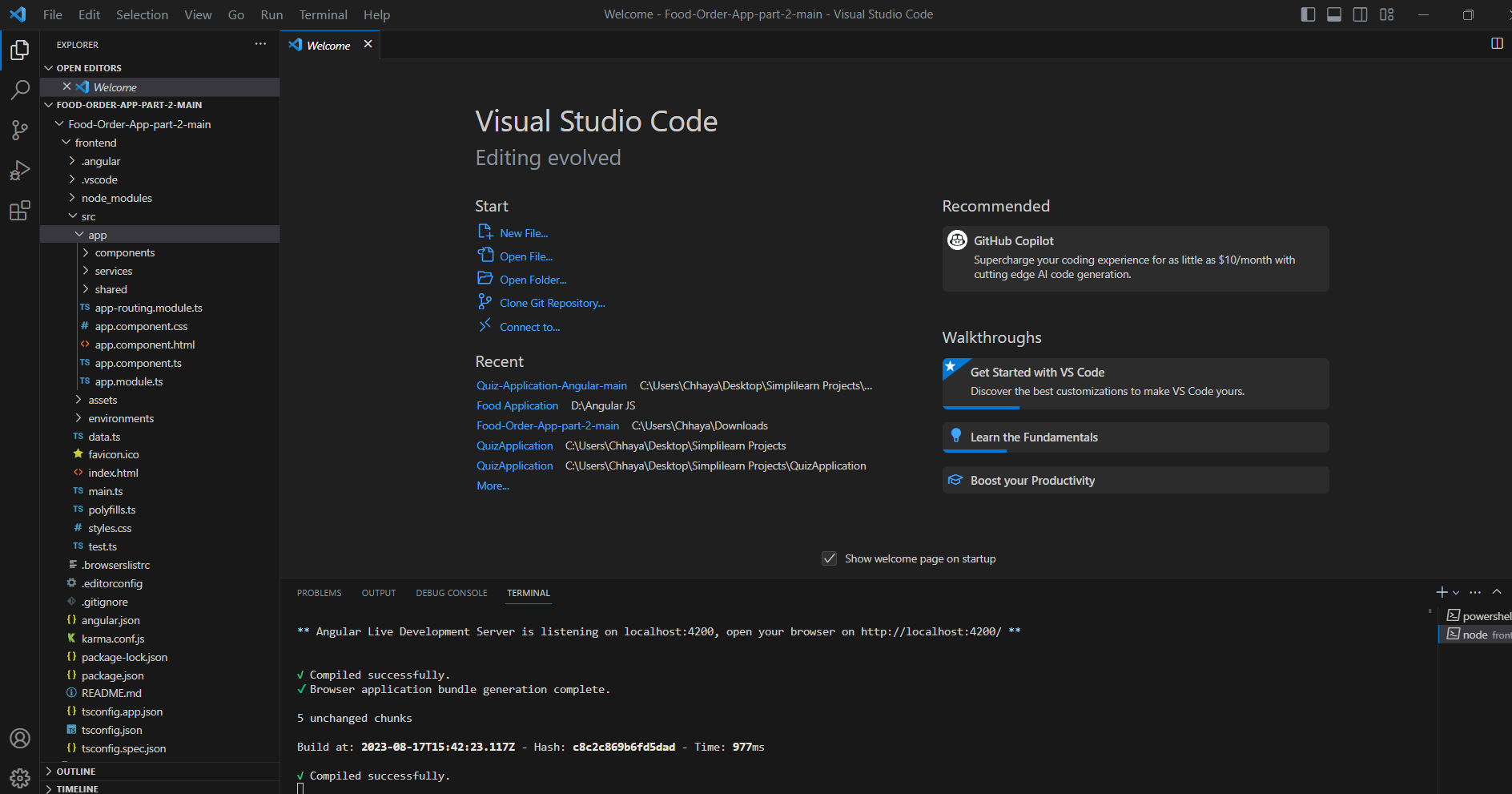
* Home Page



* App.routing.module.ts:
* import { NgModule } from '@angular/core';
* import { RouterModule, Routes } from '@angular/router';
* import { CartPageComponent } from './components/pages/cart-page/cart-page.component';
* import { FoodPageComponent } from './components/pages/food-page/food-page.component';
* import { HomeComponent } from './components/pages/home/home.component';
* const routes: Routes = [
* { path: '', component: HomeComponent },
* { path: 'search/:searchTerm', component: HomeComponent },
* { path: 'tag/:tag', component: HomeComponent },
* {path:'food/:id', component:FoodPageComponent},
* {path:'cart-page', component: CartPageComponent}
* ];
* @NgModule({
* imports: [RouterModule.forRoot(routes)],
* exports: [RouterModule]
* })
* export class AppRoutingModule { }
* app.component.html:

<app-header></app-header>

<router-outlet></router-outlet>

* app.component.ts:
* import { Component } from '@angular/core';
* @Component({
* selector: 'app-root',
* templateUrl: './app.component.html',
* styleUrls: ['./app.component.css']
* })
* export class AppComponent {
* title = 'frontend';
* }
* Data.ts:
* import {Food} from './app/shared/models/Food';
* import { Tag } from './app/shared/models/Tag';
* export const sample\_foods: Food[] = [
* {
* id:'1',
* name: 'Pizza Pepperoni',
* cookTime: '10-20',
* price: 10,
* favorite: false,
* origins: ['italy'],
* stars: 4.5,
* imageUrl: 'assets/food-1.jpg',
* tags: ['FastFood', 'Pizza', 'Lunch'],
* },
* {
* id:'2',
* name: 'Meatball',
* price: 20,
* cookTime: '20-30',
* favorite: true,
* origins: ['persia', 'middle east', 'china'],
* stars: 4.7,
* imageUrl: 'assets/food-2.jpg',
* tags: ['SlowFood', 'Lunch'],
* },
* {
* id:'3',
* name: 'Hamburger',
* price: 5,
* cookTime: '10-15',
* favorite: false,
* origins: ['germany', 'us'],
* stars: 3.5,
* imageUrl: 'assets/food-3.jpg',
* tags: ['FastFood', 'Hamburger'],
* },
* {
* id:'4',
* name: 'Fried Potatoes',
* price: 2,
* cookTime: '15-20',
* favorite: true,
* origins: ['belgium', 'france'],
* stars: 3.3,
* imageUrl: 'assets/food-4.jpg',
* tags: ['FastFood', 'Fry'],
* },
* {
* id:'5',
* name: 'Chicken Soup',
* price: 11,
* cookTime: '40-50',
* favorite: false,
* origins: ['india', 'asia'],
* stars: 3.0,
* imageUrl: 'assets/food-5.jpg',
* tags: ['SlowFood', 'Soup'],
* },
* {
* id:'6',
* name: 'Vegetables Pizza',
* price: 9,
* cookTime: '40-50',
* favorite: false,
* origins: ['italy'],
* stars: 4.0,
* imageUrl: 'assets/food-6.jpg',
* tags: ['FastFood', 'Pizza', 'Lunch'],
* },
* ]
* export const sample\_tags:Tag[] = [
* { name: 'All', count: 6 },
* { name: 'FastFood', count: 4 },
* { name: 'Pizza', count: 2 },
* { name: 'Lunch', count: 3 },
* { name: 'SlowFood', count: 2 },
* { name: 'Hamburger', count: 1 },
* { name: 'Fry', count: 1 },
* { name: 'Soup', count: 1 },
* ]
* Index.html:
* <!doctype html>
* <html lang="en">
* <head>
* <meta charset="utf-8">
* <title>Frontend</title>
* <base href="/">
* <meta name="viewport" content="width=device-width, initial-scale=1">
* <link rel="icon" type="image/x-icon" href="favicon.ico">
* </head>
* <body>
* <app-root></app-root>
* </body>
* </html>
* Main.ts:
* import { enableProdMode } from '@angular/core';
* import { platformBrowserDynamic } from '@angular/platform-browser-dynamic';
* import { AppModule } from './app/app.module';
* import { environment } from './environments/environment';
* if (environment.production) {
* enableProdMode();
* }
* platformBrowserDynamic().bootstrapModule(AppModule)
* .catch(err => console.error(err));
* Style.css:
* /\* You can add global styles to this file, and also import other style files \*/
* @import url("https://fonts.googleapis.com/css2?family=Quicksand&display=swap");
* \*{
* box-sizing: border-box;
* }
* html{
* font-size:16px;
* }
* html,
* button{
* font-family: "Quicksand", "sans-serif";
* }
* body{
* margin: 0;
* }
* a{
* text-decoration: none;
* }
* **Component/pages/cart-page:**
* **Cart-page.component.css:**
* .container{
* display: flex;
* flex-wrap: wrap;
* align-items: flex-start;
* margin: 1.5rem;
* margin-top: 0.5rem;
* }
* ul{
* display: flex;
* flex-direction: column;
* flex:3 0;
* justify-content: space-evenly;
* border: 1px solid #ffbbbb;
* border-radius: 1rem;
* list-style: none;
* margin: 0.5rem;
* padding: 0;
* }
* ul li{
* display: flex;
* justify-content: space-around;
* align-items: center;
* flex-wrap: wrap;
* border-bottom: 1px solid #e4e4e4;
* }
* ul li:last-child{
* border: none;
* }
* li img{
* width: 5rem;
* height: 5rem;
* border-radius: 100rem;
* object-fit: cover;
* }
* li div{
* padding: 1rem;
* }
* li div:not(:first-child){
* flex-basis: 18%;
* }
* li select{
* width: 3rem;
* outline: none;
* border: none;
* border-bottom: 1px solid lightgrey;
* font-size: 1.1rem;
* font-weight: 100;
* }
* li .remove-button{
* border-radius: 1rem;
* border: none;
* padding: 0.5rem;
* color: #e72929;
* opacity: 0.7;
* outline: none;
* }
* li .remove-button:hover{
* opacity: 1;
* cursor: pointer;
* }
* .checkout{
* display: flex;
* flex-direction: column;
* justify-content: space-between;
* align-items: center;
* flex:1 3;
* height: 20rem;
* border: 1px solid #ffbbbb;
* border-radius: 1rem;
* padding: 0.5rem;
* margin: 0.5rem;
* }
* .checkout>div{
* font-size: 1.4rem;
* margin: 1rem;
* flex: 3;
* display: flex;
* flex-direction: column;
* justify-content: center;
* align-items: flex-start;
* }
* .foods-count{
* margin-bottom: 1.5rem;
* }
* .foods-count::before{
* content:"Count: ";
* color: grey;
* }
* .total-price::before{
* content: 'Price: ';
* color: grey;
* }
* .checkout a{
* padding: 1rem;
* color: white;
* background:#e72929;
* display: block;
* width: 99%;
* border-radius: 1rem;
* text-align: center;
* opacity: 0.7;
* justify-self: center;
* }
* .checkout a:hover{
* opacity: 1;
* cursor: pointer;
* }
* **Cart-page.component.html:**
* <app-title title="Cart Page" margin="1.5rem 0 0 2.5rem"></app-title>
* <div class="container">
* <ul>
* <li \*ngFor="let cartItem of cart.items">
* <div>
* <img [src]="cartItem.food.imageUrl" [alt]="cartItem.food.name">
* </div>
* <div>
* <a routerLink="/food/{{cartItem.food.id}}">
* {{cartItem.food.name}}
* </a>
* </div>
* <div>
* <select #quantitySelect
* [value]="cartItem.quantity"
* (change)="changeQuantity(cartItem, quantitySelect.value)">
* <option>1</option>
* <option>2</option>
* <option>3</option>
* <option>4</option>
* <option>5</option>
* </select>
* </div>
* <div>
* {{cartItem.price | currency}}
* </div>
* <div>
* <button class="remove-button" (click)="removeFromCart(cartItem)">
* Remove
* </button>
* </div>
* </li>
* </ul>
* <div class="checkout">
* <div>
* <div class="foods-count">{{cart.totalCount}}</div>
* <div class="total-price">{{cart.totalPrice|currency}}</div>
* </div>
* <a routerLink="/checkout">Proceed to Checkout</a>
* </div>
* </div>
* **Cart-page.component.ts:**
* import { Component, OnInit } from '@angular/core';
* import { CartService } from 'src/app/services/cart.service';
* import { Cart } from 'src/app/shared/models/Cart';
* import { CartItem } from 'src/app/shared/models/CartItem';
* @Component({
* selector: 'app-cart-page',
* templateUrl: './cart-page.component.html',
* styleUrls: ['./cart-page.component.css']
* })
* export class CartPageComponent implements OnInit {
* cart!: Cart;
* constructor(private cartService: CartService) {
* this.cartService.getCartObservable().subscribe((cart) => {
* this.cart = cart;
* })
* }
* ngOnInit(): void {
* }
* removeFromCart(cartItem:CartItem){
* this.cartService.removeFromCart(cartItem.food.id);
* }
* changeQuantity(cartItem:CartItem,quantityInString:string){
* const quantity = parseInt(quantityInString);
* this.cartService.changeQuantity(cartItem.food.id, quantity);
* }
* }
* **Component/pages/food-page:**
* **Food-page.component.css:**
* .container{
* display: flex;
* justify-content: center;
* align-items: center;
* flex-wrap: wrap;
* margin: 3rem;
* }
* .container > \*{
* min-width: 25rem;
* max-width: 40rem;
* }
* img{
* border-radius: 3rem;
* flex: 1 0;
* object-fit: cover;
* height: 35rem;
* margin: 1rem;
* }
* .details{
* width: 100%;
* display: flex;
* flex-direction: column;
* flex:1 0;
* padding: 2rem;
* color: black;
* margin-left: 1rem;
* }
* .header{
* display: flex;
* justify-content: space-between;
* }
* .name{
* font-size: 2rem;
* font-weight: bold;
* }
* .favorite{
* color: #e72929;
* font-size: 2.5rem;
* }
* .favorite.not{
* color: grey;
* }
* .origins{
* display: flex;
* flex-wrap: wrap;
* margin: 1rem 0;
* }
* .origins span{
* padding: 0.5rem;
* font-size: 1.2rem;
* margin: 0.5rem 0.5rem 0.5rem 0;
* border-radius: 2rem;
* background-color: aliceblue;
* }
* .tags{
* display: flex;
* flex-wrap: wrap;
* }
* .tags a{
* background-color: #f0f0f0;
* padding: 0.3rem 1rem;
* margin: 0.2rem 0.15rem;
* border-radius: 10rem;
* font-weight: 600;
* color: blue;
* }
* .cook-time{
* margin-top: 1rem;
* }
* .cook-time span{
* padding: 0.6rem 2rem 0.6rem 0;
* border-radius: 10rem;
* font-size: 1.3rem;
* }
* .price{
* font-size: 1.8rem;
* margin: 2rem 2rem 2rem 0;
* color: green;
* }
* .price::before{
* content: "Price: ";
* color: darkgrey;
* }
* button{
* color: white;
* background: #e72929;
* border: none;
* font-size: 1.2rem;
* padding: 1rem;
* border-radius: 10rem;
* outline: none;
* }
* button:hover{
* opacity: 0.9;
* cursor: pointer;
* }
* **Food-page.component.html:**
* <div class="container">
* <img [src]="food.imageUrl" [alt]="food.name"/>
* <div class="details">
* <div class="header">
* <span class="name">
* {{food.name}}
* </span>
* <span class="favorite {{food.favorite? '': 'not'}}">❤</span>
* </div>
* <div class="rating">
* <star-rating
* [value]="food.stars"
* [totalstars]="5"
* checkedcolor="red"
* uncheckedcolor="black"
* size="25px"
* [readonly]="true">
* </star-rating>
* </div>
* <div class="origins">
* <span \*ngFor="let origin of food.origins">
* {{origin}}
* </span>
* </div>
* <div class="tags">
* <a \*ngFor="let tag of food.tags" routerLink="/tag/{{tag}}">
* {{tag}}
* </a>
* </div>
* <div class="cook-time">
* <span>
* Time to cook about <strong>{{food.cookTime}}</strong> minutes
* </span>
* </div>
* <div class="price">
* <span>
* {{food.price | currency}}
* </span>
* </div>
* <button (click)="addToCart()">Add To Cart</button>
* </div>
* </div>
* **Food-page.component.ts:**

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

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

import { CartService } from 'src/app/services/cart.service';

import { FoodService } from 'src/app/services/food.service';

import { Food } from 'src/app/shared/models/Food';

@Component({

  selector: 'app-food-page',

  templateUrl: './food-page.component.html',

  styleUrls: ['./food-page.component.css']

})

export class FoodPageComponent implements OnInit {

  food!: Food;

  constructor(activatedRoute:ActivatedRoute, foodService:FoodService,

    private cartService:CartService, private router: Router) {

    activatedRoute.params.subscribe((params) => {

      if(params.id)

      this.food = foodService.getFoodById(params.id);

    })

   }

  ngOnInit(): void {

  }

  addToCart(){

    this.cartService.addToCart(this.food);

    this.router.navigateByUrl('/cart-page');

  }

}

* **Component/pages/home:**
* [**Home.component.css**](http://Home.component.css)**:**
* ul{
* display: flex;
* justify-content: center;
* align-items: center;
* flex-wrap: wrap;
* list-style-type: none;
* padding: 0;
* }
* li a{
* height: 22.5rem;
* width: 20rem;
* border: 1px solid whitesmoke;
* border-radius: 1rem;
* margin: 0.5rem;
* display: flex;
* flex-direction: column;
* overflow: hidden;
* color: #e72929;
* }
* li a:hover{
* opacity: 0.9;
* cursor: pointer;
* }
* li a img{
* object-fit: cover;
* height: 14.5rem;
* }
* .content{
* margin-top: 0.3rem;
* padding: 0.5rem 1rem;
* position: relative;
* height: 7rem;
* }
* .favorite{
* position: absolute;
* top: 0.5rem;
* right: 1rem;
* font-size: 1.2rem;
* }
* .favorite.not{
* color: grey;
* }
* .stars{
* margin-bottom: 0.2rem;
* }
* .product-item-footer{
* display: flex;
* justify-content: space-between;
* align-items: flex-start;
* }
* .origins{
* flex: 9;
* }
* .origins span{
* border-radius: 2rem;
* background-color: whitesmoke;
* display: inline-block;
* font-size: 0.8rem;
* color: grey;
* margin-right: 0.3rem;
* padding: 0 0.5rem;
* margin-top: 0.2rem;
* }
* .cook-time{
* font-size: 0.9rem;
* flex: 3;
* text-align: right;
* }
* .price{
* position: absolute;
* bottom: 0;
* left: 0.9rem;
* background: white;
* padding: 0.3rem 100% 0 0;
* color: #414141;
* margin-top: 0.7rem;
* font-size: 1.1rem;
* }
* [**Home.component.html:**](http://Home.component.html:)
* <app-search></app-search>
* <app-tags></app-tags>
* <ul>
* <li \*ngFor="let food of foods">
* <a routerLink="/food/{{food.id}}">
* <img [src]="food.imageUrl" [alt]="food.name" />
* <div class="content">
* <div class="name">
* {{food.name}}
* </div>
* <span class="favorite {{food.favorite?'':'not'}}">
* ❤
* </span>
* <div class="stars">
* <star-rating [value]="food.stars" [totalstars]="5" checkedcolor="red" uncheckedcolor="black" size="22px"
* [readonly]="true">
* </star-rating>
* </div>
* <div class="product-item-footer">
* <div class="origins">
* <span \*ngFor="let origin of food.origins">
* {{origin}}
* </span>
* </div>
* <div class="cook-time">
* <span>🕒</span>
* {{food.cookTime}}
* </div>
* </div>
* <div class="price">
* <span>
* {{food.price | currency}}
* </span>
* </div>
* </div>
* </a>
* </li>
* </ul>
* **Home.component.ts:**
* import { Component, OnInit } from '@angular/core';
* import { ActivatedRoute } from '@angular/router';
* import { FoodService } from 'src/app/services/food.service';
* import { Food } from 'src/app/shared/models/Food';
* @Component({
* selector: 'app-home',
* templateUrl: './home.component.html',
* styleUrls: ['./home.component.css']
* })
* export class HomeComponent implements OnInit {
* foods: Food[] = [];
* constructor(private foodService: FoodService, activatedRoute: ActivatedRoute) {
* activatedRoute.params.subscribe((params) => {
* if (params.searchTerm)
* this.foods = this.foodService.getAllFoodsBySearchTerm(params.searchTerm);
* else if (params.tag)
* this.foods = this.foodService.getAllFoodsByTag(params.tag);
* else
* this.foods = foodService.getAll();
* })
* }
* ngOnInit(): void {
* }
* }
* **Component/partial/header:**
* **Header.component.css**
* header{
* position: relative;
* background: white;
* padding: 0;
* border-bottom: 1px solid #e72929;
* }
* a{
* color: #af1313;
* }
* a:hover{
* background:#e72929;
* color: white;
* cursor: pointer;
* }
* .container{
* margin: 0 auto;
* display: flex;
* justify-content: space-between;
* }
* a.logo{
* font-weight: bold;
* padding: 1rem;
* }
* ul{
* display: flex;
* list-style-type: none;
* margin: 0;
* }
* ul a{
* padding: 1rem;
* display: inline-block;
* }
* a span{
* background: #ff4d4d;
* color: white;
* padding: 0.1rem 0.45rem;
* border-radius: 100rem;
* font-size:0.9rem;
* }
* .menu-container{
* position: relative;
* }
* .menu{
* position:absolute;
* z-index:1;
* background:whitesmoke;
* display: none;
* }
* .menu-container:hover .menu{
* display: block;
* }
* .menu a{
* width: 100%;
* min-width:8rem;
* }
* **Header.component.html**
* <header>
* <div class="container">
* <a routerLink="/" class="logo">Food Mine!</a>
* <nav>
* <ul>
* <li>
* <a routerLink="/login">Login</a>
* </li>
* <li class="menu-container">
* <a routerLink="/dashboard">John</a>
* <div class="menu">
* <a routerLink="/profile">Profile</a>
* <a routerLink="/orders">Orders</a>
* <a>Logout</a>
* </div>
* </li>
* <li>
* <a routerLink="/cart-page">
* Cart <span \*ngIf="cartQuantity">{{cartQuantity}}</span>
* </a>
* </li>
* </ul>
* </nav>
* </div>
* </header>
* **Header.component.ts**

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

import { CartService } from 'src/app/services/cart.service';

@Component({

  selector: 'app-header',

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

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

})

export class HeaderComponent implements OnInit {

  cartQuantity=0;

  constructor(cartService:CartService) {

    cartService.getCartObservable().subscribe((newCart) => {

      this.cartQuantity = newCart.totalCount;

    })

   }

  ngOnInit(): void {

  }

}

* **Component/partial/Search:**
* **Search.component.css**
* div{
* display: flex;
* justify-content: center;
* margin-top: 3rem;
* margin-bottom: 1.5rem;
* }
* input{
* border-radius: 10rem 0 0 10rem;
* border: none;
* height: 3rem;
* width: 20rem;
* background-color: #f1f1f1;
* padding: 1.2rem;
* font-size: 1.1rem;
* font-weight: 500;
* outline: none;
* }
* div button{
* color: grey;
* height: 3rem;
* width: 5rem;
* font-size: 1rem;
* border-radius: 0 10rem 10rem 0;
* border: none;
* background-color: #e72929;
* color: white;
* opacity: 0.8;
* outline: none;
* }
* div button:hover{
* opacity: 1;
* cursor: pointer;
* }
* **Search.component.html**
* <div>
* <input #s type="text" placeholder="Search Food Mine!"
* (keyup.enter)="search(s.value)"
* [value]="searchTerm" />
* <button (click)="search(s.value)">Search</button>
* </div>
* **Search.component.tc**
* import { Component, OnInit } from '@angular/core';
* import { ActivatedRoute, Router } from '@angular/router';
* @Component({
* selector: 'app-search',
* templateUrl: './search.component.html',
* styleUrls: ['./search.component.css']
* })
* export class SearchComponent implements OnInit {
* searchTerm = '';
* constructor(activatedRoute:ActivatedRoute,private router:Router) {
* activatedRoute.params.subscribe((params) => {
* if(params.searchTerm) this.searchTerm = params.searchTerm;
* });
* }
* ngOnInit(): void {
* }
* search(term:string):void{
* if(term)
* this.router.navigateByUrl('/search/'+ term);
* }
* }
* **Component/partial/title:**
* **title.component.css**
* div{
* display: flex;
* flex-wrap: wrap;
* justify-content: center;
* }
* div a{
* background-color: #f0f0f0;
* padding: 0.3rem 1rem;
* margin: 0.2rem 0.15rem;
* border-radius: 10rem;
* font-weight: 600;
* color: blue;
* }
* **title.component.html**
* <div \*ngIf="tags">
* <a \*ngFor="let tag of tags" routerLink="/tag/{{tag.name}}">
* {{tag.name}}({{tag.count}})
* </a>
* </div>
* **title.component.ts**
* import { Component, OnInit } from '@angular/core';
* import { FoodService } from 'src/app/services/food.service';
* import { Tag } from 'src/app/shared/models/Tag';
* @Component({
* selector: 'app-tags',
* templateUrl: './tags.component.html',
* styleUrls: ['./tags.component.css']
* })
* export class TagsComponent implements OnInit {
* tags?:Tag[];
* constructor(foodService:FoodService) {
* this.tags = foodService.getAllTags();
* }
* ngOnInit(): void {
* }
* }
* **Component/partial/tag:**
* **tag.component.css**
* h1 {
* color: #616161;
* }
* **tagcomponent.html**
* <h1 [ngStyle]="{margin:margin, fontSize: fontSize}">
* {{title}}
* </h1>
* **tag.component.ts**
* import { Component, Input, OnInit } from '@angular/core';
* @Component({
* selector: 'app-title',
* templateUrl: './title.component.html',
* styleUrls: ['./title.component.css']
* })
* export class TitleComponent implements OnInit {
* constructor() { }
* @Input()
* title!: string;
* @Input()
* margin? = '1rem 0 1rem 0.2rem';
* @Input()
* fontSize? = '1.7rem';
* ngOnInit(): void {
* }
* }
* **Sevices:**
* **Food.service.ts**
* import { Injectable } from '@angular/core';
* import { BehaviorSubject, Observable } from 'rxjs';
* import { Cart } from '../shared/models/Cart';
* import { CartItem } from '../shared/models/CartItem';
* import { Food } from '../shared/models/Food';
* @Injectable({
* providedIn: 'root'
* })
* export class CartService {
* private cart: Cart = this.getCartFromLocalStorage();
* private cartSubject: BehaviorSubject<Cart> = new BehaviorSubject(this.cart);
* constructor() { }
* addToCart(food: Food): void {
* let cartItem = this.cart.items
* .find(item => item.food.id === food.id);
* if (cartItem)
* return;
* this.cart.items.push(new CartItem(food));
* this.setCartToLocalStorage();
* }
* removeFromCart(foodId: string): void {
* this.cart.items = this.cart.items
* .filter(item => item.food.id != foodId);
* this.setCartToLocalStorage();
* }
* changeQuantity(foodId: string, quantity: number) {
* let cartItem = this.cart.items
* .find(item => item.food.id === foodId);
* if (!cartItem) return;
* cartItem.quantity = quantity;
* cartItem.price = quantity \* cartItem.food.price;
* this.setCartToLocalStorage();
* }
* clearCart() {
* this.cart = new Cart();
* this.setCartToLocalStorage();
* }
* getCartObservable(): Observable<Cart> {
* return this.cartSubject.asObservable();
* }
* private setCartToLocalStorage(): void {
* this.cart.totalPrice = this.cart.items
* .reduce((prevSum, currentItem) => prevSum + currentItem.price, 0);
* this.cart.totalCount = this.cart.items
* .reduce((prevSum, currentItem) => prevSum + currentItem.quantity, 0);
* const cartJson = JSON.stringify(this.cart);
* localStorage.setItem('Cart', cartJson);
* this.cartSubject.next(this.cart);
* }
* private getCartFromLocalStorage(): Cart {
* const cartJson = localStorage.getItem('Cart');
* return cartJson ? JSON.parse(cartJson) : new Cart();
* }
* }
* **Cart.service.ts**
* import { Injectable } from '@angular/core';
* import { sample\_foods, sample\_tags } from 'src/data';
* import { Food } from '../shared/models/Food';
* import { Tag } from '../shared/models/Tag';
* @Injectable({
* providedIn: 'root'
* })
* export class FoodService {
* constructor() { }
* getAll(): Food[] {
* return sample\_foods;
* }
* getAllFoodsBySearchTerm(searchTerm: string) {
* return this.getAll().filter(food => food.name.toLowerCase().includes(searchTerm.toLowerCase()))
* }
* getAllTags(): Tag[] {
* return sample\_tags;
* }
* getAllFoodsByTag(tag: string): Food[] {
* return tag === "All" ?
* this.getAll() :
* this.getAll().filter(food => food.tags?.includes(tag));
* }
* getFoodById(foodId:string):Food{
* return this.getAll().find(food => food.id == foodId) ?? new Food();
* }
* }
* **Shared:**
* **Cart.ts**
* import { CartItem } from "./CartItem";
* export class Cart{
* items:CartItem[] = [];
* totalPrice:number = 0;
* totalCount:number = 0;
* }
* **Cartitem.ts**
* import { Food } from "./Food";
* export class CartItem{
* constructor(public food:Food){ }
* quantity:number = 1 ;
* price: number = this.food.price;
* }
* **Food.ts**
* export class Food{
* id!:string;
* name!:string;
* price!:number;
* tags?: string[];
* favorite!:boolean;
* stars!: number;
* imageUrl!: string;
* origins!: string[];
* cookTime!:string;
* }
* **Tag.ts**
* export class Tag{
* name!:string;
* count!:number;
* }