|  |
| --- |
| **SimpliLearn Phase 4 Project: KitchenStory**  **e-commerce website**  **Submitted by: KAVIN K R** |
| Date of submission : 17-01-2023  GitHub Project Repository URL : [GitHub Link](https://github.com/kavink-r/SimpliLearn-Phase4-Project-KitchenStory.git) |
| **Sprints Planned:**  Sprint 1: Creation of navigation bar feature.  Sprint 2: Creation of food item page and cart item page feature.  Sprint 3: Creation of checkout page and payment gateway feature.  Sprint 4: Creation admin login page.  Sprint 5: Creation of admin panel features for master list of foods, adding foods, changing password and final publish.  **Build Tool:**  Angular CLI is used to create, build and run the project.  **Technologies used:**  HTML, CSS, Typescript, Angular. |
| **Project structure:** |
| **App.component.html**  <div class="whole-container">  <div class="top-container">  <div class="brand-name">  <div class="logo-container">  <img src="/assets/chef1.png" width="80" height="80">  </div>  <div class="brand\_name\_container">  <p class="brand\_name\_text">KitchenStory</p>  </div>  <div class="clear"></div>  </div>  <div class="login-container">  <button type="button" class="btn btn-outline-warning login\_button" routerLink="login">Login</button>    <div class="clear"></div>  </div>  </div>  <div class="clear"></div>  <div class="middle-container">  <router-outlet></router-outlet>  </div>  </div> |
| **App.component.css**  @import url('https://fonts.googleapis.com/css2?family=Libre+Baskerville&display=swap');  .whole-container {  width: 80%;  margin: 0 auto;  }  .top-container {  height: 85px;  background-color: antiquewhite;  border-bottom: 2px solid chocolate;  }  .middle-container{  margin-top: 10px;  border-radius: 0.5rem;  background-color: rgb(255, 249, 241);  }  .logo-container {  height: 80px;  width: 15%;  float: left;  display: inline;  }  .brand\_name\_container {  width: 85%;  text-align: center;  font-family: 'Libre Baskerville', serif;  font-size: x-large;  height: 80px;  float: left;  display: table;  }  .brand\_name\_text {  display: table-cell;  vertical-align: middle;  text-align: left;  }  .brand-name {  width: 50%;  float: left;  display: inline;  }  .login-container {  width: 50%;  float: right;  display: inline;  }  .clear {  clear: both;  }  .login\_button {  margin-top: 3%;  float: right;  width: 15%;  }  .login-text {  width: 85%;  display: table;  float: right;  height: 80px;  font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;  }  .login-text p {  display: table-cell;  vertical-align: middle;  text-align: right;  padding-right: 5%;  } |
| **App.component.ts**  import { Component } from '@angular/core';  import { AdminService } from './services/admin.service';  @Component({  selector: 'app-root',  templateUrl: './app.component.html',  styleUrls: ['./app.component.css']  })  export class AppComponent {  title = 'KitchenStory';    constructor(private adminservice:AdminService){  }  ngOnInit(){    }  } |
| **App-routing.module.ts**  import { Component, NgModule } from '@angular/core';  import { RouterModule, Routes } from '@angular/router';  import { AppComponent } from './app.component';  import { AdminAddFoodComponent } from './components/admin-add-food/admin-add-food.component';  import { AdminAllFoodsComponent } from './components/admin-all-foods/admin-all-foods.component';  import { AdminHomeComponent } from './components/admin-home/admin-home.component';  import { ChangePasswordComponent } from './components/change-password/change-password.component';  import { CheckoutComponent } from './components/checkout/checkout.component';  import { FoodItemListComponent } from './components/food-item-list/food-item-list.component';  import { HomeComponent } from './components/home/home.component';  import { LoginComponent } from './components/login/login.component';  const routes: Routes = [  {  path: '', component: HomeComponent  },  {  path: 'search/:searchTerm', component: FoodItemListComponent  },  {path: 'checkout', component: CheckoutComponent},  {path: 'login', component:LoginComponent},  {path: 'admin-home', component:AdminHomeComponent,  children: [  {path: '', component: AdminAllFoodsComponent},  { path: 'all-foods', component: AdminAllFoodsComponent },  { path: 'add-food', component: AdminAddFoodComponent },  { path: 'change-password', component: ChangePasswordComponent },  ],}  ];  @NgModule({  imports: [RouterModule.forRoot(routes)],  exports: [RouterModule]  })  export class AppRoutingModule { } |
| **home.component.html**  <div class="food-item-list">  <app-food-item-list></app-food-item-list>  </div>  <div class="cart-list">  <div><a>🛒Cart info:</a></div>  <app-cart-list></app-cart-list>  </div>  <div class="clear"></div> |
| **home.component.css**  .food-item-list {  height: 1000px;  width: 80%;  display: inline;  float: left;  /\*background-color: bisque;\*/  }  .cart-list {  height: max-content;  width: 20%;  display: inline;  float: right;  /\*border-left: 2px solid #ca1515;  background-color: coral;\*/  }  .cart-list div {  margin-top: 1rem;  text-align: center;  } |
| **home.component.ts**  import { Component, OnInit } from '@angular/core';  @Component({  selector: 'app-home',  templateUrl: './home.component.html',  styleUrls: ['./home.component.css']  })  export class HomeComponent implements OnInit {  constructor() { }  ngOnInit(): void {  }  **}** |
| **Food-item-list.component.html**  <div class="search-btn">  <input type="text" placeholder="Search Food!" id="search\_text" [(ngModel)]="searchTerm" (change)="search()">  <button (click)="search()">Search</button>  </div>  <app-not-found [visible]="!foods || foods.length <= 0" resetLinkText="Reset Search"></app-not-found>  <button (click)="reset()" \*ngIf="!foods || foods.length <= 0" class="btn btn-danger rst-button">Reset</button>  <div class="tag-menu" \*ngIf="tags" [style.justifyContent]="justifyContent">  <a \*ngFor="let tag of tags" routerLink="/tag/{{tag.name}}" (click)="getbytag(tag.name)">  {{tag.name}}({{tag.count}})  </a>  </div>  <ul>  <li \*ngFor="let food of foods">  <a href="#">  <img src="{{ food.imageUrl }}" alt="food image" />  <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="20px"  [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:"INR" }}  </span>  </div>  <button class="btn btn-danger add\_to\_cart" (click)="addToCart(food.id)">add to cart</button>  </div>  </a>  </li>  </ul> |
| **Food-item-list.component.css**  ul{  display: flex;  justify-content: center;  align-items: center;  flex-wrap: wrap;  list-style-type: none;  padding: 0%;  }  ul li a{  text-decoration: none;  height: 17rem;  width: 10rem;  border: 1px solid rgb(232, 232, 232);  border-radius: 0.5rem;  margin: 0.5rem;  display: flex;  flex-direction: column;  overflow: hidden;  color: red;  }  li a:hover{  opacity: 0.9;  cursor: pointer;  }  .content{  padding: .25rem 0.5rem;  position:relative;  }  .favorite{  position:absolute;  top: .25rem;  right: .5rem;  font-size: 0.75rem;  }  .favorite not{  color: grey;  }  .stars{  margin-bottom: .1rem;  }  .product-item-footer{  display:flex;  justify-content: space-between;  align-items: center;  }  .origins{  flex: 9;  }  .origins span{  border-radius: 0.1rem;  background-color: whitesmoke;  display: inline-block;  font-size: .7rem;  color: grey;  margin-right: .3rem;  }  .cook-time{  font-size: .8rem;  flex: 3;  text-align: right;  }  .price{  display: inline;  }  .add\_to\_cart{  width:70px;  height:30px;  float:right;  margin-top: 30px;  text-align: center;  font-size: 10px;  }  .search-btn{  display: flex;  justify-content: center;  margin-top: 3rem;  margin-bottom: 1.5rem;  }  .search-btn input{  border-radius: 10rem 0 0 10rem;  border: none;  height: 3rem;  width: 20rem;  background-color: #f1f1f1;  padding: 1.2rem;  font-size: 1rem;  font-weight: 500;  outline: none;  }  .search-btn button{  color: grey;  height: 3rem;  width: 5rem;  font-size: 1rem;  border-radius: 0 10rem 10rem 0;  border: none;  background-color: #e72929;  color: white;  opacity: .8;  outline: none;  }  .search-btn button:hover{  opacity: 1;  cursor: pointer;  }  .rst-button{  margin-left: 300px;  }  .tag-menu{  display: flex;  flex-wrap: wrap;  }  .tag-menu a{  background-color: #f0f0f0;  padding: .3rem 1rem;  margin: .2rem .15rem;  border-radius: 10rem;  font-weight: 600;  color: rgb(220, 40, 0);  } |
| **Food-item-list.component.ts**  import { Component, Input, OnInit } from '@angular/core';  import { ActivatedRoute, Router } from '@angular/router';  import { Food } from 'src/app/models/food';  import { Tag } from 'src/app/models/tags';  import { CartserviceService } from 'src/app/services/cartservice.service';  import { FoodserviceService } from 'src/app/services/foodservice.service';  @Component({  selector: 'app-food-item-list',  templateUrl: './food-item-list.component.html',  styleUrls: ['./food-item-list.component.css']  })  export class FoodItemListComponent implements OnInit {  foods:Food[]=[];  searchTerm:string = '';  @Input()  foodPageTags?:string[];  @Input()  justifyContent:string = 'center';    tags?:Tag[];  constructor(  private foodservice:FoodserviceService,  private route: ActivatedRoute,  private cartservice:CartserviceService,  private router:Router  ) { }  ngOnInit(): void {  this.route.params.subscribe(params => {  console.log(params['searchTerm']);  if (params['searchTerm']){  this.foods = this.foodservice.getAllFoodsBySearchTerm(params['searchTerm']);  console.log('search by searchterm:' + params['searchTerm']);}  else if(params['tag']){  this.foods = this.foodservice.getAllFoodsByTag(params['tag']);  console.log('search by tag:' + params['tag']);}  else{  this.foods = this.foodservice.getAll();  console.log('no searchterm:' + params['searchTerm']);}    })  if(!this.foodPageTags)  this.tags = this.foodservice.getAllTags();  }  addToCart(id:any){  let fud:Food;  fud = this.foodservice.getFoodById(id);  this.cartservice.addToCart(fud);  console.log(fud);  }  search():void{  if(this.searchTerm)  this.foods = this.foodservice.getAllFoodsBySearchTerm(this.searchTerm);  }  reset():void{  this.foods = this.foodservice.getAll();    }  getbytag(tag:any){  this.foods = this.foodservice.getAllFoodsByTag(tag);  }    } |
| **Cart-list.component.html**  <div \*ngIf="cart && cart.items.length > 0" 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>  <a>Qty: {{cartItem.quantity}}</a>  </div>  <div>Price: ₹{{ cartItem.price }}</div>  <button class="remove-button" (click)="removeFromCart(cartItem)">Remove</button>  </li>  </ul>    </div>  <div class="checkout">  <div>  <div class="foods-count">{{ cart.items.length }}</div>  <div class="total-price">₹{{ cart.totalprice }}</div>  <button class="btn btn-warning sm-btn" \*ngIf="cart && cart.items.length > 0" routerLink="checkout">Checkout</button>  </div>  </div> |
| **Cart-list.component.css**  .container {  display: flex;  flex-wrap: wrap;    }    ul {  display: flex;  flex-direction: column;  flex: 3 0;  justify-content: space-evenly;  border: 1px solid #ffbbbb;  border-radius: 1rem;  list-style-type: 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;  }    ul li img {  width: 3rem;  height: 3rem;  border-radius: 100rem;  object-fit: cover;  }    ul li div {  padding: 0.3rem;  }    ul li div:not(:first-child) {  flex-basis: 18%;  }    ul li select {  width: 2rem;  outline: none;  border-radius: 0.5rem;  border-bottom: 1px solid lightgrey;  font-size: 1.1rem;  font-weight: 100;  }    ul .remove-button {  border-radius: 0.3rem;  border: none;  padding: 0.1rem;  color: #ca1515;  opacity: 0.7;  outline: none;  }    ul .remove-button:hover {  opacity: 1;  cursor: pointer;  }    .checkout {  display: flex;  flex-direction: column;  justify-content: space-between;  align-items: center;  flex: 1 3;  height: 6rem;  border: 1px solid #ffbbbb;  border-radius: 1rem;  padding: 0.2rem;  margin-top: 1rem;  margin-left: 1rem;  margin-right: 1rem;  color: black;  }    .checkout > div {  font-size: 1rem;  margin: 0.1rem;  flex: 3;  display: flex;  flex-direction: column;  justify-content: center;  align-items: flex-start;  }    .checkout .foods-count {  margin-bottom: 0rem;  }    .checkout .foods-count::before {  content: "Count: ";  color:#ca1515;  }    .checkout .total-price::before {  content: "Price: ";  color: #ca1515;  }  .sm-btn{  padding:3px;  font-size: small;  margin-top: 0.5rem;  }  a{  text-decoration: none;  color: #ca1515;  } |
| **Cart-list.component.ts**  import { Component, OnInit } from '@angular/core';  import { Cart } from 'src/app/models/cart';  import { cartitem } from 'src/app/models/cartitem';  import { CartserviceService } from 'src/app/services/cartservice.service';  @Component({  selector: 'app-cart-list',  templateUrl: './cart-list.component.html',  styleUrls: ['./cart-list.component.css']  })  export class CartListComponent implements OnInit {  cart!: Cart;  constructor(private cartservice:CartserviceService) { }  ngOnInit(): void {  this.updateCart();  }  removeFromCart(cartItem:cartitem){  this.cartservice.removeFromCart(cartItem.food.id);  this.updateCart();  }  changeQuantity(cartItem:cartitem, quantityInString:string){  const quantity = parseInt(quantityInString);  this.cartservice.changeQuantity(cartItem.food.id, quantity);  this.updateCart();  }  updateCart(){  this.cart=this.cartservice.getCart();  }  } |
| **Checkout.component.html**  <button class="btn btn-warning" routerLink="">Go Home</button>  <div class="clear"></div>  <div \*ngIf="cart && cart.items.length > 0" class="container">  <a>Your Cart Items 🛒</a>  <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>  <a>Qty: {{cartItem.quantity}}</a>  </div>  <div>Price: ₹{{ cartItem.price }}</div>    </li>  </ul>    </div>  <div class="checkout">  <div>  <div class="foods-count">{{ cart.items.length }}</div>  <div class="total-price">₹{{ cart.totalprice }}</div>  <ngbd-modal-basic></ngbd-modal-basic>  </div>  </div>  <div class="clear"></div> |
| **Checkout.component.css**  .container {  display: inline;  flex-wrap: wrap;  width:70%;  float:left  }    ul {  display: flex;  flex-direction: column;  flex: 3 0;  justify-content: space-evenly;  border: 1px solid #ffbbbb;  border-radius: 1rem;  list-style-type: 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;  }    ul li img {  width: 3rem;  height: 3rem;  border-radius: 100rem;  object-fit: cover;  }    ul li div {  padding: 0.3rem;  }    ul li div:not(:first-child) {  flex-basis: 18%;  }    ul li select {  width: 2rem;  outline: none;  border-radius: 0.5rem;  border-bottom: 1px solid lightgrey;  font-size: 1.1rem;  font-weight: 100;  }    ul .remove-button {  border-radius: 0.3rem;  border: none;  padding: 0.1rem;  color: #ca1515;  opacity: 0.7;  outline: none;  }    ul .remove-button:hover {  opacity: 1;  cursor: pointer;  }    .checkout {  display: flex;  flex-direction: column;  justify-content: space-between;  align-items: center;  flex: 1 3;  height: 6rem;  border: 1px solid #ffbbbb;  border-radius: 1rem;  padding: 0.2rem;  margin-right: 1rem;  color: black;  width:20%;  float:right;  }    .checkout > div {  font-size: 1rem;  margin: 0.1rem;  flex: 3;  display: flex;  flex-direction: column;  justify-content: center;  align-items: flex-start;  }    .checkout .foods-count {  margin-bottom: 0rem;  }    .checkout .foods-count::before {  content: "Cart Item Count: ";  color:#ca1515;  }    .checkout .total-price::before {  content: "Total Price: ";  color: #ca1515;  }  .sm-btn{  padding:3px;  font-size: small;  margin-top: 0.5rem;  }  a{  text-decoration: none;  color: #ca1515;  }  .clear{  clear: both;  } |
| **Checkout.component.ts**  import { Component, OnInit } from '@angular/core';  import { Cart } from 'src/app/models/cart';  import { CartserviceService } from 'src/app/services/cartservice.service';  @Component({  selector: 'app-checkout',  templateUrl: './checkout.component.html',  styleUrls: ['./checkout.component.css']  })  export class CheckoutComponent implements OnInit {  cart:Cart=new Cart();  constructor(private cartservice:CartserviceService) { }  ngOnInit(): void {  this.cart = this.cartservice.getCart();  }    } |
| **Modal-basic.html**  <ng-template #content let-modal>  <div class="modal-header">  <h4 class="modal-title" id="modal-basic-title">Payment-Gateway</h4>  <button type="button" class="btn-close" aria-label="Close" (click)="modal.dismiss('Cross click')"></button>  </div>  <div class="modal-body">  <a class="payment-confirmation-message">Confirm purchase of cart items for the total amount of:</a><br>  <a class="payment-confirmation-amount" style="font-size: 5rem;text-align: center;">₹{{cart.totalprice}}</a>  </div>  <div class="modal-footer">  <button type="button" class="btn btn-success" (click)="modal.close('Save click')">Confirm Payment</button>  </div>  </ng-template>  <button class="btn btn-warning sm-btn" (click)="open(content)">Pay Amount</button> |
| **Modal-basic.ts**  import { Component } from '@angular/core';  import { Router } from '@angular/router';  import { ModalDismissReasons, NgbDatepickerModule, NgbModal } from '@ng-bootstrap/ng-bootstrap';  import { Cart } from 'src/app/models/cart';  import { CartserviceService } from 'src/app/services/cartservice.service';  @Component({  selector: 'ngbd-modal-basic',  standalone: true,  imports: [NgbDatepickerModule],  templateUrl: './model-basic.html',  })  export class NgbdModalBasic {  closeResult = '';  cart:Cart=new Cart();  constructor(private modalService: NgbModal,  private cartservice:CartserviceService,  private router:Router) {}    open(content:any) {  this.cart=this.cartservice.getCart();  if(!(this.cart.totalprice === 0)){  this.modalService.open(content, { ariaLabelledBy: 'modal-basic-title' }).result.then(  (result) => {  if(result==='Save click'){  alert('Payment of ₹' +this.cart.totalprice + ' confirmed');  this.router.navigateByUrl("");  this.cartservice.resetCart();  }  },  (reason) => {  console.log(reason);  if (reason === ModalDismissReasons.ESC) {  alert('payment cancelled');  } else if (reason === ModalDismissReasons.BACKDROP\_CLICK) {  alert('payment cancelled');  }  else if(reason ==='Cross click'){  alert('payment cancelled');    } else {    }  },  );  }  else{  alert("Kindly add items to cart!!");  }  }    } |
| **Login.component.html**  <button class="btn btn-warning" routerLink="">Go Home</button>  <div class="login-page">  <h3>Admin Login</h3>  <div class="form">  <form class="login-form" [formGroup]="adminLoginForm" (ngSubmit)="loginAdmin()">  <span \*ngIf="email && email.invalid && email.touched">This email is invalid</span>  <input type="email" placeholder="email" formControlName="email" />  <span \*ngIf="password && password.invalid && password.touched">This password is invalid</span>  <input type="password" placeholder="password" formControlName="password" />  <button type="submit" [disabled]="!adminLoginForm.valid">login</button>  </form>  </div>  </div> |
| **Login.component.css**  @import url(https://fonts.googleapis.com/css?family=Roboto:300);  .login-page {  width: 360px;  padding: 1% 0 1%;  margin: auto;  }  /\* .form{  background-color: rgb(87, 70, 70);  } \*/  .form input {  font-family: "Roboto", sans-serif;  outline: 0;  background: #f2f2f2;  width: 100%;  border: 0;  margin: 0 0 15px;  padding: 15px;  box-sizing: border-box;  font-size: 14px;  }  .form button {  font-family: "Roboto", sans-serif;  text-transform: uppercase;  outline: 0;  background: #4caf50;  width: 100%;  border: 0;  padding: 15px;  color: #ffffff;  font-size: 14px;  -webkit-transition: all 0.3 ease;  transition: all 0.3 ease;  cursor: pointer;  }  .form button:hover,  .form button:active,  .form button:focus {  opacity: 90%;  }  body {  font-family: "Roboto", sans-serif;  }  span {  font-size: small;  color: red;  } |
| **Login.component.ts**  import { Component, OnInit } from '@angular/core';  import { FormControl, FormGroup, Validators } from '@angular/forms';  import { Router } from '@angular/router';  import { AdminService } from 'src/app/services/admin.service';  import { AuthService } from 'src/app/services/auth.service';  @Component({  selector: 'app-login',  templateUrl: './login.component.html',  styleUrls: ['./login.component.css']  })  export class LoginComponent implements OnInit {  constructor(private router: Router, private authservice:AuthService) { }  ngOnInit(): void {  }  adminLoginForm = new FormGroup({  email: new FormControl('', [Validators.required, Validators.email]),  password: new FormControl('', [  Validators.required,  Validators.minLength(8),  ]),  });  get email() {  return this.adminLoginForm.get('email');  }  get password() {  return this.adminLoginForm.get('password');  }  loginAdmin() {  if (this.email?.value == this.authservice.adminEmail && this.password?.value===this.authservice.adminPassword) {  this.router.navigateByUrl('admin-home');    } else {    alert(`❌ Incorrect credentials`);  this.router.navigateByUrl('login');  }  }  } |
| **Admin-home.component.html**  <div class="nav">    <div class="nav-title">  <h1>Welcome Admin</h1>  </div>    <div class="nav-menu">  <ul>  <li [routerLink]="['./all-foods']" [routerLinkActive]="['active-link']" >All Foods</li>  <li [routerLink]="['./add-food']" [routerLinkActive]="['active-link']" >Add Food</li>  <li [routerLink]="['./change-password']" [routerLinkActive]="['active-link']" >Change Password</li>  <li><span (click)=" logout()" class="material-symbols-outlined">  logout  </span></li>  </ul>  </div>  </div>  <router-outlet></router-outlet> |
| **Admin-home.component.css**  \*{  padding: 0px;  margin: 0px;  box-sizing: border-box;  }  .nav{  background: black;  color: white;  height: 70px;  display: flex;  flex-wrap:wrap;  align-items: center;  }    .nav-title{  width: 30%;  background: red;  padding: 0;  text-align: center;  -webkit-animation: cssAnimation 3s forwards;  animation: cssAnimation 3s forwards;  }  @keyframes cssAnimation {  0% {opacity: 1; height: auto;}  90% {opacity: 1; height: auto;}  100% {opacity: 0; height: 0;}  }  @-webkit-keyframes cssAnimation {  0% {opacity: 1; height: auto;}  90% {opacity: 1; height: auto;}  100% {opacity: 0; height: 0;}  }  .nav-menu{  width: 70%;  }  .nav-menu ul {  display: flex;  justify-content: flex-end;  }  .active-link{  color: #fff !important;  font-weight: bold;  }  .nav-menu ul li{  padding: 10px;  color: #ccc;  font-size: 20px;  list-style-type: none;  cursor: pointer;  }  .nav-menu ul li:hover {  color: #fff;  }    /\*style for responsive menu\*/  @media screen and (max-width:1000px){  .nav{  height: auto;  }    .nav .nav-title{  text-align: left;  width: 100%;  }  .nav .nav-menu{  width: 100%;  }  .nav .nav-menu ul{  justify-content: space-around;  }  .nav .nav-menu ul li{  width: 20%;  padding: 10px 0px;  text-align: center;  font-size: 16px;  }  } |
| **Admin-home.component.ts**  import { Component, OnInit } from '@angular/core';  import { Router } from '@angular/router';  @Component({  selector: 'app-admin-home',  templateUrl: './admin-home.component.html',  styleUrls: ['./admin-home.component.css']  })  export class AdminHomeComponent implements OnInit {  constructor(private router:Router) { }  ngOnInit(): void {  }  logout() {  this.router.navigateByUrl("");  }  } |
| **Admin-all-foods.component.html**  <h3>Master list of foods</h3>  <div class="table-container">  <table>  <tr>  <th>ID</th>  <th>Name</th>  <th>Cook Time</th>  <th>Price</th>  <th>Origins</th>  <th>Stars</th>  <th>ImageUrl</th>  <th>Tags</th>  <th>Image</th>  <th>Actions</th>  </tr>  <tr \*ngFor="let food of foods">  <td>{{ food.id }}</td>  <td>{{ food.name }}</td>  <td>{{ food.cookTime }}</td>  <td>₹ {{ food.price }}</td>  <td>  <span \*ngFor="let origin of food.origins">  {{ origin }}  </span>  </td>  <td>  <star-rating  [value]="food.stars"  [totalstars]="5"  checkedcolor="red"  uncheckedcolor="black"  size="20px"  [readonly]="true"  ></star-rating>  </td>  <td>{{ food.imageUrl }}</td>  <td>  <span \*ngFor="let tag of food.tags">  {{ tag }}  </span>  </td>  <td><img src="{{ food.imageUrl }}" alt="food image" /></td>  <td class="center">  <span class="material-symbols-outlined" (click)="deleteFood(food)">  delete  </span>  </td>  </tr>  </table>  </div> |
| **Admin-all-foods.component.css**  .table-container{  overflow-x:auto;  }    table {  border-collapse: collapse;  border-spacing: 0;  width: 100%;  border: 1px solid #ddd;  }    th, td {  text-align: left;  padding: 8px;  }    tr:nth-child(even){background-color: #f2f2f2}    img {  width: 3rem;  height: 3rem;  border-radius: 100rem;  object-fit: cover;  }  .center{  text-align: center;  }  .material-symbols-outlined{  cursor: pointer;  } |
| **Admin-all-foods.component.ts**  import { Component, OnInit } from '@angular/core';  import { Food } from 'src/app/models/food';  import { FoodserviceService } from 'src/app/services/foodservice.service';  import { FoodItemListComponent } from '../food-item-list/food-item-list.component';  @Component({  selector: 'app-admin-all-foods',  templateUrl: './admin-all-foods.component.html',  styleUrls: ['./admin-all-foods.component.css']  })  export class AdminAllFoodsComponent implements OnInit {  foods: Food[] = [];  constructor(private foodservice:FoodserviceService) { }  ngOnInit(): void {  this.foods = this.foodservice.getAll();  }  deleteFood(food: Food) {  this.foodservice.deleteFood(food.id);  this.foods = this.foodservice.getAll();  }  } |
| **Admin-add-food.component.html**  <div class="page">  <h3>Add Food Form</h3>  <div class="form">  <form [formGroup]="addFoodForm" (ngSubmit)="addfood()">  <span \*ngIf="id && id.invalid && id.touched">This id is invalid</span>  <input type="number" placeholder="ID" formControlName="id" />  <span \*ngIf="name && name.invalid && name.touched">This name is invalid</span>  <input type="text" placeholder="Name" formControlName="name" />  <span \*ngIf="price && price.invalid && price.touched">This price is invalid</span>  <input type="number" placeholder="Price" formControlName="price" />  <span \*ngIf="cookTime && cookTime.invalid && cookTime.touched">This cookTime is invalid</span>  <input type="text" placeholder="Cook Time (00-00)" formControlName="cookTime" />  <fieldset>  <legend>Origins | <span class="material-symbols-outlined" (click)="addOrigins()">add</span></legend>  <div formArrayName="origins">  <div \*ngFor="let originsControl of getControlsOrigins(); let i = index">  <input type="text" [formControlName]="i" placeholder="Origins">  <span class="material-symbols-outlined" (click)="deleteOrigins(i)">delete</span>  </div>  </div>  </fieldset>  <span \*ngIf="stars && stars.invalid && stars.touched">This stars is invalid</span>  <input type="number" placeholder="Stars" formControlName="stars" />  <span \*ngIf="imageUrl && imageUrl.invalid && imageUrl.touched">This imageUrl is invalid</span>  <input type="text" placeholder="Image URL" formControlName="imageUrl" />  <fieldset>  <legend>Tags | <span class="material-symbols-outlined" (click)="addTags()">add</span></legend>  <div formArrayName="tags">  <div \*ngFor="let tagsControl of getControls(); let i = index">  <input type="text" [formControlName]="i" placeholder="Tags">  <span class="material-symbols-outlined" (click)="deleteTags(i)">delete</span>  </div>  </div>  </fieldset>  <input type="hidden" placeholder="Favorite" formControlName="favorite" />  <button type="submit" [disabled]="!addFoodForm.valid">Add Food</button>  </form>  </div>  </div> |
| **Admin-add-food.component.css**  @import url(https://fonts.googleapis.com/css?family=Roboto:300);  .page {  width: 360px;  padding: 1% 0 0;  margin: auto;  }  .form input {  font-family: "Roboto", sans-serif;  outline: 0;  background: #f2f2f2;  width: 93%;  border: 0;  margin: 0 0 15px;  padding: 10px;  box-sizing: border-box;  font-size: 14px;  }  .form button {  font-family: "Roboto", sans-serif;  text-transform: uppercase;  outline: 0;  background: #4caf50;  width: 93%;  border: 0;  padding: 10px;  color: #ffffff;  font-size: 14px;  -webkit-transition: all 0.3 ease;  transition: all 0.3 ease;  cursor: pointer;  }  .form button:hover,  .form button:active,  .form button:focus {  opacity: 90%;  }  body {  font-family: "Roboto", sans-serif;  }  span {  font-size: small;  color: red;  }  .material-symbols-outlined {  font-size: large;  cursor: pointer;  color: #4caf50;  border: 0.1px solid black;  border-radius: 100%;  }  legend {  font-size: 14px;  color: gray;  }  fieldset {  outline: 0;  background: #f2f2f2;  width: 93%;  border-color: #ffffff;  margin: 0 0 15px;  padding: 10px;  box-sizing: border-box;  font-size: 14px  } |
| **Admin-add-food.component.ts**  import { Component, OnInit } from '@angular/core';  import { FormArray, FormControl, FormGroup, Validators } from '@angular/forms';  import { Router } from '@angular/router';  import { FoodserviceService } from 'src/app/services/foodservice.service';  @Component({  selector: 'app-admin-add-food',  templateUrl: './admin-add-food.component.html',  styleUrls: ['./admin-add-food.component.css']  })  export class AdminAddFoodComponent implements OnInit {  addFoodForm!: FormGroup;  constructor(private router: Router,private foodService:FoodserviceService) { }  ngOnInit(): void {  this.addFoodForm = new FormGroup({  id: new FormControl('', [Validators.required]),  name: new FormControl('', [Validators.required]),  price: new FormControl('', [Validators.required]),  tags: new FormArray([]),  stars: new FormControl('', [Validators.required]),  imageUrl: new FormControl('', [Validators.required]),  origins: new FormArray([]),  cookTime: new FormControl('', [Validators.required]),  favorite: new FormControl(false),  });  }  getControls() {  return (this.addFoodForm.get('tags') as FormArray).controls;  }  // }  addTags() {  (<FormArray>this.addFoodForm.get('tags')).push(new FormControl(''));  }  deleteTags(index: number) {  (<FormArray>this.addFoodForm.get('tags')).removeAt(index);  }  getControlsOrigins() {  return (this.addFoodForm.get('origins') as FormArray).controls;  }  addOrigins() {  (<FormArray>this.addFoodForm.get('origins')).push(new FormControl(''));  }  deleteOrigins(index: number) {  (<FormArray>this.addFoodForm.get('origins')).removeAt(index);  }  get id() {  return this.addFoodForm.get('id');  }  get name() {  return this.addFoodForm.get('name');  }  get price() {  return this.addFoodForm.get('price');  }  get tags() {  return this.addFoodForm.get('tags');  }  get stars() {  return this.addFoodForm.get('stars');  }  get imageUrl() {  return this.addFoodForm.get('imageUrl');  }  get origins() {  return this.addFoodForm.get('origins');  }  get cookTime() {  return this.addFoodForm.get('cookTime');  }  addfood(): void {  console.log(this.addFoodForm.value);  this.foodService.addfood(this.addFoodForm.value);  this.router.navigateByUrl('/admin-home/all-foods');  }  } |
| **Foodservice.service.ts**  import { Injectable } from '@angular/core';  import { Food } from '../models/food';  import { Tag } from '../models/tags';  @Injectable({  providedIn: 'root'  })  export class FoodserviceService {  constructor() {}  getFoodById(id: number): Food {  return this.getAll().find((food) => food.id == id)!;  }  getAllFoodsBySearchTerm(searchTerm: string): Food[] {  return this.getAll().filter((food) =>  food.name.toLowerCase().includes(searchTerm.toLowerCase())  );  }  getAllTags(): Tag[] {  return [  { name: 'All', count: 14 },  { 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 },  ];  }  getAllFoodsByTag(tag: string): Food[] {  return tag == 'All'  ? this.getAll()  : this.getAll().filter((food) => food.tags?.includes(tag));  }    deleteFood(foodId: number): void {  this.food = this.food.filter((item) => item.id != foodId);  }    addfood(newFood: any) {  this.food.push(newFood);  }  getAll(): Food[] {  return this.food;  }  private food: Food[] = [  {  id: 1,  name: 'Pizza Pepperoni',  cookTime: '10-20',  price: 1000,  favorite: false,  origins: ['italy'],  stars: 4.5,  imageUrl: '/assets/images/foods/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/images/foods/food-2.jpg',  tags: ['SlowFood', 'Lunch'],  },  {  id: 3,  name: 'Hamburger',  price: 550,  cookTime: '10-15',  favorite: false,  origins: ['germany', 'us'],  stars: 3.5,  imageUrl: '/assets/images/foods/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/images/foods/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/images/foods/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/images/foods/food-6.jpg',  tags: ['FastFood', 'Pizza', 'Lunch'],  },  {  id: 7,  name: 'Chicken Burger',  price: 9,  cookTime: '10-15',  favorite: false,  origins: ['germany', 'us'],  stars: 4.2,  imageUrl: '/assets/images/foods/food-7.jpg',  tags: ['FastFood', 'Burger', 'Lunch'],  },  {  id: 8,  name: 'Pizza',  price: 9,  cookTime: '40-50',  favorite: false,  origins: ['italy'],  stars: 4.5,  imageUrl: '/assets/images/foods/food-8.jpg',  tags: ['FastFood', 'Pizza', 'Lunch'],  },  ];  } |
| **Cartservice.service.ts**  import { Injectable } from '@angular/core';  import { Food } from '../models/food';  import { Tag } from '../models/tags';  @Injectable({  providedIn: 'root'  })  export class FoodserviceService {  constructor() {}  getFoodById(id: number): Food {  return this.getAll().find((food) => food.id == id)!;  }  getAllFoodsBySearchTerm(searchTerm: string): Food[] {  return this.getAll().filter((food) =>  food.name.toLowerCase().includes(searchTerm.toLowerCase())  );  }  getAllTags(): Tag[] {  return [  { name: 'All', count: 14 },  { 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 },  ];  }  getAllFoodsByTag(tag: string): Food[] {  return tag == 'All'  ? this.getAll()  : this.getAll().filter((food) => food.tags?.includes(tag));  }    deleteFood(foodId: number): void {  this.food = this.food.filter((item) => item.id != foodId);  }    addfood(newFood: any) {  this.food.push(newFood);  }  getAll(): Food[] {  return this.food;  }  private food: Food[] = [  {  id: 1,  name: 'Pizza Pepperoni',  cookTime: '10-20',  price: 1000,  favorite: false,  origins: ['italy'],  stars: 4.5,  imageUrl: '/assets/images/foods/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/images/foods/food-2.jpg',  tags: ['SlowFood', 'Lunch'],  },  {  id: 3,  name: 'Hamburger',  price: 550,  cookTime: '10-15',  favorite: false,  origins: ['germany', 'us'],  stars: 3.5,  imageUrl: '/assets/images/foods/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/images/foods/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/images/foods/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/images/foods/food-6.jpg',  tags: ['FastFood', 'Pizza', 'Lunch'],  },  {  id: 7,  name: 'Chicken Burger',  price: 9,  cookTime: '10-15',  favorite: false,  origins: ['germany', 'us'],  stars: 4.2,  imageUrl: '/assets/images/foods/food-7.jpg',  tags: ['FastFood', 'Burger', 'Lunch'],  },  {  id: 8,  name: 'Pizza',  price: 9,  cookTime: '40-50',  favorite: false,  origins: ['italy'],  stars: 4.5,  imageUrl: '/assets/images/foods/food-8.jpg',  tags: ['FastFood', 'Pizza', 'Lunch'],  },  ];  } |
| **Auth.service.ts**  import { Injectable } from '@angular/core';  @Injectable({  providedIn: 'root'  })  export class AuthService {  constructor() { }  adminEmail: string = 'admin@gmail.com';  adminPassword: any = 'admin123';  } |
| **Admin.service.ts**  import { Injectable } from '@angular/core';  @Injectable({  providedIn: 'root'  })  export class AdminService {  user:string = 'Guest';  constructor() { }  setCurrentUser(usr:any){  this.user=usr;  }  getCurrentUser():string{  return this.user;  }  } |
| **Tags.ts**  export class Tag{  name!:string;  count!:number;  } |
| **Food.ts**  export class Food{  id!:number;  name!:string;  price!:number;  tags?:string[];  favorite:boolean = false;  stars:number = 0;  imageUrl!:string;  origins!:string[];  cookTime!:string;  } |
| **Cartitem.ts**  import { Food } from "./food";  export class cartitem{  constructor(food:Food){  this.food=food;  }  food:Food;  quantity:number =1;  get price():number{  return this.food.price\*this.quantity;  }  } |
| **Cart.ts**  import { cartitem } from "./cartitem";  export class Cart{  items:cartitem[]=[];  get totalprice():number{  let totalprice:number = 0;  this.items.forEach(item=>{totalprice+=item.price});  return totalprice;  }  } |