**ANGULAR PROJECT**

RECIPEBOX

(SALONI SHARMA 755006)

**ABSTRACT**

This Web Application is on Angular using Web API , All the images are coming from WEB API

This Application consist Components Like Home, navbar , details ,services(for web api), Admin,

Footer , search , filters.

We are adding services for the connection pf web api to this project,

This project have well defines homepage using cards and carousels with a good footer.

We have an admin page for performing any CRUDs.

We have install bootstrap,css,ngx2Search etc. packages .

Functionalities

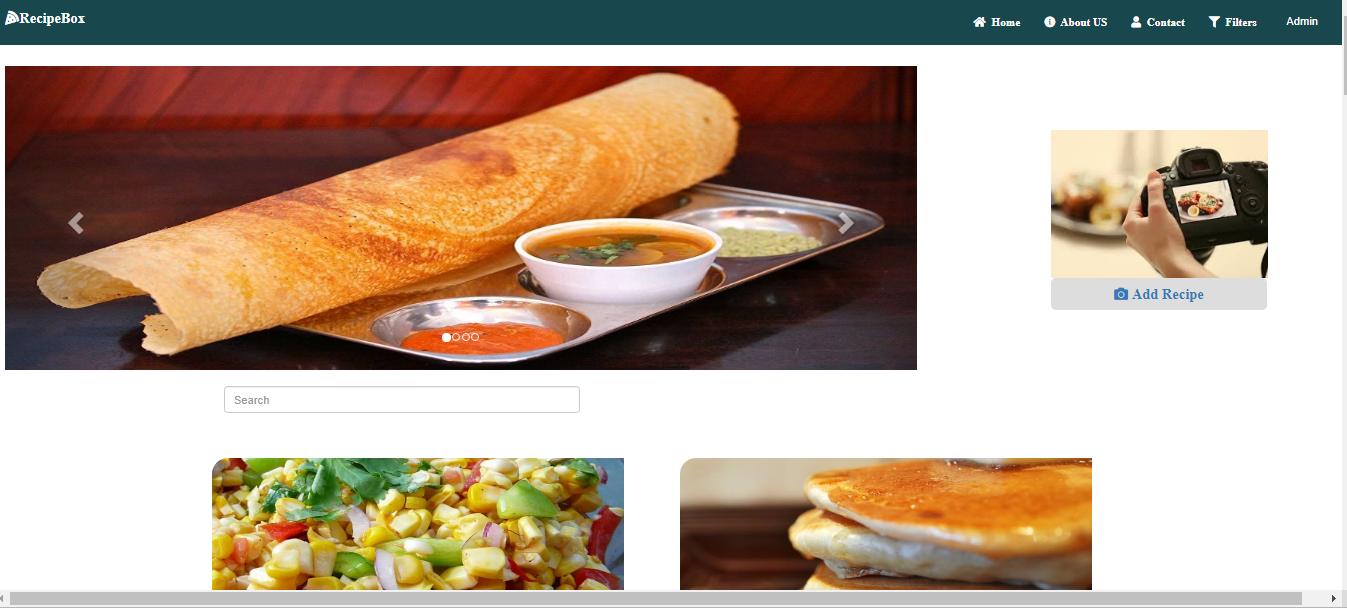
1. Homepage
2. Navbar
3. Showing details of every recipe using routing with passing ids of the particular item of database.
4. We are providing Search bar that can anything from your entire data base,

( using ngx2searcg package).

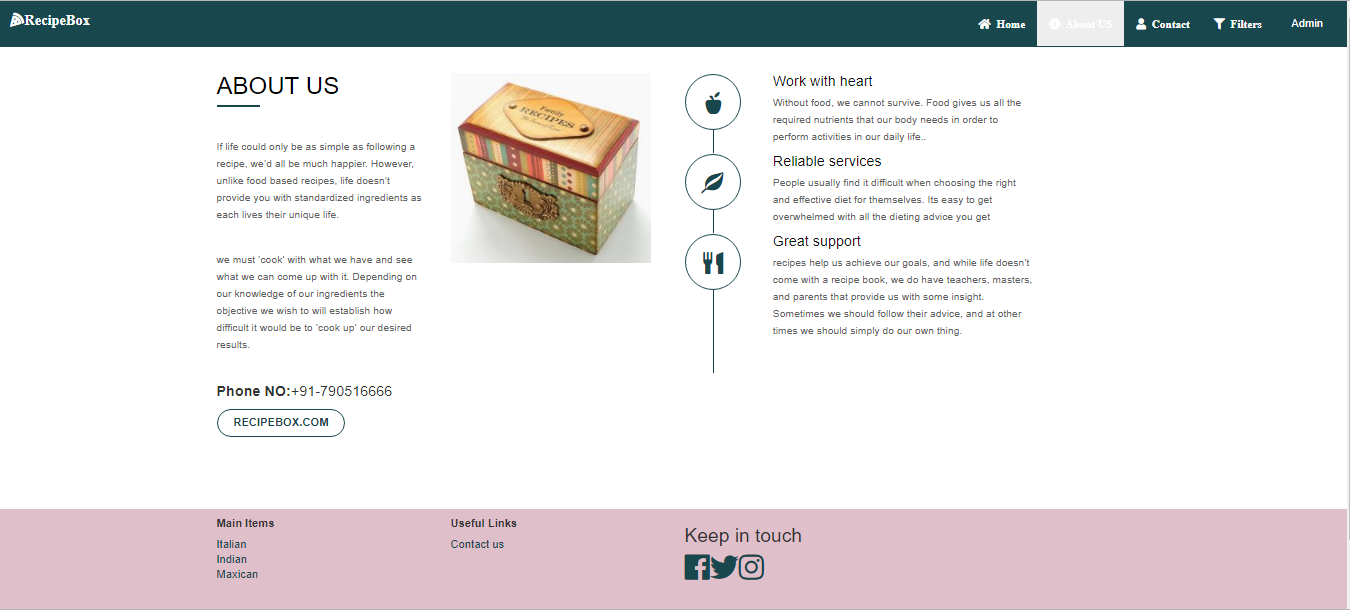
1. We are providing Filters using Pipes concepts filter by name, ingredients etc.
2. We are allowing any user to add their own recipe to this application with images.
3. We have well define footer.
4. About-us page that includes information and good features of this application.

SCREENSHOTS

1. HOMEPAGE



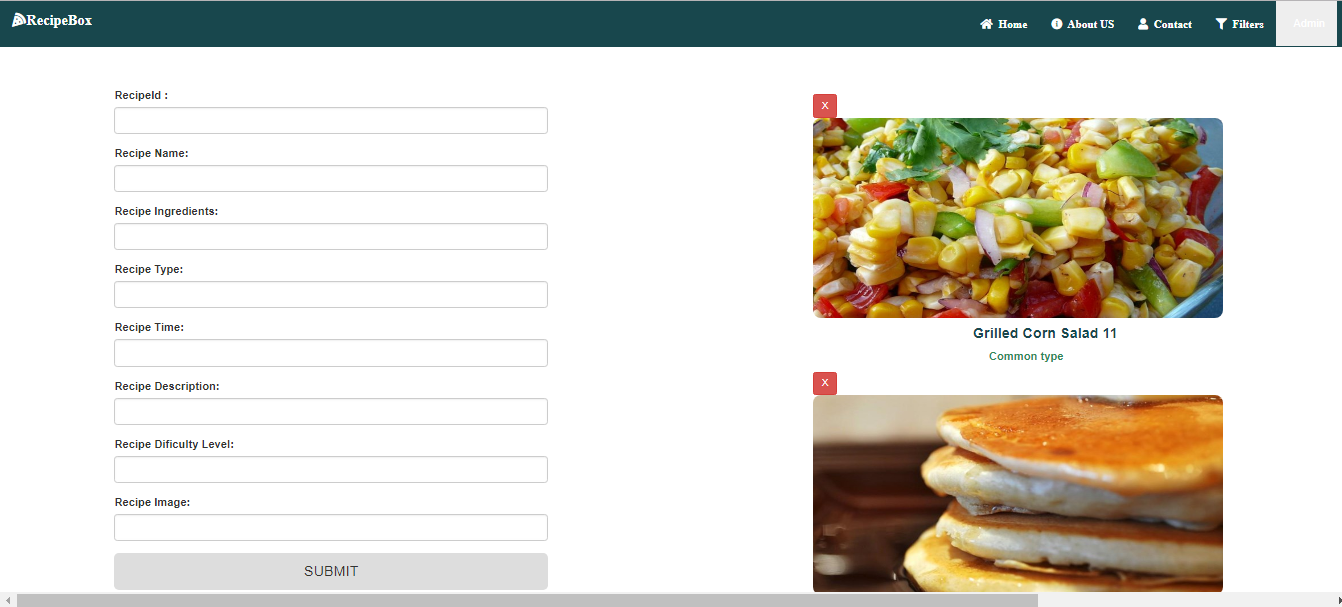
1. About-us page



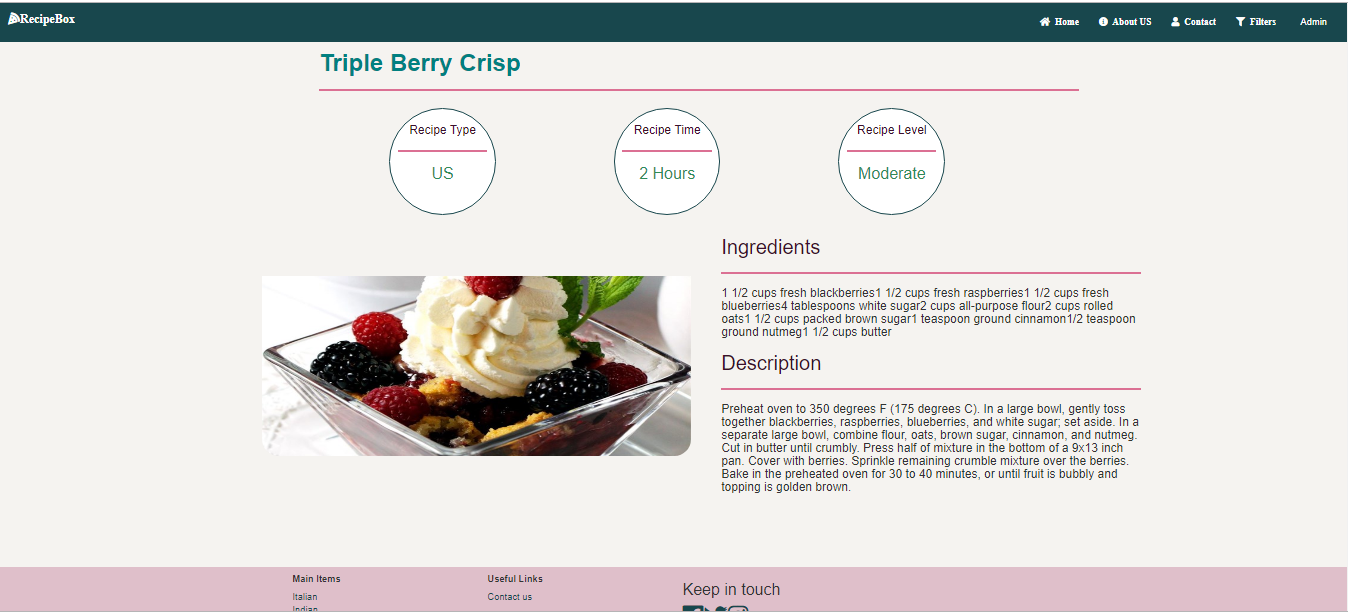
1. .contact us page



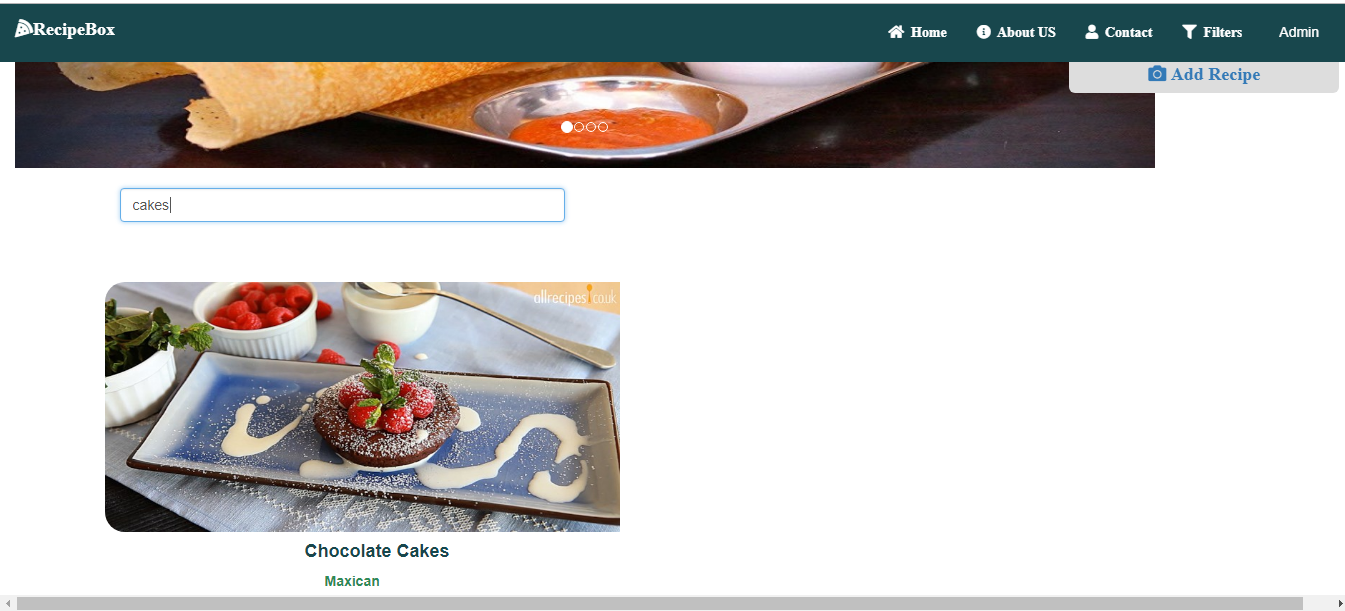
4. ADMIN PAGE



5. Details page



6. SEARCH BAR



Components:

**Homepage Component**

1 :Home page html file

<div class="row">

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

<label style="font-size:20px">Enter Recipe Name:</label>

</div>

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

<input type="text" class="form-control" placeholder="Recipe" />

</div>

</div>

<div class="row">

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

<div class="container"style="height:40%;padding-top:50px">

<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>

<li data-target="#myCarousel" data-slide-to="3"></li>

</ol>

<!-- Wrapper for slides -->

<div class="carousel-inner">

<div class="item active">

<img src="/assets/c3.jpg" />

</div>

<div class="item">

<img src="/assets/c2.jpg" />

</div>

<div class="item">

<img src="/assets/c4.jpg" />

</div>

<div class="item">

<img src="/assets/c5.jpg" />

</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>

</div>

<div class="col-md-3"style="float:right;Padding-left:50px;padding-top:130px">

<a routerLink="/admin">

<img src="/assets/camera.jpg" />

</a>

<div>

<div >

<a routerLink="/admin"><button class="btn btn-lg btn-block fa fa-camera"

style="width:270px"> Add Recipe </button>

</a>

</div>

</div>

</div>

<div class="row">

<div class="container" style="padding-top:20px;">

<div class="col-sm-5" style="padding-top:20px" >

<input class="form-control" type="text" placeholder="Search"

[(ngModel)]="filter" uib-typeahead="let emp of allEmp | async|filter:filter| limitTo:8" >

</div>

</div>

</div>

</div>

<div >

<div class="container">

<br/>

<br/>

<br/>

<div class="row r">

<div class="card" \*ngFor="let emp of allEmp|async|filter:filter " >

<a routerLink="/recipe-details/{{emp.RecipeId}}">

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

<img src="{{'.'+emp.RecipeImage}}" style="width:100%;height:250px;border-radius:20px;padding-right:40px" >

<h4 style="Padding-left:200px;color:rgb(24, 71, 77)"><b>{{emp.RecipeName}}</b></h4>

<p style="Padding-left:220px;color:seagreen"><b>{{emp.RecipeType}}</b></p>

</div>

</a>

</div>

</div>

</div>

</div>

2. Homepage ts file

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

import { RecipeBoxService } from '../shared/recipe-box.service';

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

import { Observable } from "rxjs";

@Component({

selector: 'app-home',

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

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

})

export class HomeComponent implements OnInit {

private \_allEmp: Observable<RecipeBox[]>;

public get allEmp(): Observable<RecipeBox[]> {

return this.\_allEmp;

}

public set allEmp(value: Observable<RecipeBox[]>) {

this.\_allEmp = value;

}

constructor(public employeeService:RecipeBoxService) { }

loadDisplay(){

debugger;

this.allEmp= this.employeeService.GetEmployee();

}

ngOnInit() {

this.loadDisplay();

}

}

3. Homepage css

.car{

color:rgb(24, 71, 77);

}

.ss

ca.car{

width:100%;padding-left:0px;margin-top:50px;

margin-bottom:28px;margin-left:0px

}

.r{

height:500px;

}

.img

{

width:100%;height:250px;border-radius:20px;padding-right :40px;

Padding-left:200px;color:rgb(24, 71, 77);

}

.pa

{

Padding-left:220px;color:seagreen;

}

.zoom {

transition: transform .1s;

}

.zoom:hover {

transform: scale(1.1);

}

**Nav component**

1. Nav html
2. <nav class="navbar nav navbar-fixed-top"style="margin-bottom:100px" >
3. <div class="container-fluid">
4. <div class="navbar-header">
5. <a class="navbar-brand logo fas fa-pizza-slice" routerLink="/"style="color:white">{{Apptitle}}</a>
6. </div>
7. <ul class="nav navbar-nav nav" style="float:right">
8. <li><a routerLink="/"class ="fa fa-home">&nbsp;&nbsp;Home</a></li>
9. <li><a routerLink="/about-us"style="color:white" class="fa fa-info-circle">&nbsp;&nbsp;About US</a></li>
10. <li><a routerLink="/contact"class="fa fa-user">&nbsp;&nbsp;Contact</a></li>
11. <li><a routerLink="/filters"class="fa fa-filter">&nbsp;&nbsp;Filters</a></li>
12. <li><a routerLink="/admin">&nbsp;&nbsp;Admin</a></li>
13. <!-- <li><div class="row">
14. <div class="col-md-2">
15. <label>Enter Recipe Name:</label>
16. </div>
17. <div class="col-md-4">
18. <input type="text" class="form-control" placeholder="Recipe" />
19. </div>
20. </div></li> -->
21. </ul>
22. </div>
23. </nav>

2. NAV CSS

.nav

{

background-color:rgb(24, 71, 77);

}

.navbar-nav>li>a{

color:white;

line-height:27px;

}

3. Nav TS File

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

@Component({

selector: 'app-nav',

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

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

})

export class NavComponent implements OnInit {

Apptitle:string='RecipeBox'

constructor() { }

ngOnInit() {

}

}

**FOOTER COMPONENET**

1. FOOTER HTML FILE

<footer>

<div class="footer" style="background-color:rgb(223, 191, 202); height:200px">

<div class="container">

<div class="row">

<div class="col-xs-12 col-sm-6 col-md-3 footer-two">

<h5><b>Main Items</b> </h5>

<ul class="list-unstyled"style="color:#EA1553">

<li><a style="color:rgb(24, 71, 77)">Italian</a></li>

<li><a style="color:rgb(24, 71, 77)">Indian</a></li>

<li><a style="color:rgb(24, 71, 77)">

Maxican</a></li>

</ul>

</div>

<div class="col-xs-12 col-sm-6 col-md-3 footer-three">

<h5><b>Useful Links </b></h5>

<ul class="list-unstyled">

<li><a routerLink="/contact" style="color:rgb(24, 71, 77)">Contact us</a></li>

</ul>

</div>

<div class="col-xs-12 col-sm-6 col-md-3 footer-one">

<h3>Keep in touch</h3>

<div class="social-icons ">

<a href="https://www.facebook.com/"style="color:rgb(24, 71, 77)"><i class='fab fa-facebook zoom' style='font-size:36px'></i></a>

<a href="https://twitter.com/" style="color:rgb(24, 71, 77)"><i class='fab fa-twitter zoom' style='font-size:36px'></i></a>

<a href="https://plus.google.com/"style="color:rgb(24, 71, 77)"><i class='fab fa-instagram zoom' style='font-size:36px'></i></a>

</div>

</div>

</div>

</div>

</div>

</footer>

1. FOOTER CSS FILE

.zoom {

transition: transform .1s;

}

.zoom:hover {

transform: scale(1.1); /\* (150% zoom - Note: if the zoom is too large, it will go outside of the viewport) \*/

}

1. FOOTER TS FILE
2. .zoom {
3. transition: transform .1s;
4. }
5. .zoom:hover {
6. transform: scale(1.1); /\* (150% zoom - Note: if the zoom is too large, it will go outside of the viewport) \*/
7. }

**ADMIN COMPONENT**

1. ADMIN HTML
2. <div class="row">
3. <div class="col-md-1"></div>
4. <div class="col-md-4">
5. <app-recipeform></app-recipeform>
6. </div>
7. <div class="col-md-2"></div>
8. <div class="col-md-4">
9. <app-recipelist></app-recipelist>
10. </div>
11. </div >

2. ADMIN 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() {

}

}

**CONTACT component**

Contact html

<div class="container-fluid bgimage">

<img src="" alt="Snow" style="width:100%;">

<div class="bottom-right">Bottom Right</div>

<div class="centered">

<h1> <a class="aboutus-more"routerLink="/">RecipeBox.com</a>

</h1>

<h4 style="padding-left:40px"> Phone No: 790517699</h4>

</div>

</div>

Contact css file

.bgimage

{

background-image: url('/assets/contact.jpg') ;

background-repeat: no-repeat;

margin-top:0%;

padding-top :0px;

height:100%;

background-size:100%;

height:800px;}

.container {

position: relative;

text-align: center;

color: white;

}

.centered {

position: absolute;

top: 50%;

left: 50%;

transform: translate(-50%, -50%);

}

3. contact ts file

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

@Component({

selector: 'app-contact',

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

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

})

export class ContactComponent implements OnInit {

constructor() { }

ngOnInit() {

}

}

**ADD RECIPE COMPONENT**

1. ADD RECIPE HTML
2. <div class="row">
3. <div class="col-md-4">
4. <app-recipeform></app-recipeform>
5. </div>
6. <!-- <div class="col-md-8">
7. <app-recipelist></app-recipelist>
8. </div> -->
9. </div >

2. ADD-RECIPETS FILE

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

@Component({

selector: 'app-add-recipe',

templateUrl: './add-recipe.component.html',

styleUrls: ['./add-recipe.component.css']

})

export class AddRecipeComponent implements OnInit {

constructor() { }

ngOnInit() {

}

**RECIPE FORM COMPONENT**

1. RECIPE FORM HTML FILE
2. <div >
3. <form #form="ngForm" (submit)=OnSubmit(form) autocomplete="off">
4. <div class="form-group"style="margin-top:20%">
5. <label> RecipeId :</label>
6. <input class="form-control" class="form-control" name="RecipeId" required [(ngModel)]="this.service.formData.RecipeId" #RecipeId=ngModel>
7. </div>
9. <div class="form-group">
10. <label> Recipe Name:</label>
11. <input class="form-control" name="RecipeName" [(ngModel)]="this.service.formData.RecipeName" #RecipeName="ngModel" required>
12. <div class="validation-error" \*ngIf="RecipeName.invalid && RecipeName.touched">This field is required!</div>
13. </div>
14. <div class="form-group">
15. <label> Recipe Ingredients:</label>
16. <input class="form-control" id="RecipeIngredients" name="RecipeIngredients" [(ngModel)]="this.service.formData.RecipeIngredients" #RecipeIngredients="ngModel" required>
17. <div class="validation-error" \*ngIf="RecipeIngredients.invalid && RecipeIngredients.touched">This field is required!</div>
18. </div>
19. <div class="form-group">
20. <label> Recipe Type:</label>
21. <input class="form-control" name="RecipeType" [(ngModel)]="this.service.formData.RecipeType" #RecipeType="ngModel" required>
22. <div class="validation-error" \*ngIf="RecipeType.invalid && RecipeType.touched">This field is required!</div>
23. </div>
24. <div class="form-group">
25. <label> Recipe Time:</label>
26. <input class="form-control" name="RecipeTime" [(ngModel)]="this.service.formData.RecipeTime" #RecipeTime="ngModel" required>
27. <div class="validation-error" \*ngIf="RecipeTime.invalid && RecipeTime.touched">This field is required!</div>
28. </div>
29. <div class="form-group">
30. <label> Recipe Description:</label>
31. <input class="form-control" name="RecipeDescription" [(ngModel)]="this.service.formData.RecipeDescription" #RecipeDescription="ngModel" required>
32. <div class="validation-error" \*ngIf="RecipeDescription.invalid && RecipeDescription.touched">This field is required!</div>
33. </div>
35. <div class="form-group">
36. <label> Recipe Dificulty Level:</label>
37. <input class="form-control" name="RecipeDificultyLevel" [(ngModel)]="this.service.formData.RecipeDificultyLevel" #RecipeDificultyLevel="ngModel" required >
38. <div class="validation-error" \*ngIf="RecipeDificultyLevel.invalid && RecipeDificultyLevel.touched">This field is required!</div>
39. </div>
41. <div class="form-group">
42. <label> Recipe Image:</label>
43. <input class="form-control" name="RecipeImage" [(ngModel)]="this.service.formData.RecipeImage" #RecipeImage="ngModel" required>
44. <div class="validation-error" \*ngIf="RecipeImage.invalid && RecipeImage.touched">This field is required!</div>
45. </div>
46. <div class="form-group">
47. <button type="submit" class="btn btn-lg btn-block" (click)="postEmployee"> SUBMIT </button>
48. </div>
50. </form>
51. </div>

2. RECIPE FORM CSS

.validation-error

{

color:red;

}

3. RECIPE FORM TS

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

import { RecipeBoxService } from '../../shared/recipe-box.service';

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

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

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

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

import { ToastrService } from 'ngx-toastr';

@Component({

selector: 'app-recipeform',

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

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

})

export class RecipeformComponent implements OnInit {

public empids:number[];

public result:any;

pid:number;

emp:RecipeBox;

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

{

this.service.formData={

RecipeId:null,

RecipeName:null,

RecipeIngredients:null,

RecipeType:null,

RecipeTime:null,

RecipeDescription:null,

RecipeDificultyLevel:null,

RecipeImage:null

}

this.service.refreshList();

this.result=this.service.list;

}

ngOnInit() {

this.resetForm();

}

OnSubmit(form:NgForm)

{

this.pid=form.value.RecipeId;

this.emp=this.service.list.find(e=>e.RecipeId==this.pid);

console.log(this.pid);

if(this.emp==null)

{

this.InsertRecord(form);

}

else

{

this.UpdateRecord(form);

}

}

InsertRecord(form:NgForm)

{

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

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

this.resetForm(form)

this.service.refreshList();

});

}

UpdateRecord(form:NgForm)

{

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

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

this.resetForm(form)

this.service.refreshList();

});

}

resetForm(form?:NgForm)

{

if(form!=null)

{

form.resetForm();

this.service.formData={

RecipeId:null,

RecipeName:null,

RecipeIngredients:null,

RecipeType:null,

RecipeTime:null,

RecipeDescription:null,

RecipeDificultyLevel:null,

RecipeImage:null

}

}

}

}

**RECIPE LIST COMPONENET**

1. RECIPELIST HTML
2. <div class="container">
3. <div class="row">
4. <div class="col-md-2">
5. <label style="font-size:20px">Enter Recipe Name:</label>
6. </div>
7. <div class="col-md-4">
8. <input type="text" class="form-control" placeholder="Recipe" />
9. </div>
10. </div>
11. <br/>
12. <br/>
13. <br/>
14. <div class="row">
15. <div class="card" \*ngFor="let emp of service.list " >
16. <div class="col-md-1">
17. <button (click)="onDelete(emp.RecipeId)" class="btn btn-sm btn-danger">X</button>
18. </div>
19. <div class="col-md-12">
20. <img src="{{'.'+emp.RecipeImage}}" (click)=Populateform(emp) style="width:45%;height:250px;border-radius:10px ;" >
21. <h4 (click)=Populateform(emp) style="Padding-left:200px;color:rgb(24, 71, 77)"><b>{{emp.RecipeName}}</b></h4>
23. <p (click)=Populateform(emp) style="Padding-left:220px;color:seagreen"><b>{{emp.RecipeType}}</b></p>

26. </div>
28. </div>
29. </div>
30. </div>

2. RECIPE LIST TS

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

import { RecipeBoxService } from '../../shared/recipe-box.service';

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

import { ToastrService } from 'ngx-toastr';

@Component({

selector: 'app-recipelist',

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

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

})

export class RecipelistComponent implements OnInit {

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

ngOnInit() {

{debugger}

this.service.refreshList();

}

Populateform(emp:RecipeBox)

{

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

}

onDelete(id:number)

{

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

console.log(id);

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

{

this.service.refreshList();

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

});

}

}

}

**Recipebox filter component**

1. Filters html file
2. <div class="row"style="padding-top:100px;padding-bottom:100px">
3. <div class="col-md-5" style="padding-left:100px">
4. <h4><b>Enter Name Of Recipe:</b></h4>
5. <hr />
6. <input type="text" class="form-control" [(ngModel)]="recname" placeholder="SEARCH BY RECIPE NAME" />
7. </div>
8. <div class="col-md-5" style="padding-left:200px">
9. <h4><b>Enter Type Of Recipe:</b></h4>
10. <hr/>
11. <input type="text" class="form-control"[(ngModel)]="rectype" placeholder="SEARCH BY RECIPE TYPE " />
12. </div>
13. </div>
15. <div class="row r">
16. <div class="card" \*ngFor="let emp of service.list|RecipeNamefilterPipe:recname|RecipetypefilterPipe:rectype" >
17. <a routerLink="/recipe-details/{{emp.RecipeId}}">
18. <div class="col-lg-6 zoom">
19. <img src="{{'.'+emp.RecipeImage}}" style="width:100%;height:250px;border-radius:20px;padding-right :40px;padding-left:40px" >
20. <h4 style="Padding-left:200px;color:rgb(24, 71, 77)"><b>{{emp.RecipeName}}</b></h4>
22. <p style="Padding-left:220px;color:seagreen"><b>{{emp.RecipeType}}</b></p>

25. </div>
26. </a>
27. </div>
28. </div>

2. filters css file

\* {

box-sizing: border-box;

}

body {

font-family: -apple-system,BlinkMacSystemFont,"Segoe UI","Roboto","Oxygen","Ubuntu","Cantarell","Fira Sans","Droid Sans","Helvetica Neue",sans-serif;

padding: 1rem;

background-color: #FFFFFF;

-webkit-font-smoothing: antialiased;

-moz-font-smoothing: antialiased;

-moz-osx-font-smoothing: grayscale;

-webkit-overflow-scrolling: touch;

text-rendering: optimizelegibility;

}

/\*START of Product List\*/

/\* Fallback Layout \*/

.productlist {

max-width: 500px;

margin: 0 auto;

}

.product {

margin: 1rem 0;

}

/\*Grid Layout Properties\*/

@supports (display: grid) {

.productlist{

max-width: 10000px;

margin: 0;

display: grid;

grid-template-columns: repeat(auto-fit, minmax(300px, 1fr));

grid-gap: 1rem;

}

.product {

margin: 0;

transition: all 0.3s ease-in-out;

}

}

/\*Internal Properties\*/

.productlist {

font-family: Avenir, Roboto, Helvetica, san-serif;

font-size: 85%;

}

@keyframes float {

0% {

box-shadow: 0 5px 15px 0px rgba(0,0,0,0.6);

transform: translatey(0px);

}

100% {

box-shadow: 0 25px 15px 0px rgba(0,0,0,0.2);

transform: translatey(-5px);

}

/\*100% {

box-shadow: 0 5px 15px 0px rgba(0,0,0,0.6);

transform: translatey(0px);

}\*/

}

.product {

box-sizing: border-box;

background-color:#a8193d;

display: flex;

flex-flow: column;

padding-bottom: 40px;

box-shadow: 2px 2px 5px 1px rgba(0,0,0,0.6);

}

.product:hover{

box-shadow: 0 5px 15px 0px rgba(0,0,0,0.6);

transform: translateY(-3px);

animation: float 0.5s linear;

animation-fill-mode: forwards;

border-radius: 3px;

}

h2 {

height: auto;

width: 100%;

font-size: 1.7em;

text-align: center;

}

a{

color: #FFFFFF;

text-decoration: none;

}

.inventory {

color: #FFFFFF;

text-align: center;

margin: 0 0 0rem;

}

.cost {

color:#FFFFFF;

font-size: 1.2rem;

text-align: center;

margin: 0 0 1rem;

}

.image {

order: -1;

align-self: center;

}

img{

max-width: 100%;

height: auto;

border-radius: 3px 3px 0 0;

box-sizing:border-box;

box-shadow: 0 2px 5px 1px rgba(0,0,0,0.4);

}

form{

margin: auto;

color: white;

padding-left: 2rem;

padding-bottom: 0.5rem;

border: none;

border-radius: 1rem;

font-size: 1.2rem;

}

input[type=number] {

transition: all 0.3s ease-in-out;

padding:5px 15px;

background-color: #dadada;

border:0 none;

cursor:pointer;

-webkit-border-radius: 4px;

border-radius: 4px;

border: 2px solid #dadada;

border-radius: 7px;

}

input[type=submit] {

transition: all 0.3s ease-in-out;

padding:5px 15px;

background:#dadada;

border: 2px solid #dadada;

-webkit-border-radius: 5px;

border-radius: 5px;

cursor: pointer;

}

input[type=number]:hover {

outline: none;

border-color: #9ecaed;

background-color:#9ecaed;

box-shadow: 0 0 10px #9ecaed;

}

input[type=submit]:hover {

outline: none;

color: var(--dark);

background-color:#9ecaed;

border-color: #9ecaed;

box-shadow: 0 0 10px #9ecaed;

}

/\*END of Product List\*/

Filters.componenet.ts file

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

import { RecipeBoxService } from '../shared/recipe-box.service';

import { ToastrService } from 'ngx-toastr';

@Component({

selector: 'app-filters',

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

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

})

export class FiltersComponent implements OnInit {

constructor(public service:RecipeBoxService,public toaster:ToastrService) { }

ngOnInit() {

this.service.refreshList();

}

}

RecipeFilrer.pipe.ts file

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

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

@Pipe({

name:'RecipeNamefilterPipe'

})

export class RecipeNamefilterPipe implements PipeTransform{

transform(rec:RecipeBox[],recname:string)

{

if(!rec||!recname)

{

return rec;

}

return rec.filter((x=>x.RecipeName.toLowerCase().indexOf(recname.toLowerCase())!==-1));

}

}

@Pipe({

name:'RecipetypefilterPipe'

})

export class RecipetypefilterPipe implements PipeTransform{

transform(rec1:RecipeBox[],rectype:string)

{

if(!rec1|| !rectype)

{

return rec1;

}

return rec1.filter((x=>x.RecipeType.toLowerCase().indexOf(rectype.toLowerCase())!==-1));

}

}

**RecipeDetails componenet**

1. Recipedetails html file
2. <div class=" container-fluid b d">
3. <div class="container" style="padding-top:50px">
4. <div>
6. <h1 style="color:teal">
7. <b>{{b.RecipeName}}</b>
8. <hr style=" border-top: 3px double palevioletred;"/>
9. </h1>
10. </div>
11. <div class="row" style="Padding-left:200px;margin-right: -292px;
12. margin-left: -116px;">
13. <section class="section\_0">
14. <div class="col-sm-3">
15. <div class="circle">
16. <h4 style="color:rgb(59, 16, 43)">
17. Recipe Type
18. </h4>
19. <hr style=" border-top: 3px double palevioletred;"/>
20. <h2><p>{{b.RecipeType}}</p></h2>
21. </div>
22. </div>
23. <div class="col-sm-3">
24. <div class="circle">
25. <h4 style="color:rgb(59, 16, 43)">
26. Recipe Time
27. </h4>
28. <hr style="border-top: 3px double palevioletred;"/>
29. <h2><p>{{b.RecipeTime}}</p></h2>
30. </div>
31. </div>
32. <div class="col-sm-3">
33. <div class="circle">
34. <h4 style="color:rgb(59, 16, 43)">
35. Recipe Level
36. </h4>
37. <hr style="border-top: 3px double palevioletred;"/>
38. <h2><p>{{b.RecipeDificultyLevel}}</p></h2>
39. </div>
40. </div>
41. <!-- <div class="col-sm-3">
42. <div class="circle">
43. <a href="#section\_4"><h2>4<small>th</small><p>Title</p></h2></a>
44. </div>
45. </div> -->
46. </section>
47. </div>
48. <div class="col-md-12"style="padding-bottom:100px">
49. <div class="row"style=";margin-right: -392px; margin-left: -116px;">
50. <div class="col-md-5"style="border-radius:40%">
51. <img src="{{b.RecipeImage}}" style="width:100%;height:350px;padding-top :80px; border-radius: 25px"/>
52. </div>
53. <div class="col-md-5"style="Padding-left:30px">
54. <h2 style="color:rgb(59, 16, 43)">
55. Ingredients
56. </h2>
57. <hr style="border-top: 3px double palevioletred;"/>
58. <h4>
59. <div>
60. {{b.RecipeIngredients}}
61. </div>
62. </h4>
63. <h2 style="color:rgb(59, 16, 43)">
64. Description
65. </h2>
66. <hr style="border-top: 3px double palevioletred;"/>
67. <h4>
68. {{b.RecipeDescription}}
69. </h4>
70. </div>
71. </div>
72. </div>
73. </div>

RECIPE BOX

Recipe details css file

.d{

overflow-x:hidden;overflow-y:hidden

}

.circle {

margin: 6px;

display: inline-block;

padding: 11px;

text-align: center;

width: 160px;

height: 160px;

border-radius: 50%;

border: 2px solid rgb(24, 71, 77);

}

.circle {

position: relative;

z-index: 2;

background-color: #fff;

border: 2px solid rgb(24, 71, 77);

color:seagreen;

line-height: 50px;

overflow: hidden;

}

.circle a {

text-decoration: none;

color: seagreen;

}

.circle h2 {

font-size: 60px;

}

.circle h2 small {

color: seagreen;

}

.circle p {

font-size: 24px;

line-height: 26px;

}

.b

{

background-color:rgb(245, 243, 240);

padding-left:100px;

padding-right:20px;

}

Recipedetails ts file

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

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

import { RecipeBoxService } from '../shared/recipe-box.service';

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

import { THIS\_EXPR } from '@angular/compiler/src/output/output\_ast';

@Component({

selector: 'app-recipe-details',

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

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

})

export class RecipeDetailsComponent implements OnInit {

sub:any;

Recipe:RecipeBox[];

b:RecipeBox;

constructor(public route:ActivatedRoute,public service:RecipeBoxService) { }

ngOnInit() {

this.sub=this.route

.params

.subscribe(v =>this.dosearch(v['id']));

}

dosearch(id:number):any

{

this.service.refreshList();

console.log(id);

this.b=this.service.list.find(x =>x.RecipeId==id);

console.log(this.b);

return this.b;

}

}

**SERVICE COMPONENT**

1. RECIPEBOX TS FILE
2. import { Injectable } from '@angular/core';
3. import { RecipeBox } from './RecipeBox.model';
4. import {HttpClient, HttpHeaders, HttpParams} from '@angular/common/http';
5. import { Observable } from 'rxjs/internal/Observable';
6. @Injectable({
7. providedIn: 'root'
8. })
9. export class RecipeBoxService {
10. formData:RecipeBox;
11. list:RecipeBox[];
13. readonly rootURL="http://localhost:52797/api/"
14. constructor(private http:HttpClient) { }
15. postRec(formData:RecipeBox)
16. {
17. return this.http.post(this.rootURL+'/RecipeBoxTables',formData);
18. }
19. GetEmployee():Observable<RecipeBox[]>
20. {
21. return this.http.get<RecipeBox[]>(this.rootURL + '/RecipeBoxTables');
22. }
23. refreshList():void
24. {
26. this.http.get(this.rootURL+'/RecipeBoxTables')
27. .toPromise().then(res=>this.list=res as RecipeBox[]);
28. }
29. putRec(formData:RecipeBox)
30. {
31. const headers = new HttpHeaders().set('content-type', 'application/json');
32. const params = new HttpParams().set('ID', formData.RecipeId.toString());
33. {
34. return this.http.put<RecipeBox>(this.rootURL+'/RecipeBoxTables/'+formData.RecipeId,formData,{headers,params});
35. }
36. }
38. deleteRec(id:number)
39. {
40. return this.http.delete(this.rootURL+'/RecipeBoxTables/'+id);
41. }
43. }

**RECIPEBOX MODEL COMPONENT**

1. RECIPEBOX-MODEL TS FILE
2. export class RecipeBox{
3. RecipeId:number;
4. RecipeName:string;
5. RecipeIngredients:string;
6. RecipeType:string;
7. RecipeTime:string;
8. RecipeDescription:string;
9. RecipeDificultyLevel:string;
10. RecipeImage:string;
11. }

**App routing module**

1. App routing module ts file
2. import { NgModule } from '@angular/core';
3. import { Routes, RouterModule } from '@angular/router';
4. import { HomeComponent } from './home/home.component';
5. import { AboutUsComponent } from './about-us/about-us.component';
6. import { ContactComponent } from './contact/contact.component';
7. import { AdminComponent } from './admin/admin.component';
8. import {AddRecipeComponent} from './add-recipe/add-recipe.component'
9. import { RecipeDetailsComponent } from './recipe-details/recipe-details.component';
10. import { FiltersComponent } from './filters/filters.component';
11. import { ToastrModule } from 'ngx-toastr';
12. import { Ng2SearchPipeModule } from 'ng2-search-filter';
13. const routes: Routes = [
14. { path:'',component: HomeComponent},
15. { path:'about-us',component:AboutUsComponent},
16. { path:'contact',component:ContactComponent},
17. {path:'add-recipe',component:AddRecipeComponent},
18. {path:'admin',component:AdminComponent},
19. {path:'recipe-details/:id',component:RecipeDetailsComponent},
20. {path:'filters',component:FiltersComponent}
21. ];
22. @NgModule({
23. imports: [ToastrModule,
24. RouterModule.forRoot(routes)],
25. exports: [RouterModule]
26. })
27. export class AppRoutingModule { }

App.component.html

<app-nav></app-nav>

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

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

<app-recipeform></app-recipeform>

</div>

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

<app-recipelist></app-recipelist> -->

<!-- </div>

</div > -->

<section>

<router-outlet> </router-outlet>

</section>

<app-footer></app-footer>

<!-- <app-filters></app-filters> -->

App.componenet.ts file

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

@Component({

selector: 'app-root',

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

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

})

export class AppComponent {

title = 'MyRecipeBoxProject';

}

App.module.ts file

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

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

import { ToastrModule } from 'ngx-toastr';

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

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

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

import { HomeComponent } from './home/home.component';

import { AboutUsComponent } from './about-us/about-us.component';

import { ContactComponent } from './contact/contact.component';

import { NavComponent } from './nav/nav.component';

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

import { RecipeformComponent } from './home/recipeform/recipeform.component';

import { RecipelistComponent } from './home/recipelist/recipelist.component';

import { RecipeBoxService } from './shared/recipe-box.service';

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

import { RecipeDetailsComponent } from './recipe-details/recipe-details.component';

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

import { FooterComponent } from './footer/footer.component';

import { AddRecipeComponent } from './add-recipe/add-recipe.component';

import { FiltersComponent } from './filters/filters.component';

import {RecipeNamefilterPipe,RecipetypefilterPipe} from './filters/RecipeFilter.pipe';

import { Ng2SearchPipeModule } from 'ng2-search-filter';

@NgModule({

declarations: [

AppComponent,

HomeComponent,

AboutUsComponent,

ContactComponent,

NavComponent,

RecipeformComponent,

RecipelistComponent,

RecipeDetailsComponent,

AdminComponent,

FooterComponent,

AddRecipeComponent,

FiltersComponent,

RecipeNamefilterPipe,

RecipetypefilterPipe

],

imports: [

BrowserModule,

AppRoutingModule,

FormsModule,

HttpClientModule,

BrowserAnimationsModule,Ng2SearchPipeModule,

ToastrModule.forRoot()

],

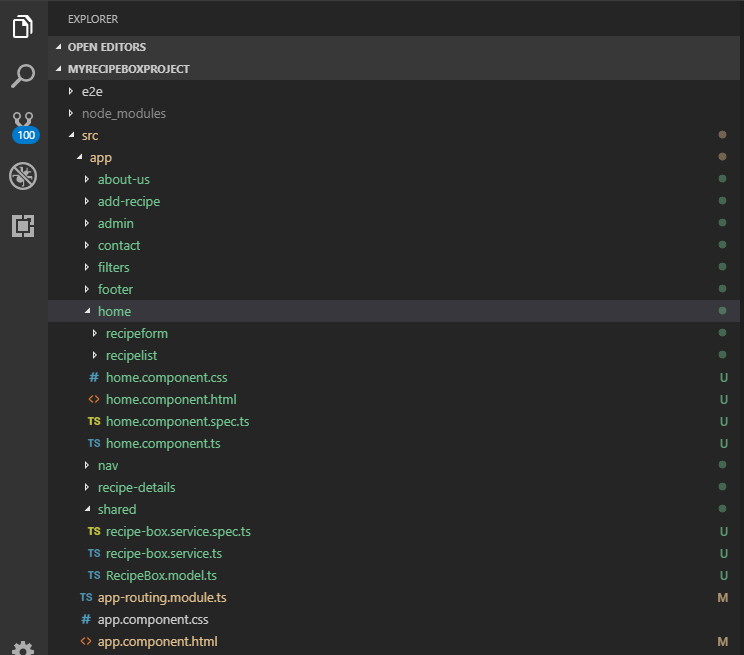
providers: [RecipeBoxService],

bootstrap: [AppComponent]

})

export class AppModule { }

**files structure**





**WEB API**

RecipeBoxTablesController

using System;

using System.Collections.Generic;

using System.Data;

using System.Data.Entity;

using System.Data.Entity.Infrastructure;

using System.Linq;

using System.Net;

using System.Net.Http;

using System.Web.Http;

using System.Web.Http.Description;

using RecipeBoxWebApi;

namespace RecipeBoxWebApi.Controllers

{

public class RecipeBoxTablesController : ApiController

{

private RecipeBoxEntities db = new RecipeBoxEntities();

// GET: api/RecipeBoxTables

public IQueryable<RecipeBoxTable> GetRecipeBoxTables()

{

return db.RecipeBoxTables;

}

// GET: api/RecipeBoxTables/5

[ResponseType(typeof(RecipeBoxTable))]

public IHttpActionResult GetRecipeBoxTable(int id)

{

RecipeBoxTable recipeBoxTable = db.RecipeBoxTables.Find(id);

if (recipeBoxTable == null)

{

return NotFound();

}

return Ok(recipeBoxTable);

}

// PUT: api/RecipeBoxTables/5

[ResponseType(typeof(void))]

public IHttpActionResult PutRecipeBoxTable(int id, RecipeBoxTable recipeBoxTable)

{

if (!ModelState.IsValid)

{

return BadRequest(ModelState);

}

if (id != recipeBoxTable.RecipeId)

{

return BadRequest();

}

db.Entry(recipeBoxTable).State = EntityState.Modified;

try

{

db.SaveChanges();

}

catch (DbUpdateConcurrencyException)

{

if (!RecipeBoxTableExists(id))

{

return NotFound();

}

else

{

throw;

}

}

return StatusCode(HttpStatusCode.NoContent);

}

// POST: api/RecipeBoxTables

[ResponseType(typeof(RecipeBoxTable))]

public IHttpActionResult PostRecipeBoxTable(RecipeBoxTable recipeBoxTable)

{

if (!ModelState.IsValid)

{

return BadRequest(ModelState);

}

db.RecipeBoxTables.Add(recipeBoxTable);

db.SaveChanges();

return CreatedAtRoute("DefaultApi", new { id = recipeBoxTable.RecipeId }, recipeBoxTable);

}

// DELETE: api/RecipeBoxTables/5

[ResponseType(typeof(RecipeBoxTable))]

public IHttpActionResult DeleteRecipeBoxTable(int id)

{

RecipeBoxTable recipeBoxTable = db.RecipeBoxTables.Find(id);

if (recipeBoxTable == null)

{

return NotFound();

}

db.RecipeBoxTables.Remove(recipeBoxTable);

db.SaveChanges();

return Ok(recipeBoxTable);

}

protected override void Dispose(bool disposing)

{

if (disposing)

{

db.Dispose();

}

base.Dispose(disposing);

}

private bool RecipeBoxTableExists(int id)

{

return db.RecipeBoxTables.Count(e => e.RecipeId == id) > 0;

}

}

}

WebApiConfig.tcs file

using System;

using System.Collections.Generic;

using System.Linq;

using System.Net.Http;

using System.Web.Http;

using Microsoft.Owin.Security.OAuth;

using Newtonsoft.Json.Serialization;

using System.Web.Http.Cors;

namespace RecipeBoxWebApi

{

public static class WebApiConfig

{

public static void Register(HttpConfiguration config)

{

config.EnableCors(new EnableCorsAttribute("http://localhost:4200", headers: "\*", methods: "\*"));

// Web API configuration and services

// Configure Web API to use only bearer token authentication.

config.SuppressDefaultHostAuthentication();

config.Filters.Add(new HostAuthenticationFilter(OAuthDefaults.AuthenticationType));

// Web API routes

config.MapHttpAttributeRoutes();

config.Routes.MapHttpRoute(

name: "DefaultApi",

routeTemplate: "api/{controller}/{id}",

defaults: new { id = RouteParameter.Optional }

);

}

}

}