour of Heroes</a></h2>

</li>

<li>

<h2><a target="\_blank" rel="noopener" href="https://angular.io/cli">CLI Documentation</a></h2>

</li>

<li>

<h2><a target="\_blank" rel="noopener" href="https://blog.angular.io/">Angular blog</a></h2>

</li>

</ul>

<router-outlet></router-outlet> -->

<app-project></app-project>

App.component.ts

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

@Component({

selector: 'app-root',

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

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

})

export class AppComponent {

title = 'AngularComputerProject';

}