**foodblogproject.component.ts**

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

import { BlogpostsservicesService } from '../Services/blogpostsservices.service';

import { postdata } from './postdata.model';

import { FoodblogService } from '../Services/foodblog.service';

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

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

@Component({

selector: 'app-foodblogproject',

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

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

})

export class FoodblogprojectComponent implements OnInit {

dbposts:postdata[]=[]

isFetching=false

errorMessage=null

createform=false

home=true

checkblog=false

constructor(private foodservice:FoodblogService) { }

ngOnInit(): void {

this.fetchposts()

}

createformdisplay(){

this.createform=true

this.home=false

}

create(postdata:{title:string,content:string,image:string},form:NgForm){

this.foodservice.createposts(postdata.title,postdata.content,postdata.image)

.subscribe((response)=>{

console.log(response)

form.reset()

this.checkblog=true

})

}

fetchposts(){

this.isFetching=true

this.foodservice.fetchposts()

.subscribe((posts)=>{

this.isFetching=false

console.log(posts);

this.dbposts=posts

// console.log(this.dbposts);

},

(error)=>{

this.errorMessage=error.error.error;

})

}

ondeleteposts(){

this.foodservice.deleteposts()

.subscribe(()=>{

console.log('posts deleted')

this.fetchposts()

})

}

onfetchposts(){

this.fetchposts()

this.home=true

this.createform=false

}

}

**foodblogproject.component.html**

<div class="container" style="width:100%">

<div class="row">

<div class="col">

<div class="center" m-4 style="width:400px">

<span \*ngIf="home"><br><br>

<br><br><br><br><br><button class="btn btn-warning btn-lg mx-5"(click)=createformdisplay()>

Create new food blog

</button>

</span>

</div>

<div class="container" m-4 style="width:400px">

<form #f="ngForm" \*ngIf="createform">

<div class="form-group">

<label for="title">Title</label>

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

id="title"

name="title"

ngModel required/>

</div>

<div class="form-group">

<label for="content">Content</label>

<textarea

name="content" id="" cols="30" rows="10"

class="form-control"

ngModel required

></textarea>

</div>

<div class="form-group">

<label for="image">Image</label>

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

id="image"

name="image"

ngModel/>

</div>

<button class="btn btn-primary"

type="submit"

[disabled]="!f.valid"

(click)="create(f.value,f)">

Create blog

</button>

<button class="btn btn-warning m-3" mx-2

[disabled]="!checkblog"

(click)="onfetchposts()">

Check blog

</button>

<!-- <button class="btn btn-danger" mx-2 (click)="ondeleteposts()">

Delete Posts

</button> -->

</form>

</div>

<div class="row m-5" m-3 \*ngIf="home">

<div class="col">

<p \*ngIf="dbposts.length<1 && !isFetching">No posts Available</p>

<ul class="list-group" \*ngIf="dbposts.length>=1 && !isFetching" >

<app-singleblog

\*ngFor="let post of dbposts"

[post]="post">

</app-singleblog>

</ul>

<p \*ngIf="isFetching && !errorMessage">Loading....</p>

<div class="alert alert-danger" \*ngIf="errorMessage">

<h5>An error occured</h5>

<p>{{errorMessage}}</p>

</div>

</div>

</div>

</div>

</div>

</div>

**foodblogproject.component.css**

.center {

margin:0 auto;

display:block;

left: 50%;

float: none;

top: 50%;

transform: translate(12%, -50%);

}

**Postdata.model.ts**

export interface postdata{

title:string

content:string

id?:string

image?:string

}

**singleblog.component.ts**

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

import { postdata } from '../postdata.model';

import { FoodblogService } from 'src/app/Services/foodblog.service';

import { FoodblogprojectComponent } from '../foodblogproject.component';

@Component({

selector: 'app-singleblog',

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

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

})

export class SingleblogComponent implements OnInit {

update=false

@Input()

post:postdata

constructor(private foodservice:FoodblogService,private foodblogpost:FoodblogprojectComponent) { }

ngOnInit(): void {

}

ondelete(){

this.foodservice.deletesinglepost(this.post.id)

.subscribe(()=>{

this.foodblogpost.onfetchposts()

})

}

updateopen(){

this.update=true

}

}

**singleblog.component.html**

<div class="container" style="width:100%">

<div class="row">

<ul>

<div class="col"><app-update \*ngIf="update" [post]="post"></app-update></div>

</ul>

</div>

</div>

<div class="card" \*ngIf="!update">

<div class="bg-image hover-overlay" mdbRipple rippleColor="light">

<img src="{{post.image}}" class="img-fluid" style="width: 100%;"/>

<a href="#!">

<div class="mask" style="background-color: rgba(251, 251, 251, 0.15);"></div>

</a>

</div>

<div class="card-body">

<h5 class="card-title">{{post.title}}</h5>

<p class="card-text">{{post.content}}</p>

<div class="end">

<button class="btn btn-primary m-3" (click)=updateopen()>Update</button>

<button class="btn btn-warning" (click)=ondelete()>Delete</button>

</div>

</div>

</div><br><br><br>

**singleblog.component.css**

.end{

text-align: right;}

**update.component.ts**

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

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

import { FoodblogService } from 'src/app/Services/foodblog.service';

import { FoodblogprojectComponent } from '../../foodblogproject.component';

import { SingleblogComponent } from '../singleblog.component';

@Component({

selector: 'app-update',

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

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

})

export class UpdateComponent implements OnInit {

@Input()

post

constructor(private service:FoodblogService,private singleblog:SingleblogComponent,private foodblog:FoodblogprojectComponent) { }

ngOnInit(): void {

}

updatepost(postdata:{title:string,content:string,image:string},form:NgForm){

console.log(postdata.image)

this.service.updatepost(this.post.id,postdata.title,postdata.content,postdata.image)

this.singleblog.update=false

this.foodblog.fetchposts()

}

}

**update.component.html**

<p>update </p>

<div class="container" m-4 style="width:400px">

<div class="row">

<div class="col">

<form #f="ngForm">

<div class="form-group">

<label for="title">Title</label>

<input type="text"

placeholder="{{post.title}}"

[(ngModel)]="post.title"

id="title"name="title"

ngModel required>

</div>

<div class="form-group">

<label for="content">Content</label>

<textarea

placeholder="{{post.content}}"

name="content"

[(ngModel)]="post.content"

ngModel required></textarea>

</div>

<div class="form-group">

<label for="title">Image</label>

<input type="text"

placeholder="place image address"

id="image"name="image"

[(ngModel)]="post.image"

ngModel>

</div>

<button class="btn btn-primary"

type="submit"

[disabled]="!f.valid"

(click)="updatepost(f.value,f)">

Update Post

</button>

</form>

</div>

</div>

</div>

**foodblog.service.ts**

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

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

import { postdata } from '../foodblogproject/postdata.model';

import{ map } from 'rxjs/operators'

@Injectable({

providedIn: 'root'

})

export class FoodblogService {

url=' http://localhost:3000/blog'

list=[]

constructor(private httpclient:HttpClient) { }

createposts(title:string,content:string,image:string){

const mypostdata:postdata={title:title,content:content,image:image}

return this.httpclient.post(

this.url, mypostdata

)

}

fetchposts(){

return this.httpclient.get(

this.url

)

.pipe(

map((responsedata)=>{

const postsarray=[]

for(const key in responsedata){

postsarray.push({...responsedata[key]})

}

return postsarray

}

))

}

deleteposts(){

return this.httpclient

.delete(

this.url

)

}

deletesinglepost(postid){

const id=parseInt(postid)

return this.httpclient

.delete(

this.url+'/'+(id)

)

}

updatepost(id :number,chtitle:string,chcontent:string,chimage:string){

console.log(chimage)

const body={

title:chtitle,

content:chcontent,

image:chimage

}

this.httpclient.patch(

this.url+"/"+id,body)

.subscribe(()=>{

this.fetchposts()

console.log("Updated")

})

}

}

**foodblogproject.json**

{

"blog": [

{

"title": "Pizza",

"content": "Pizza is a dish of Italian origin consisting of a usually round, flat base of leavened wheat-based dough topped with tomatoes, cheese, and often various other ingredients, which is then baked at a high temperature, traditionally in a wood-fired oven. A small pizza is sometimes called a pizzetta",

"id": 11,

"image": "https://cdn.pixabay.com/photo/2017/12/09/08/18/pizza-3007395\_\_480.jpg"

},

{

"title": "Pasta",

"content": "Pasta is a type of food made from a mixture of flour, eggs, and water that is formed into different shapes and then boiled.",

"image": "https://images.unsplash.com/photo-1555949258-eb67b1ef0ceb?ixlib=rb-4.0.3&ixid=MnwxMjA3fDB8MHxzZWFyY2h8MTF8fHBhc3RhfGVufDB8fDB8fA%3D%3D&w=1000&q=80",

"id": 14

},

{

"title": "Salad",

"content": "A salad is a dish consisting of mixed, mostly natural ingredients. They are typically served chilled or at room temperature, though some can be served warm.",

"image": "https://images7.alphacoders.com/300/thumb-1920-300220.jpg",

"id": 12

},

{

"title": "Burger",

"content": "A burger is a flat round mass of minced meat or vegetables, which is fried and often eaten in a bread roll.",

"image": "https://images.unsplash.com/photo-1571091718767-18b5b1457add?ixlib=rb-4.0.3&ixid=MnwxMjA3fDB8MHxzZWFyY2h8OXx8YnVyZ2VyfGVufDB8fDB8fA%3D%3D&w=1000&q=80",

"id": 13

},

{

"title": "Sandwich Cookies",

"content": "A sandwich cookie, also known as a sandwich biscuit, is a type of cookie made from two thin cookies or medium cookies with a filling between them. Many types of fillings are used, such as cream, ganache, buttercream, chocolate, cream cheese, jam, peanut butter, lemon curd, or ice cream. ",

"image": "https://hips.hearstapps.com/delish/assets/16/25/1466452319-delish-ice-cream-sandwich-cookies.jpg",

"id": 15

},

{

"title": "Flourless Chocolate Almond Cake",

"content": "Unlike regular cakes made of flour, flourless chocolate cake is not fluffy nor light. It tastes like a rich chocolate truffle. It's silky, rich, and smooth.",

"image": "https://food.fnr.sndimg.com/content/dam/images/food/fullset/2015/4/9/0/KC0510H\_Flourless-Chocolate-Almond-Cake\_s4x3.jpg.rend.hgtvcom.616.462.suffix/1449693307400.jpeg",

"id": 16

},

{

"title": "Strawberry Lemonade Bars",

"content": "Strawberry Lemonade Bars combine fresh, fruity strawberries, zesty lemonade, and a buttery shortbread crust to create a sweet, simple spring or summer dessert!",

"image": "https://images.ctfassets.net/7t85a6f8wt2m/106iPPKxlp4Yn9V4fMXit5/3f8c879a57186f7d13df19bddb5f777d/Strawberry\_Lemonade\_Bars\_image?fm=webp&q=75",

"id": 17

},

{

"title": "French Onion Soup",

"content": "French onion soup is traditionally made with yellow onions. Also, French onion soup is always made with beef stock while onion soup uses any type of stock or broth.",

"image": "https://www.cookingclassy.com/wp-content/uploads/2022/02/french-onion-soup-28.jpg",

"id": 18

},

{

"title": "Strawberry Water",

"content": "This delicious flavored water has the essence of that red berry and it is refreshing!",

"image": "https://www.acouplecooks.com/wp-content/uploads/2021/06/Strawberry-Water-005-735x919.jpg",

"id": 19

},

{

"title": "Air Fryer Roasted Potatoes",

"content": "Turn to your air fryer for this impossibly crispy and easy side dish.\n\n",

"image": "https://cdn.apartmenttherapy.info/image/upload/f\_auto,q\_auto:eco,c\_fill,g\_center,w\_730,h\_913/k%2FPhoto%2FRecipe%20Ramp%20Up%2F2022-02-Air-Fryer-Roasted-Potatoes%2FIMG\_6850",

"id": 20

},

{

"title": "Caprese Salad Bites Appetizer Recipe",

"content": "Easy caprese salad bites on a toothpick drizzled with a rosemary balsamic dressing.",

"image": "https://www.homemadefoodjunkie.com/wp-content/uploads/2016/05/Caprese-Salad-Bites-party-tray-720x405.jpg",

"id": 21

},

{

"title": "French Fries",

"content": "These crazy delicious, crispy french fries are as easy as they are impressive!",

"image": "https://images.themodernproper.com/billowy-turkey/production/posts/2022/Homemade-French-Fries\_8.jpg?w=960&h=960&q=82&fm=jpg&fit=crop&dm=1662474181&s=f6b09b96f732330eca2aae43140b3ffa",

"id": 22

},

{

"title": "Momo",

"content": "Momo is a type of Tibetan-Nepalese steamed filled dumpling. It is popular in Bhutan, Nepal and Tibet. In Nepal, where it is considered a national dish, they are usually served with a sauce known as achar influenced by the spices and herbs used within many South Asian cuisines.",

"image": "https://www.foodieist.com/wp-content/uploads/2020/12/veg-momos-recipe-672x372.jpg",

"id": 23

}

]

}