

File - C:\Users\gianni\Documents\COD\isl-capstoneproject-foodbox\src\main\resources\frontend\foodbox-app\src\app.module.ts

```
1 import {NgModule} from '@angular/core';
2 import {BrowserModule} from '@angular/platform-browser';
3 import {RouterModule} from '@angular/router';
4 import {AppComponent} from './app.component';
5 import {LoginComponent} from './login/login.component';
6 import {SearchComponent} from './search/search.component';
7 import {FormsModule, ReactiveFormsModule} from '@angular/forms';
8 import {HttpClientModule} from '@angular/common/http';
9 import {CartComponent} from './cart/cart.component';
10 import {CheckoutComponent} from './checkout/checkout.component';
11 import {InventoryComponent} from './inventory/inventory.component';
12 import {ChangeComponent} from './login/change.component';
13 import {DemoComponent} from './demo/demo.component';
14 import {BrowserAnimationsModule} from '@angular/platform-browser/animations';
15 import {ToastModule} from 'ngx-toastr';
16 import {OrdenComponent} from './orden/orden.component';
17 import {LogoutComponent} from './login/logout.component';
18 import {EditComponent} from './search/edit.component';
19 import {DeleteComponent} from './search/delete.component';
20 import {NewComponent} from './search/new.component';
21 import {AboutComponent} from './about/about.component';
22
23 @NgModule({
24   declarations: [
25     AppComponent,
26     LoginComponent,
27     SearchComponent,
28     CartComponent,
29     CheckoutComponent,
30     InventoryComponent,
31     ChangeComponent,
32     DemoComponent,
33     OrdenComponent,
34     LogoutComponent,
35     EditComponent,
36     DeleteComponent,
37     NewComponent,
38     AboutComponent
39   ],
40   imports: [
41     BrowserModule,
42     HttpClientModule,
43     BrowserAnimationsModule,
44     ToastModule.forRoot(),
45     RouterModule.forRoot([
46       {path: '', component: SearchComponent},
47       {path: 'demo', component: DemoComponent},
48       {path: 'search', component: SearchComponent},
49       {path: 'ordens', component: OrdenComponent},
```

File - C:\Users\giann\Documents\COD\sl-capstoneproject-foodbox\src\main\resources\frontend\foodbox-app\src\app\app.module.ts

```
50     {path: 'inventory', component: InventoryComponent},
51     {path: 'cart', component: CartComponent},
52     {path: 'checkout', component: CheckoutComponent},
53     {path: 'logout', component: LogoutComponent},
54     {path: 'login', component: AppComponent},
55     {path: 'editFood/:id', component: EditComponent},
56     {path: 'deleteFood/:id', component: DeleteComponent},
57     {path: 'newFood', component: NewComponent},
58     {path: 'about', component: AboutComponent},
59   ]),
60   FormsModule,
61   ReactiveFormsModule
62 ],
63 providers: [],
64 bootstrap: [AppComponent]
65 })
66 export class AppModule {
67 }
68
69
70
```

File - C:\Users\giann\Documents\COD\src\capstoneproject-foodbox\src\main\resources\frontend\foodbox-app\src\app\app.component.ts

```
1 import {Component, OnInit} from '@angular/core';
2 import {LoginService} from './login/login.service';
3 import {Router} from "@angular/router";
4 import {UntypedFormBuilder, Validators} from '@angular/forms';
5 import {IOrden} from "./model/orden";
6
7
8 @Component({
9   selector: 'fb-root',
10  templateUrl: './app.component.html',
11  styleUrls: ['./app.component.css']
12 })
13
14 export class AppComponent implements OnInit {
15   pageTitle = 'FoodBox';
16   activeTab = 'search';
17   isAdmin: boolean = false;
18   isLoggedIn: boolean = false;
19
20   loginForm = this.formBuilder.group({
21     username: ['mike', Validators.required],
22     password: ['chicago2022', Validators.required]
23   })
24
25   constructor(private loginService: LoginService,
26               private router: Router,
27               private formBuilder: UntypedFormBuilder) {
28   }
29
30   onSubmit(): void {
31
32     this.loginService.submitLogin(this.loginForm.value).subscribe({
33       next: (data) => {
34         data.profile !== "F"?this.isLoggedIn = true:this.isLoggedIn = false;
35         data.profile === "A"?this.isAdmin = true:this.isAdmin = false;
36
37         this.getLoginService().loggedIn = this.isLoggedIn;
38         this.getLoginService().admin = this.isAdmin;
39         this.getLoginService().id = data.id;
40         this.getLoginService().name = data.name;
41         this.getLoginService().contact = data.contact;
42         this.getLoginService().credit = data.credit;
43
44       },
45       error: (err => console.error("ERROR: " + err))
46     });
47   }
48
49   getLoginService(): LoginService {
```

File - C:\Users\giann\Documents\COD\isl-capstoneproject-foodbox\src\main\resources\frontend\foodbox-app\src\app.component.ts

```
50     return this.loginService;
51   }
52
53   getUsername(): string{
54
55     // @ts-ignore
56     return this.loginForm.get('username').value;
57   }
58
59   getUsernameClass(): string{
60
61     return this.getLoginService().admin?"bg-secondary text-light adminBorder":"bg-warning text-light";
62   }
63
64   getUsernameProfile(): string{
65
66     return this.getLoginService().admin?" - ADMIN":"";
67   }
68
69   search(activeTab: string) {
70     this.activeTab = activeTab;
71   }
72
73   result(activeTab: string) {
74     this.activeTab = activeTab;
75   }
76
77   ngOnInit(): void {
78   }
79
80
81 }
82
```

File - C:\Users\giann\Documents\COD\sl-capstoneproject-foodbox\src\main\resources\frontend\foodbox-app\src\app\app.component.css

```
1 .active {  
2   background-color: yellow;  
3 }
```


File - C:\Users\giann\Documents\CODE\sl-capstoneproject-foodbox\src\main\resources\frontend\foodbox-app\src\app.component.html

```
50     <i aria-hidden="true" class="fa fa-search"></i>&nbsp;&nbsp;&nbsp;</a></a>
51     <a *ngIf="!getLoginService().admin" [routerLink]="['/cart']" class='nav-link cell-gray shadow-0'
52     routerLinkActive='active'>
53         <i aria-hidden="true" class="fa fa-shopping-cart"></i>&nbsp;&nbsp;&nbsp;</a></a>
54         <i aria-hidden="true" class="fa fa-credit-card"></i>&nbsp;&nbsp;&nbsp;</a></a>
55         <a *ngIf="!getLoginService().admin" [routerLink]="['/checkout']" class='nav-link cell-gray shadow-0'
56         routerLinkActive='active'>
57             <i aria-hidden="true" class="fa fa-credit-card"></i>&nbsp;&nbsp;&nbsp;</a></a>
58             <a *ngIf="!getLoginService().admin" [routerLink]="['/orders']" class='nav-link cell-gray shadow-0'
59             routerLinkActive='active'>
60                 <i aria-hidden="true" class="fa fa-edit"></i>&nbsp;&nbsp;&nbsp;</a></a>
61                 <a [routerLink]="['/logout']" class='nav-link cell-gray shadow-0' routerLinkActive='active'>
62                     <i aria-hidden="true" class="fa fa-sign-in"></i>&nbsp;&nbsp;&nbsp;</a></a>
63                     <a [routerLink]="['/about']" class='nav-link cell-gray shadow-0' routerLinkActive='active'>
64                         <i aria-hidden="true" class="fa fa-question"></i>&nbsp;&nbsp;&nbsp;</a></a>
65         </div>
66         <div class="col-10">
67             <router-outlet></router-outlet>
68         </div>
69     </div>
70 </div>
71
72 <ng-template #loggedOut>
73     <div class="row" style="width:100%; margin-top: 3rem;">
74         <div class="cell col-4"></div>
75         <div class="cell col-4">
76             <div class="card shadow-1">
77                 <div class="card-header">
78                     Welcome!
79                 </div>
80                 <div class="card-body">
81                     <h5 class="card-title">Login to FoodBox!</h5>
82                     <div class="gapToplem">
83                         <form (ngSubmit)="onSubmit()" [formGroup]="loginForm">
84                             <div class="cell row d-flex justify-content-end align-items-center">
85                                 <div class="cell col-1 text-end">
86                                     &nbsp;&nbsp;&nbsp;
87                                 </div>
88                                 <div class="cell col-3 text-end">
89                                     <label class="cell col-form-label-sm text-end " for="username">Username:</label>
90                                 </div>
91                                 <div class="cell col-5">
92                                     <input type="text"
93                                         class="form-control"
94                                         id="username"
95                                         placeholder="username"
96                                         formControlName="username"
97                                         value="">
98                                 </div>

```

File - C:\Users\giann\Documents\CODE\sl-capstoneproject-foodbox\src\main\resources\frontend\foodbox-app\src\app.component.html

```
99     <div class="cell col-3" style="position: relative">
100         
102     </div>
103 </div>
104 <div class="cell row d-flex justify-content-end align-items-center">
105     <div class="cell col-1 text-end">
106         &nbsp;
107     </div>
108     <div class="cell col-3 text-end">
109         <label class="cell col-form-label-sm text-end" for="password">Password:</label>
110     </div>
111     <div class="cell col-5 ">
112         <input type="password"
113             class="form-control"
114             id="password"
115             placeholder="password"
116             formControlName="password"
117             value="" />
118     </div>
119     <div class="cell col-3">
120
121     </div>
122 </div>
123 <div class="cell row">
124     <div class="col-1">
125         &nbsp;
126     </div>
127     <div class="col-3">
128         &nbsp;
129     </div>
130     <div class="col-5 d-flex justify-content-end">
131         <button class="btn btn-primary gapTopIem" type="submit">Login</button>
132     </div>
133     <div class="cell col-3">
134
135     </div>
136 </form>
137 </div>
138 </div>
139 </div>
140 <div class="cell col-4"></div>
141 </div>
142 <div class="cell-blue row" style="margin-top: 60px; width: 100%">
143     <div class="cell col">&nbsp;</div>
144     <div class="cell col-2">
145         
146     </div>
147
```


File - C:\Users\giann\Documents\COD\sl-capstoneproject-foodbox\src\main\resources\frontend\foodbox-app\src\app.component.html

```
148 <div class="cell col-2">
149   
150 </div>
151 <div class="cell col-2">
152   
153 </div>
154 <div class="cell col">&nbsp;</div>
155 </div>
156 </ng-template>
157
158
```

File - C:\Users\giann\Documents\COD\tsl-capstoneproject-foodbox\src\main\resources\frontend\foodbox-app\src\app\cart\cart.component.ts

```
1 import { Component, OnInit } from '@angular/core';
2 import { SearchService } from "../search/search.service";
3 import { FormBuilder } from "@angular/forms";
4 import { IFood } from "../model/food";
5
6 @Component({
7   templateUrl: './cart.component.html',
8   styleUrls: ['./cart.component.css']
9 })
10 export class CartComponent implements OnInit {
11   cart: IFood[] = [];
12   total: number = this.searchService.absoluteTotal;
13   constructor(private searchService: SearchService) { }
14
15   ngOnInit(): void {
16     this.cart = this.searchService.getCart();
17     this.searchService.getTotal();
18     this.total = this.searchService.getTotal()
19   }
20
21   removeItem(food: IFood) {
22     this.searchService.removeItem(food)
23     this.total = this.searchService.getTotal()
24   }
25
26   RandomTilt() {
27     let tilt: number = 0;
28     tilt = Math.floor(Math.random() * 5)+1;
29     return "deg-" + tilt
30   }
31 }
32
```

File - C:\Users\giann\Documents\CODE\is-capstoneproject-foodbox\src\main\resources\frontend\foodbox-app\src\app\cart\cart.component.html

```
1 <div class="card">
2   <div class="card-header">
3     Your Cart
4   </div>
5   <div class="card-body mt-3 mb-4">
6     <h5 class="card-title">Following is a detailed list of your orders</h5>
7     <p class="card-text">Thank you for shopping with us. Please review your orders and their status. When it is
8       ready you can pick it up in the front. Enjoy your meal!</p>
9   </div>
10
11   <div class="cell container">
12     <div class="cell row">
13       <div class="cell col-12">
14         <div class="cell row bg-light border rounded p-1">
15           <div class="cell col-1 text-center"></div>
16           <div class="cell col-2 text-center">Dish Name</div>
17           <div class="cell col-2 text-center">&nbsp;</div>
18           <div class="cell col-1 text-center">Cuisine</div>
19           <div class="cell col-4 text-center">Description</div>
20           <div class="cell col-1 text-center">Price</div>
21           <div class="cell col-1 text-center"></div>
22         </div>
23         <p *ngIf="cart.length<=0" class="card-text text-center mt-1 mb-1"> - empty - </p>
24         <div *ngFor="let food of cart" class="cell row border rounded p-1">
25           <div class="cell col-1 d-flex align-items-center">
26             <button (click)="removeItem(food)"
27               class="cell btn btn-outline-primary btn-sm alert-secondary ">
28               Delete
29             </button>
30           </div>
31           <span hidden>{{food.id}}</span>
32           <div class="cell col-2 d-flex align-items-center">{{food.foodName}}</div>
33           <div class="cell col-2 d-flex align-items-center">
34             
35           </div>
36           <div class="cell col-1 d-flex align-items-center"><div class="cell text-centered">{{food.cuisine}}</div>
37           <div class="cell col-4 d-flex align-items-center">{{food.description}}</div>
38           <div class="cell col-1 d-flex align-items-center"><div class="cell text-centered">{{food.price|currency}}</div></div>
39         </div>
40         &nbsp;
41       </div>
42     </div>
43   </div>
44 </div>
45 </div>
46 </div>
47
```

File - C:\Users\giann\Documents\COD\src\capstoneproject-foodbox\src\main\resources\frontend\foodbox-app\src\app\demo\demo.component.ts

```
1 import {Component, OnInit} from '@angular/core';
2
3 @Component({
4   selector: 'ks-demo',
5   templateUrl: './demo.component.html',
6   styleUrls: ['./demo.component.css']
7 })
8 export class DemoComponent implements OnInit {
9
10   constructor() {
11   }
12
13   ngOnInit(): void {
14
15   }
16
17 }
18
```

File - C:\Users\giann\Documents\COD\lsl-capstoneproject-foodbox\src\main\resources\frontend\foodbox-app\src\app\demo\demo.component.html

```
1 <br /><h2>FoodBox is a Full Stack application written in Angular and Springboot.</h2>
2 <p>This web application is the Capstone Project for my bootcamp.</p>
3
4 
```

File - C:\Users\giann\Documents\COD\lsl-capstoneproject-foodbox\src\main\resources\frontend\foodbox-app\src\app\about\about.component.ts

```
1 import { Component, OnInit } from '@angular/core';
2
3 @Component({
4   selector: 'ks-about',
5   templateUrl: './about.component.html',
6   styleUrls: ['./about.component.css']
7 })
8 export class AboutComponent implements OnInit {
9
10  constructor() { }
11
12  ngOnInit(): void {
13  }
14
15 }
16
```

File - C:\Users\giann\Documents\CODE\is1-capstoneproject-foodbox\src\main\resources\frontend\foodbox-app\src\app\about\about.component.html

```
1 <div class="cell card" style="width: 70%; ">
2   <div class="cell card-header">
3     About This Application
4   </div>
5   <div class="cell card-body mb-1 ">
6     <div class="ml-2 mx-2" style="width: 70%; ">
7       <h3 class="mt-3">The Bootcamp</h3>
8     <p>Food Box is the Capstone Project for my Full Stack Java Bootcamp. It is the culmination of many months of learning in the
evenings and
9       on weekends. The Bootcamp started in December
10      2021 and I am delivering this final project at the end of October 2022, so it took almost 11 months to complete.
11    <b><span style="display:inline-block">NO START-CODE WAS PROVIDED</span></b></p>
12    rewarding!</p>
13    <h3 class="mt-5">Future Improvements</h3>
14    <p>Currently, the application stores data in MySQL database, but the goal is to migrate to MongoDB, a NO_SQL
15    document-based database. Also, it would be a good challenge to migrate it from Amazon AWS to Microsoft Azure.
16    Finally, a phone-friendly application is a must. It should not take much effort to modify the CSS files and make
17    a mobile version.</p>
18
19    <h3 class="mt-5">The Application</h3>
20    <p>Angular is used for the frontend and Springboot + MySQL for the backend. It runs in the Amazon AWS cloud, specifically the
21    backend runs in the Amazon Beanstalk and the frontend in S3 static storage.</p>
22  </div>
23 </div>
24 
25 </div>
26 </div>
27
28
29 </div>
30
```

File - C:\Users\giann\Documents\COD\Isl-capstoneproject-foodbox\src\main\resources\frontend\foodbox-app\src\app\about\about.component.spec.ts

```
1 import { ComponentFixture, TestBed } from '@angular/core/testing';
2
3 import { AboutComponent } from './about.component';
4
5 describe('AboutComponent', () => {
6   let component: AboutComponent;
7   let fixture: ComponentFixture<AboutComponent>;
8
9   beforeEach(async () => {
10     await TestBed.configureTestingModule({
11       declarations: [ AboutComponent ]
12     })
13     .compileComponents();
14
15     fixture = TestBed.createComponent(AboutComponent);
16     component = fixture.componentInstance;
17     fixture.detectChanges();
18   });
19
20   it('should create', () => {
21     expect(component).toBeTruthy();
22   });
23 });
24
```


File - C:\Users\giann\Documents\CODE\sl-capstoneproject-foodbox\src\main\resources\frontend\foodbox-app\src\app\login\login.service.ts

```
1 import {Injectable} from '@angular/core';
2 import {HttpClient, HttpResponse, HttpErrorResponse, HttpHeaders} from '@angular/common/http';
3 import {Observable, throwError} from 'rxjs';
4 import {catchError, tap} from 'rxjs/operators';
5 import {IUser} from '../model/user';
6
7 @Injectable({
8   providedIn: 'root'
9 })
10 export class LoginService {
11   baseUrl: string = 'http://foodboxapp-env.eba-5ye8smp8.us-west-2.elasticbeanstalk.com'
12   userUrl: string = this.baseUrl + '/login';
13   putUrl: string = 'https://62e8570a249bb1284ead379a.mockapi.io/api/v1/users/';
14   private isAdmin: boolean = false;
15   private isLoggedIn: boolean = false;
16   private _id!: number;
17   private _name!: string;
18   private _contact!: string;
19   private _credit!: string;
20
21
22
23
24
25
26   constructor(private http: HttpClient) {
27
28   }
29
30   set credit(credit: string) {
31     this._credit = credit;
32   }
33
34   get credit(): string {
35     return this._credit;
36   }
37
38   set contact(contact: string) {
39     this._contact = contact;
40   }
41
42   get contact(): string {
43     return this._contact;
44   }
45
46   set name(name: string) {
47     this._name = name;
48   }
49
50   get name(): string {
51     return this._name;
52   }
53 }
```

File - C:\Users\giann\Documents\COD\isl-capstoneproject-foodbox\src\main\resources\frontend\foodbox-app\src\app\login\login.service.ts

```
50 set id(id: number) {
51   this._id = id;
52 }
53
54 get id(): number {
55   return this._id;
56 }
57
58 set admin(isAdmin: boolean) {
59   this.isAdmin = isAdmin;
60 }
61
62 get admin(): boolean {
63   return this.isAdmin;
64 }
65
66 set loggedIn(isLoggedIn: boolean) {
67   this.isLoggedIn = isLoggedIn;
68 }
69
70 get loggedIn(): boolean {
71   return this.isLoggedIn;
72 }
73
74 submitLogin(formValue: any): Observable<IUser> {
75   const httpOptions = {
76     headers: new HttpHeaders({'Accept': 'application/json', 'Content-Type': 'application/json'})
77   };
78
79   return this.http.post<IUser>(this.userUrl, formValue, httpOptions).pipe(
80     tap(data => {{data}}),
81     catchError(err => this.handleError(err))
82   )
83 }
84
85 changePassword(formValue: any): Observable<any> {
86   return this.http.put(this.putUrl + "/" + formValue.idx, formValue).pipe(
87     tap(data => data),
88     catchError(err => this.handleError(err))
89   );
90 }
91
92 handleError(error: HttpErrorResponse) {
93   let errorMessage
94   if (error.error instanceof ErrorEvent) {
95     errorMessage = 'An error occurred: ${error.error.message}';
96   } else {
97     errorMessage = 'Server returned code: ${error.status}, error message is: ${error.message}';
98   }
```

File - C:\Users\giann\Documents\COD\isl-capstoneproject-foodbox\src\main\resources\frontend\foodbox-app\src\app\login\login.service.ts

```
99     console.error(errorMessage);
100     alert(`Login Error. \n\nVerify your username and password are correct.\nAlso verify you use the link provided for the
    application:\nhttp://foodbox.s3.us-west-2.amazonaws.com/index.html (http, not https)`);
101     return throwError(errorMessage);
102   }
103 }
104 }
105 }
```

File - C:\Users\giann\Documents\COD\sl-capstoneproject-foodbox\src\main\resources\frontend\foodbox-app\src\app\login\login.component.ts

```
1 import {Component, OnInit} from '@angular/core';
2 import {UntypedFormBuilder} from "@angular/forms";
3 import {LoginService} from "../login.service";
4 import {Router} from "@angular/router";
5 import {InventoryService} from "../inventory/inventory.service";
6
7
8 @Component({
9   templateUrl: '',
10  styleUrls: ['./login.component.css']
11 })
12 export class LoginComponent implements OnInit {
13
14   loginForm = this.formBuilder.group({
15     username: '',
16     password: ''
17   })
18
19   constructor(private FormBuilder: UntypedFormBuilder,
20               private loginService: LoginService,
21               private router: Router,
22               private inventoryService: InventoryService) {
23
24   }
25
26   ngOnInit(): void {
27   }
28
29   onSubmit() {
30
31   }
32
33   onChangePassword() {
34
35   }
36 }
37
```

File - C:\Users\giann\Documents\CODE\sl-capstoneproject-foodbox\src\main\resources\frontend\foodbox-app\src\app\login\change.component.ts

```
1 import {Component, OnInit} from '@angular/core';
2 import {UntypedFormBuilder} from "@angular/forms";
3 import {LoginService} from "../login.service";
4 import {Router} from "@angular/router";
5
6 @Component({
7   selector: 'ks-change',
8   templateUrl: './change.component.html',
9   styleUrls: ['./change.component.css']
10 })
11 export class ChangeComponent implements OnInit {
12
13   loginGroup = this.formBuilder.group({
14     idx: '1',
15     username: 'admin',
16     oldPassword: 'admin',
17     newPassword: '',
18     newPassword2: ''
19   })
20
21   constructor(private formBuilder: UntypedFormBuilder,
22               private loginService: LoginService,
23               private router: Router) {
24
25   }
26
27   ngOnInit(): void {
28   }
29
30   onSubmit(): void {
31
32     if (this.loginGroup.value.newPassword !== this.loginGroup.value.newPassword2) {
33       alert("New Password does not match. Please try again.")
34     } else {
35       this.loginService.changePassword(this.loginGroup.value).subscribe({
36         next: (data: any) => {
37           const parsed = JSON.parse(JSON.stringify(data))
38           let retVal = parsed.return
39           if (retVal === ("OK")) {
40             alert("Password successfully changed")
41             this.router.navigate(["/inventory"])
42           }
43         },
44         error: err => {
45           console.error(">>> " + err)
46           alert("error: >>> " + err);
47         }
48       })
49     }
50   }
```

File - C:\Users\giann\Documents\COD\sl-capstoneproject-foodbox\src\main\resources\frontend\foodbox-app\src\app\login\change.component.ts

50 }

51 }

52 }

53

File - C:\Users\giann\Documents\COD\src\capstoneproject-foodbox\src\main\resources\frontend\foodbox-app\src\app\login\logout.component.ts

```
1 import { Component, OnInit } from '@angular/core';
2 import { Router } from '@angular/router';
3
4 @Component({
5   selector: 'ks-logout',
6   templateUrl: './logout.component.html',
7   styleUrls: ['./logout.component.css']
8 })
9 export class LogoutComponent implements OnInit {
10
11   constructor(private router: Router) { }
12
13   ngOnInit(): void {
14     //wait 3 sec
15     setTimeout(() => {
16       window.location.href="http://foodbox.s3.us-west-2.amazonaws.com/index.html";
17     }, 1000);
18   }
19
20 }
21
```

File - C:\Users\giann\Documents\COD\isl-capstoneproject-foodbox\src\main\resources\frontend\foodbox-app\src\app\login\logout.component.css

```
1 .shadow-0{
2     box-shadow: 0 0px 1px 0 rgba(0, 0, 0, 0.1), 0 3px 3px 0
3     rgba(0, 0, 0, 0.1);
4 }
5
6 .shadow-1{
7     box-shadow: 0 1px 2px 0 rgba(0, 0, 0, 0.2), 0 6px 20px 0 rgba(0, 0, 0, 0.19);
8 }
9
10 .roundedPicture{
11     border-radius: 5% !important;
12     opacity: 0.2;
13     box-shadow: 0 1px 2px 0 rgba(0, 0, 0, 0.2), 0 6px 20px 0 rgba(0, 0, 0, 0.19);
14 }
```


File - C:\Users\giann\Documents\COD\is1-capstoneproject-foodbox\src\main\resources\frontend\foodbox-app\src\app\login\change.component.html

```
1 <div class="row">
2   <div class="col-2"></div>
3   <div class="col-6">
4     <div class="card">
5       <div class="card-header">
6         Password Change
7       </div>
8       <div class="card-body">
9         <h5 class="card-title">Change Admin Login</h5>
10
11         <form (ngSubmit)="onSubmit()" [formGroup]="loginGroup">
12           <div class="mb-3">
13             <label class="form-label" for="idx">ID</label>
14             <input class="form-control" formControlName="idx" id="idx" placeholder="ID"
15               type="text">
16           </div>
17           <div class="mb-3">
18             <label class="form-label" for="username">Username</label>
19             <input class="form-control" formControlName="username" id="username"
20               placeholder="username"
21               type="text">
22           </div>
23           <div class="mb-3">
24             <label class="form-label" for="oldPassword">Old Password</label>
25             <input class="form-control" formControlName="oldPassword" id="oldPassword"
26               placeholder="old password"
27               type="password">
28           </div>
29           <div class="mb-3">
30             <label class="form-label" for="newPassword2">New Password</label>
31             <input class="form-control" formControlName="newPassword" id="newPassword"
32               placeholder="type your new password"
33               type="password">
34           </div>
35           <div class="mb-4">
36             <label class="form-label" for="newPassword2">New Password (again)</label>
37             <input class="form-control" formControlName="newPassword2" id="newPassword2"
38               placeholder="type again your new password"
39               type="password">
40           </div>
41
42           <button class="btn btn-primary" type="submit">Change Password</button>
43
44         </form>
45       </div>
46     </div>
47   </div>
48   <div class="col-4"></div>
49
```

File - C:\Users\giann\Documents\COD\sl-capstoneproject-foodbox\src\main\resources\frontend\foodbox-app\src\app\login\change.component.html
50 </div>

File - C:\Users\giann\Documents\COD\sl-capstoneproject-foodbox\src\main\resources\frontend\foodbox-app\src\app\login\logout.component.html

```
1 <div class="container">
2
3   <div class="cell row d-flex align-items-center justify-content-center" style="height: 60vh">
4
5     <div
6   ></div>
7   </div>
8 </div>
9
```

>Thank You for Your Business!</h1

File - C:\Users\giann\Documents\CODE\isl-capstoneproject-foodbox\src\main\resources\frontend\foodbox-app\src\app\login\logout.component.spec.ts

```
1 import { ComponentFixture, TestBed } from '@angular/core/testing';
2
3 import { LogoutComponent } from './logout.component';
4
5 describe('LogoutComponent', () => {
6   let component: LogoutComponent;
7   let fixture: ComponentFixture<LogoutComponent>;
8
9   beforeEach(async () => {
10     await TestBed.configureTestingModule({
11       declarations: [ LogoutComponent ]
12     })
13     .compileComponents();
14
15     fixture = TestBed.createComponent(LogoutComponent);
16     component = fixture.componentInstance;
17     fixture.detectChanges();
18   });
19
20   it('should create', () => {
21     expect(component).toBeTruthy();
22   });
23 });
24
```

File - C:\Users\giann\Documents\COD\isl-capstoneproject-foodbox\src\main\resources\frontend\foodbox-app\src\app\model\food.ts

```
1 export interface IFood {
2   id: number;
3   foodName: string;
4   price: number;
5   cuisine: string;
6   description: string;
7   isEnabled: string;
8   imageUrl: string;
9 }
10
11
12
```

File - C:\Users\giann\Documents\COD\sl-capstoneproject-foodbox\src\main\resources\frontend\foodbox-app\src\app\model\user.ts

```
1 export interface IUser {
2   id: number;
3   username: string;
4   name: string;
5   password: string;
6   contact: string;
7   credit: string;
8   profile: string;
9 }
```

File - C:\Users\giann\Documents\COD\sl-capstoneproject-foodbox\src\main\resources\frontend\foodbox-app\src\app\model\orden.ts

```
1 import {IFood} from "../food";
2 import {IUser} from "../user";
3
4 export interface IOrden {
5     user: IUser;
6     cart: IFood[];
7     status: string;
8 }
9
```

File - C:\Users\giann\Documents\COD\sl-capstoneproject-foodbox\src\main\resources\frontend\foodbox-app\src\app\model\displayOrder.ts

```
1 export interface IDisplayOrder{
2   id: number;
3   username: string;
4   foodName: string;
5   price: number;
6   status: string;
7 }
```


File - C:\Users\giann\Documents\CODE\sl-capstoneproject-foodbox\src\main\resources\frontend\foodbox-app\src\app\orden\orden.service.ts

```
1 import { Injectable } from '@angular/core';
2 import { HttpClient, HttpResponse, HttpHeaders } from '@angular/common/http';
3 import { catchError, tap } from 'rxjs/operators';
4 import { Observable, throwError } from 'rxjs';
5 import { IDisplayOrden } from '../model/displayOrden';
6
7
8 @Injectable({
9   providedIn: 'root'
10 })
11 export class OrdenService {
12   httpOptions = {
13     headers: new HttpHeaders({
14       'Accept': 'application/json',
15       'Content-Type': 'application/json',
16       'Access-Control-Allow-Origin': '*'
17     })
18   };
19
20   baseUrl: string = "http://foodboxapp-env.eba-5ye8smp8.us-west-2.elasticbeanstalk.com"
21   ordenUrl: string = this.baseUrl + '/cart/getDisplayOrden/';
22   constructor(private http: HttpClient) {
23
24   }
25
26
27   getDisplayOrden(ordenNum: number): Observable<IDisplayOrden[]> {
28     return this.http.get<IDisplayOrden[]>(<this.ordenUrl + ordenNum, this.httpOptions>).pipe(
29       tap(data => data),
30       catchError(this.handleError)
31     );
32   }
33
34   private handleError(err: HttpResponse) {
35     let errorMessage = '';
36     if (err.error instanceof ErrorEvent) {
37       errorMessage = `-----> An error occurred: ${err.error.message}`;
38     } else {
39       errorMessage = `-----> Server returned code: ${err.status}, error message is: ${err.message}`;
40     }
41     console.error(errorMessage);
42     return throwError(() => errorMessage);
43   }
44 }
45
46
```

File - C:\Users\giann\Documents\CODE\sl-capstoneproject-foodbox\src\main\resources\frontend\foodbox-app\src\app\orden\orden.component.ts

```
1 import {Component, OnInit} from '@angular/core';
2 import {OrdenService} from "../orden.service";
3 import {IDisplayOrden} from "../model/displayOrden";
4 import {Subscription} from "rxjs";
5 import {LoginService} from "../login/login.service";
6
7 @Component({
8   selector: 'ks-orden',
9   templateUrl: './orden.component.html',
10  styleUrls: ['./orden.component.css']
11 })
12 export class OrdenComponent implements OnInit {
13
14   displayOrdens: IDisplayOrden[] = [];
15   sub!: Subscription;
16
17   constructor(private loginService: LoginService,
18               private ordenService: OrdenService) {
19   }
20
21   ngOnInit(): void {
22
23     let ordenNum: number = this.loginService.id;
24
25     this.sub = this.ordenService.getDisplayOrden(ordenNum).subscribe({
26       next: data => {
27         this.displayOrdens = data
28       }
29     });
30
31     let titleUserName: string = "";
32     let titleId: number = 0;
33     let titleStatus: string = "";
34     this.displayOrdens.forEach(orden => {
35       if (orden.id === titleId) {
36         orden.id = 0;
37       } else {
38         titleId = orden.id;
39         titleUserName = "";
40         titleStatus = "";
41       }
42       if (orden.username === titleUserName) {
43         orden.username = "";
44       } else {
45         titleUserName = orden.username
46       }
47       if (orden.status === titleStatus) {
48         orden.status = "";
49       } else {
50         titleStatus = orden.status
51       }
52     });
53   }
```

File - C:\Users\giann\Documents\COD\sl-capstoneproject-foodbox\src\main\resources\frontend\foodbox-app\src\app\orden\orden.component.ts

```
50     }
51
52     })
53
54     return this.displayOrdens;
55
56     },
57     error: err => alert(err)
58   });
59 }
60
61 handleError(err: any): void {
62   alert(err.message)
63 }
64
65 private retClass: string = "";
66 private bGris: boolean = true;
67
68 setClass() {
69
70   this.retClass = this.bGris ? "pintaGris0" : "pintaGris1";
71   this.bGris = !this.bGris;
72   return this.retClass
73 }
74
75 getClass() {
76   return this.retClass;
77 }
78 }
79 }
80
```

File - C:\Users\giann\Documents\COD\isl-capstoneproject-foodbox\src\main\resources\frontend\foodbox-app\src\app\orden\orden.component.html

```
1 <div class="cell card">
2   <div class="cell card-header">
3     Thank you!
4   </div>
5   <div class="container">
6     <div class="cell card-body mt-3 mb-4">
7       <h5 class="cell card-title">This is your list of orders. We appreciate your patience when we prepare your box.</h5>
8     <p class="cell card-text">Your orders will show up here. </p>
9     <div class="row alert-info border rounded">
10
11
12     <div class="col-3 text-center fw-bold">Order Id:</div>
13     <div class="col-3 fw-bold">Username:</div>
14     <div class="col-3 fw-bold">Dish:</div>
15     <div class="col-3 fw-bold text-center">Status:</div>
16
17   </div>
18   <div *ngFor="let displayOrden of displayOrdens" class="cell">
19     <div class="row">
20       <div class="col-3"
21         [ngClass]="displayOrden.id === 0 ? getClass(): getClass()"
22         <div class="text-center">{{displayOrden.id === 0 ? "" : displayOrden.id}}</div>
23       </div>
24       <div class="col-3" [ngClass]="getClass()">{{displayOrden.username}}</div>
25       <div class="col-3" [ngClass]="getClass()">{{displayOrden.foodName}}</div>
26       <div class="col-3 text-center" [ngClass]="getClass()">{{displayOrden.status}}</div>
27     </div>
28   </div>
29 </div>
30 </div>
31 </div>
32
```

File - C:\Users\giann\Documents\COD\Isl-capstoneproject-foodbox\src\main\resources\frontend\foodbox-app\src\app\orden\orden.service.spec.ts

```
1 import { TestBed } from '@angular/core/testing';
2
3 import { OrdenService } from './orden.service';
4
5 describe('OrdenService', () => {
6   let service: OrdenService;
7
8   beforeEach(() => {
9     TestBed.configureTestingModule({});
10    service = TestBed.inject(OrdenService);
11  });
12
13  it('should be created', () => {
14    expect(service).toBeTruthy();
15  });
16 });
17
```

File - C:\Users\giann\Documents\COD\Isl-capstoneproject-foodbox\src\main\resources\frontend\foodbox-app\src\app\orden\orden.component.spec.ts

```
1 import { ComponentFixture, TestBed } from '@angular/core/testing';
2
3 import { OrdenComponent } from './orden.component';
4
5 describe('OrdenComponent', () => {
6   let component: OrdenComponent;
7   let fixture: ComponentFixture<OrdenComponent>;
8
9   beforeEach(async () => {
10     await TestBed.configureTestingModule({
11       declarations: [ OrdenComponent ]
12     })
13     .compileComponents();
14
15     fixture = TestBed.createComponent(OrdenComponent);
16     component = fixture.componentInstance;
17     fixture.detectChanges();
18   });
19
20   it('should create', () => {
21     expect(component).toBeTruthy();
22   });
23 });
24
```

File - C:\Users\giann\Documents\COD\El-capstoneproject-foodbox\src\main\resources\frontend\foodbox-app\src\app\tests\demo.component.spec.ts

```
1 import { ComponentFixture, TestBed } from '@angular/core/testing';
2
3 import { DemoComponent } from '../demo/demo.component';
4
5 describe('DemoComponent', () => {
6   let component: DemoComponent;
7   let fixture: ComponentFixture<DemoComponent>;
8
9   beforeEach(async () => {
10     await TestBed.configureTestingModule({
11       declarations: [ DemoComponent ]
12     })
13     .compileComponents();
14   });
15
16   beforeEach(() => {
17     fixture = TestBed.createComponent(DemoComponent);
18     component = fixture.componentInstance;
19     fixture.detectChanges();
20   });
21
22   it('should create', () => {
23     expect(component).toBeTruthy();
24   });
25 });
26
```

File - C:\Users\giann\Documents\COD\Isl-capstoneproject-foodbox\src\main\resources\frontend\foodbox-app\src\app\tests\account.service.spec.ts

```
1 import { TestBed } from '@angular/core/testing';
2
3 import { AccountService } from '../services/account.service';
4
5 describe('AccountService', () => {
6   let service: AccountService;
7
8   beforeEach(() => {
9     TestBed.configureTestingModule({});
10    service = TestBed.inject(AccountService);
11  });
12
13  it('should be created', () => {
14    expect(service).toBeTruthy();
15  });
16 });
17
```


File - C:\Users\giann\Documents\COD\Elis-capstoneproject-foodbox\src\main\resources\frontend\foodbox-app\src\app\tests\inventory.component.spec.ts

```
1 import { ComponentFixture, TestBed } from '@angular/core/testing';
2
3 import { InventoryComponent } from '../inventory/inventory.component';
4
5 describe('InventoryComponent', () => {
6   let component: InventoryComponent;
7   let fixture: ComponentFixture<InventoryComponent>;
8
9   beforeEach(async () => {
10     await TestBed.configureTestingModule({
11       declarations: [ InventoryComponent ]
12     })
13     .compileComponents();
14   });
15
16   beforeEach(() => {
17     fixture = TestBed.createComponent(InventoryComponent);
18     component = fixture.componentInstance;
19     fixture.detectChanges();
20   });
21
22   it('should create', () => {
23     expect(component).toBeTruthy();
24   });
25 });
26
```

File - C:\Users\giann\Documents\COD\Isl-capstoneproject-foodbox\src\main\resources\frontend\foodbox-app\src\app\tests\notification.service.spec.ts

```
1 import { TestBed } from '@angular/core/testing';
2
3 import { NotificationService } from '../notification/notification.service';
4
5 describe('NotificationService', () => {
6   let service: NotificationService;
7
8   beforeEach(() => {
9     TestBed.configureTestingModule({});
10    service = TestBed.inject(NotificationService);
11  });
12
13  it('should be created', () => {
14    expect(service).toBeTruthy();
15  });
16 });
17
```

File - C:\Users\giann\Documents\CODE\sl-capstoneproject\foodbox\src\main\resources\frontend\foodbox-app\src\app\search\new.service.ts

```
1 import {Injectable} from '@angular/core';
2 import {HttpClient, HttpResponse, HttpErrorResponse, HttpHeaders} from '@angular/common/http';
3 import {Observable, throwError} from 'rxjs';
4 import {IFood} from '../model/food';
5 import {catchError, tap} from 'rxjs/operators';
6
7 @Injectable({
8   providedIn: 'root'
9 })
10 export class NewService {
11   baseUrl: string = "http://foodboxapp-env.eba-5ye8smp8.us-west-2.elasticbeanstalk.com"
12   addFoodItemUrl: string = this.baseUrl + '/food/addFoodItem/';
13
14   httpOptions = {
15     headers: new HttpHeaders({
16       'Accept': 'application/json',
17       'Content-Type': 'application/json'
18     })
19   };
20   constructor(private http: HttpClient) {
21   }
22
23
24
25   addFoodItem(food: IFood): Observable<IFood> {
26
27
28     return this.http.post<IFood>(<this.addFoodItemUrl, food, this.httpOptions>).pipe(
29       tap(data => data),
30       catchError(this.handleError)
31     )
32   }
33
34
35   private handleError(err: HttpResponse) {
36     let errorMessage = '';
37     if (err.error instanceof ErrorEvent) {
38       errorMessage = `-----> An error occurred: ${err.error.message}`;
39     } else {
40       errorMessage = `-----> Server returned code: ${err.status}, error message is: ${err.message}`;
41     }
42     console.error(errorMessage);
43     return throwError(() => errorMessage);
44   }
45 }
46
```

File - C:\Users\giann\Documents\CODE\sl-capstoneproject-foodbox\src\main\resources\frontend\foodbox-app\src\app\search\edit.service.ts

```
1 import {Injectable} from '@angular/core';
2 import {HttpClient, HttpResponse, HttpErrorResponse, HttpHeaders} from '@angular/common/http';
3 import {Observable, throwError} from "rxjs";
4 import {IFood} from "../model/food";
5 import {catchError, tap} from "rxjs/operators";
6
7 @Injectable({
8   providedIn: 'root'
9 })
10 export class EditService {
11   baseUrl: string = "http://foodboxapp-env.eba-5ye8smp8.us-west-2.elasticbeanstalk.com"
12   getFoodItemUrl: string = this.baseUrl + '/food/getFoodItem/';
13   updateFoodItemUrl: string = this.baseUrl + '/food/updateFoodItem/';
14
15   httpOptions = {
16     headers: new HttpHeaders({
17       'Accept': 'application/json',
18       'Content-Type': 'application/json'
19     })
20   };
21   constructor(private http: HttpClient) {
22   }
23
24   getFoodItem(id: string | null): Observable<IFood> {
25     return this.http.get<IFood>(`${this.getFoodItemUrl} ${id}`);
26   }
27
28   updateFoodItem(food: IFood): Observable<IFood> {
29
30
31     return this.http.post<IFood>(`${this.updateFoodItemUrl}`, food, this.httpOptions).pipe(
32       tap(data => data),
33       catchError(this.handleError)
34     )
35   }
36
37   private handleError(err: HttpResponse) {
38     let errorMessage = '';
39     if (err.error instanceof ErrorEvent) {
40       errorMessage = `-----> An error occurred: ${err.error.message}`;
41     } else {
42       errorMessage = `-----> Server returned code: ${err.status}, error message is: ${err.message}`;
43     }
44     console.error(errorMessage);
45     return throwError(() => errorMessage);
46   }
47 }
48 }
49
```

File - C:\Users\giann\Documents\CODE\is-capstoneproject-foodbox\src\main\resources\frontend\foodbox-app\src\app\search\new.component.ts

```
1  import {Component, OnInit} from '@angular/core';
2  import {ActivatedRoute, Router} from "@angular/router";
3  import {FormBuilder, Validators} from "@angular/forms";
4  import {NewService} from "../new.service";
5  import {IFood} from "../model/food";
6
7  @Component({
8    selector: 'ks-new',
9    templateUrl: './new.component.html',
10   styleUrls: ['./new.component.css']
11 })
12 export class NewComponent implements OnInit {
13
14   private _myImageUrl: string = "http://tinyurl.com/dishBlankImage"
15
16   addFood = this.formBuilder.group({
17     id: [0, Validators.required],
18     description: ['', Validators.required],
19     foodName: ['', Validators.required],
20     price: [0, Validators.required],
21     imageUrl: ['', Validators.required],
22     cuisine: ['', Validators.required],
23     isEnabled: ['Y', Validators.required],
24   });
25
26
27   constructor(private FormBuilder: FormBuilder,
28               private activatedRoute: ActivatedRoute,
29               private newService: NewService,
30               private router: Router) {
31   }
32
33   get myImageUrl(): string {
34     return this._myImageUrl;
35   }
36
37   set myImageUrl(value: string) {
38     this._myImageUrl = value;
39   }
40
41
42   ngOnInit(): void {
43
44
45   }
46
47   onSubmit(): void {
48     const newFoodItem = <IFood><unknown>{
49
```

File - C:\Users\giann\Documents\COD\isl-capstoneproject-foodbox\src\main\resources\frontend\foodbox-app\src\app\search\new.component.ts

```
50     id: this.addFood.controls.id.value,
51     foodName: this.addFood.controls.foodName.value,
52     price: this.addFood.controls.price.value,
53     cuisine: this.addFood.controls.cuisine.value,
54     description: this.addFood.controls.description.value,
55     isEnabled: this.addFood.controls.isEnabled.value,
56     imageUrl: this.addFood.controls.imageUrl.value
57   }
58   this.newService.addFoodItem(newFoodItem).subscribe(data => {
59     alert('Dish was added successfully')
60     this.router.navigate(['//search'])
61   })
62 }
63 }
64
65
```

File - C:\Users\giann\Documents\CODE\sl-capstoneproject\foodbox\src\main\resources\frontend\foodbox-app\src\app\search\delete.service.ts

```
1 import {Injectable} from '@angular/core';
2 import {HttpClient, HttpResponse, HttpErrorResponse, HttpHeaders} from '@angular/common/http';
3 import {Observable, throwError} from "rxjs";
4 import {IFood} from "../model/food";
5 import {catchError, tap} from "rxjs/operators";
6
7 @Injectable({
8   providedIn: 'root'
9 })
10 export class DeleteService {
11   baseUrl: string = "http://foodboxapp-env.eba-5ye8smp8.us-west-2.elasticbeanstalk.com"
12   getFoodItemUrl: string = this.baseUrl + '/food/getFoodItem/';
13   deleteFoodItemUrl: string = this.baseUrl + '/food/deleteFoodItem/';
14
15   httpOptions = {
16     headers: new HttpHeaders({
17       'Accept': 'application/json',
18       'Content-Type': 'application/json',
19       'Access-Control-Allow-Origin': '*'
20     })
21   };
22   constructor(private http: HttpClient) {
23   }
24
25   getFoodItem(foodId: string | null): Observable<IFood> {
26     return this.http.get<IFood>(<this>.getFoodItemUrl + foodId);
27   }
28
29   deleteFoodItem(foodId: number | null): Observable<IFood> {
30
31
32     return <this>.http.delete<IFood>(<this>.deleteFoodItemUrl + foodId, <this>.httpOptions).pipe(
33       tap(data => data),
34       catchError(<this>.handleError)
35     )
36   }
37
38   private handleError(err: HttpResponse) {
39     let errorMessage = '';
40     if (err.error instanceof ErrorEvent) {
41       errorMessage = `-----> An error occurred: ${err.error.message}`;
42     } else {
43       errorMessage = `-----> Server returned code: ${err.status}, error message is: ${err.message}`;
44     }
45     console.error(errorMessage);
46     return throwError(() => errorMessage);
47   }
48 }
49 }
```

File - C:\Users\giann\Documents\COD\src\capstoneproject\foodbox\src\main\resources\frontend\foodbox-app\src\app\search\edit.component.ts

```
1 import {Component, OnInit} from '@angular/core';
2 import {ActivatedRoute, Router} from '@angular/router';
3 import {EditService} from './edit.service';
4 import {FormBuilder, Validators} from '@angular/forms';
5 import {IFood} from '../model/food';
6
7 @Component({
8   selector: 'ks-edit',
9   templateUrl: './edit.component.html',
10  styleUrls: ['./edit.component.css']
11 })
12 export class EditComponent implements OnInit {
13
14   private _myImageUrl: string = "http://tinyurl.com/dishBlankImage"
15
16   editFood = this.formBuilder.group({
17     id: [0, Validators.required],
18     description: ['', Validators.required],
19     foodName: ['', Validators.required],
20     price: [0, [Validators.required, Validators.min(0), Validators.max(100)]],
21     imageUrl: ['', Validators.required],
22     cuisine: ['', Validators.required],
23     isEnabled: ['Y', Validators.required],
24   });
25
26
27   constructor(private FormBuilder: FormBuilder,
28               private activatedRoute: ActivatedRoute,
29               private editService: EditService,
30               private router: Router) {
31
32   }
33   get myImageUrl(): string {
34     return this._myImageUrl;
35   }
36
37   set myImageUrl(value: string) {
38     this._myImageUrl = value;
39   }
40
41
42   ngOnInit(): void {
43     this.activatedRoute.paramMap.subscribe(params => {
44       this.editService.getItem(params.get("id")).subscribe(data => {
45
46         this.editFood.setValue({
47           id: data.id,
48           description: data.description,
49
```


File - C:\Users\giann\Documents\CODE\sl-capstoneproject-foodbox\src\main\resources\frontend\foodbox-app\src\app\search\edit.component.ts

```
50     foodName: data.foodName,
51     price: data.price,
52     imageUrl: data.imageUrl,
53     cuisine: data.cuisine,
54     isEnabled: data.isEnabled
55   });
56   })
57   })
58   }
59   onSubmit(): void {
60     if (!this.editFood.invalid){
61       const editedFood = <IFood><unknown>{
62         id: this.editFood.controls.id.value,
63         foodName: this.editFood.controls.foodName.value,
64         price: this.editFood.controls.price.value,
65         cuisine: this.editFood.controls.cuisine.value,
66         description: this.editFood.controls.description.value,
67         isEnabled: this.editFood.controls.isEnabled.value,
68         imageUrl: this.editFood.controls.imageUrl.value
69       }
70       this.editService.updateFoodItem(editedFood).subscribe(data => {
71         alert('Dish was updated successfully')
72         this.router.navigate(['search'])
73       })
74     }
75   }
76   }
77 }
78 }
79 }
```

File - C:\Users\giann\Documents\CODE\sl-capstoneproject-foodbox\src\main\resources\frontend\foodbox-app\src\app\search\search.service.ts

```
1 import {Injectable} from '@angular/core';
2 import {IFood} from "../model/food";
3 import {HttpClient, HttpErrorResponse} from "@angular/common/http";
4 import {Observable, throwError} from "rxjs";
5 import {catchError, tap} from "rxjs/operators";
6 import {IOrden} from "../model/orden";
7 import {NotificationType} from "../notification/notification.message";
8 import {NotificationService} from "../notification/notification.service";
9
10
11 @Injectable({
12   providedIn: 'root'
13 })
14 export class SearchService {
15
16   baseUrl: string = "http://foodboxapp-env.eba-5ye8smp8.us-west-2.elasticbeanstalk.com"
17   foodUrl: string = this.baseUrl + '/food/getAllFoods';
18   postPurchaseUrl: string = '';
19   cart: IFood[] = [];
20   orden!: IOrden;
21   absoluteTotal: number = 0;
22
23   constructor(private http: HttpClient, private notificationService: NotificationService) {
24   }
25
26   getFoods(): Observable<IFood[]> {
27     return this.http.get<IFood[]>(this.foodUrl).pipe(
28       tap(data => data),
29       catchError(this.handleError)
30     );
31   }
32
33   postPurchase(orden: any): Observable<any> {
34     return this.http.post<any>(this.postPurchaseUrl, orden).pipe(
35       tap(data => data),
36       catchError(this.handleError)
37     );
38   }
39
40   handleError(error: HttpErrorResponse) {
41     let errorMessage
42     if (error.error instanceof ErrorEvent) {
43       errorMessage = `An error occurred: ${error.error.message}`;
44     } else {
45       errorMessage = `Server returned code: ${error.status}, error message is: ${error.message}`;
46     }
47     console.error(errorMessage);
48     return throwError(errorMessage);
49   }
```

File - C:\Users\giann\Documents\COD\tsl-capstoneproject-foodbox\src\main\resources\frontend\foodbox-app\src\app\search\search.service.ts

```
50
51   addToCart(food: IFood) {
52     this.cart.push(food);
53     this.notificationService.sendMessage({text: food.foodName + " added to cart", type: NotificationType.success});
54
55   }
56
57   getCart(): IFood[] {
58     return this.cart;
59   }
60
61   removeItem(food: IFood) {
62
63     let foodIdToDelete = (element: { id: number; }) => food.id === element.id;
64
65     let index = this.cart.findIndex(foodIdToDelete)
66     this.cart.splice(index, 1)
67     this.notificationService.sendMessage({
68       text: food.foodName + " removed from cart",
69       type: NotificationType.error
70     });
71   }
72
73   getTotal(): number {
74     this.absoluteTotal = 0;
75     this.cart.forEach(item => this.absoluteTotal += item.price)
76     return this.absoluteTotal;
77   }
78 }
```

File - C:\Users\giann\Documents\COD\is1-capstoneproject-foodbox\src\main\resources\frontend\foodbox-app\src\app\search\new.component.html

```
1 <div class="row">
2   <div class="col-1">
3
4   </div>
5   <div class="col-8">
6     <div class="card shadow-1">
7       <div class="card-header">
8         Add a New Dish
9       </div>
10      <div class="card-body">
11        <form (ngSubmit)="onSubmit()" [formGroup]="addFood" class="gapTop1em">
12          <h5 class="card-title gapTop1em">On this screen, you add a new Dish</h5>
13          <div class="cell row">
14            <div class="cell col-7 ">
15
16            <div class="cell form-group row">
17              <label class="col-3 col-form-label text-end" for="foodName">ID: </label>
18              <div class="col-9">
19                <input [readOnly]="true" type="number" class="form-control" formControlName="id" id="id" value="" />
20              </div>
21            </div>
22
23            <div class="cell form-group row">
24              <label class="col-3 col-form-label text-end" for="foodName">Dish Name: </label>
25              <div class="col-9">
26                <input type="text" class="form-control" formControlName="foodName" id="foodName" value="" />
27                <div *ngIf="addFood.controls['foodName'].invalid && addFood.controls['foodName'].touched">
28                  <div *ngIf="addFood.controls['foodName'].errors!.required"
29                    class="badge shadow-0 bg-danger">
30                    Dish Name is required.
31                  </div>
32                </div>
33              </div>
34            </div>
35
36            <div class="cell form-group row">
37              <label class="col-3 col-form-label text-end" for="cuisine">Cuisine: </label>
38              <div class="col-9">
39                <select class="form-select" formControlName="cuisine" id="cuisine">
40                  <option>MEXICAN</option>
41                  <option>ITALIAN</option>
42                  <option>CARIBBEAN</option>
43                  <option>ASIAN</option>
44                  <option>AFRICAN</option>
45                </select>
46                <div *ngIf="addFood.controls['cuisine'].invalid && addFood.controls['cuisine'].touched">
47                  <div *ngIf="addFood.controls['cuisine'].errors!.required"
48                    class="badge shadow-0 bg-danger">
49                    Cuisine is required.
```

File - C:\Users\giann\Documents\COD\sl-capstoneproject-foodbox\src\main\resources\frontend\foodbox-app\src\app\search\new.component.html

```
50     </div>
51   </div>
52 </div>
53 </div>
54
55 <div class="cell form-group row">
56   <label class="col-3 col-form-label text-end" for="description">Description: </label>
57   <div class="col-9">
58     <textarea class="form-control" formControlName="description" id="description" rows="3"></textarea>
59     <div *ngIf="addFood.controls['description'].invalid && addFood.controls['description'].touched">
60       <div *ngIf="addFood.controls['description'].errors!.required"
61         class="badge shadow-0 bg-danger">
62         Holy guacamole! Description is required.
63       </div>
64     </div>
65   </div>
66 </div>
67
68 <div class="cell form-group row">
69   <label class="col-3 col-form-label text-end">is Enabled: </label>
70   <div class="col-9">
71
72
73
74
75     <div class="form-check">
76       <label class="form-check-label" for="Y"> Yes </label>
77       <input class="form-check-input" type="radio" id="Y" value="Y" name="isEnabled" formControlName="
78         isEnabled"
79         checked>
80     </div>
81
82
83
84
85
86
87
88 <div class="cell form-group row">
89   <label class="col-3 col-form-label text-end" for="price">Price: </label>
90   <div class="col-9">
91     <input type="number" class="form-control" formControlName="price" id="price" value="" />
92     <div *ngIf="addFood.controls['price'].invalid && addFood.controls['price'].touched">
93       <div *ngIf="addFood.controls['price'].errors!.required" class="badge shadow-0 bg-danger">
94         ('max ')" class="badge shadow-0 bg-danger">-->
95       </div>
96     </div>
97   </div>
98 </div>
99 </div>
```

```

96     </div>
97   </div>
98   </div>
99   </div>
100   <div class="cell form-group row">
101     <label class="col-3 col-form-label text-end" for="imageUrl">Image:</label>
102     <div class="col-9">
103       <input type="text" class="form-control" formControlName="imageUrl" id="imageUrl" value=""
104         [(ngModel)]="myImageUrl"/>
105       <div *ngIf="addFood.controls['imageUrl'].invalid && addFood.controls['imageUrl'].touched">
106         <div *ngIf="addFood.controls['imageUrl'].errors!.required"
107           class="badge shadow-0 bg-danger">
108           Image is required.
109         </div>
110       </div>
111     </div>
112   </div>
113   </div>
114   <div class="cell col-5 d-flex align-items-center gapBottom2em">
115
116
117   <div class="cell-blue col-12">
118     <div class="cell row">
119       <div class="cell col-1"></div>
120       <div class="cell col-10">
121         
122       </div>
123       <div class="cell col-1"></div>
124     </div>
125     <div class="cell row">
126       <div class="cell d-flex justify-content-center">
127         <button type="submit" class="gapTop2em btn btn-primary">
128           Add New Dish
129         </button>
130       </div>
131     </div>
132   </div>
133
134
135   </div>
136
137   </div>
138   </form>
139 </div>
140 </div>
141 <div class="col-3"></div>
142 <div class="col-3"></div>
143 </div>
144

```

File - C:\Users\giann\Documents\COD\isl-capstoneproject-foodbox\src\main\resources\frontend\foodbox-app\src\app\search\delete.component.ts

```
1 import {Component, OnInit} from '@angular/core';
2 import {ActivatedRoute, Router} from '@angular/router';
3 import {DeleteService} from '../delete.service';
4 import {FormBuilder, Validators} from '@angular/forms';
5 import {IFood} from '../model/food';
6
7 @Component({
8   selector: 'ks-edit',
9   templateUrl: './delete.component.html',
10  styleUrls: ['./delete.component.css']
11 })
12 export class DeleteComponent implements OnInit {
13
14   private _myImageUrl: string = "http://tinyurl.com/dishBlankImage"
15
16   deleteFood = this.formBuilder.group({
17     id: [0, Validators.required],
18     description: ['', Validators.required],
19     foodName: ['', Validators.required],
20     price: [0, Validators.required],
21     imageUrl: ['', Validators.required],
22     cuisine: ['', Validators.required],
23     isEnabled: ['Y', Validators.required],
24   });
25
26
27   constructor(private formBuilder: FormBuilder,
28               private activatedRoute: ActivatedRoute,
29               private deleteService: DeleteService,
30               private router: Router) {
31
32
33
34   get myImageUrl(): string {
35     return this._myImageUrl;
36   }
37
38   set myImageUrl(value: string) {
39     this._myImageUrl = value;
40
41   }
42   ngOnInit(): void {
43     this.activatedRoute.paramMap.subscribe(params => {
44       this.deleteService.getItem(params.get("id")).subscribe(data => {
45
46         this.deleteFood.setValue({
47           id: data.id,
48           description: data.description,
49           foodName: data.foodName,
```

File - C:\Users\giann\Documents\COD\isl-capstoneproject-foodbox\src\main\resources\frontend\foodbox-app\src\app\search\delete.component.ts

```
50     price: data.price,
51     imageUrl: data.imageUrl,
52     cuisine: data.cuisine,
53     isEnabled: data.isEnabled
54   });
55   })
56   })
57 }
58
59 ;
60 onSubmit(): void {
61
62   this.deleteService.deleteFoodItem(Number(this.deleteFood.controls.id.value)).subscribe(data => {
63     alert('Dish was deleted successfully')
64     this.router.navigate(['/search'])
65   })
66 }
67 }
68 }
69
```


File - C:\Users\giann\Documents\CODE\is-capstoneproject-foodbox\src\main\resources\frontend\foodbox-app\src\app\search\edit.component.html

```
1 <div class="row">
2   <div class="col-1"></div>
3   <div class="col-8">
4     <div class="card shadow-1">
5       <div class="card-header">
6         Edit Dish
7       </div>
8       <div class="card-body">
9         <form (ngSubmit)="onSubmit()" [formGroup]="editFood" class="gapTop1em">
10           <h5 class="card-title gapTop1em">On this screen, you edit the details of the a Dish</h5>
11           <div class="cell row">
12             <div class="cell col-7 ">
13
14             <div class="cell form-group row">
15               <label class="col-3 col-form-label text-end" for="foodName">ID: </label>
16               <div class="col-9">
17                 <input [readOnly]="true" type="number" class="form-control" formControlName="id" id="id" value="" />
18               </div>
19             </div>
20
21             <div class="cell form-group row">
22               <label class="col-3 col-form-label text-end" for="foodName">Dish Name: </label>
23               <div class="col-9">
24                 <input type="text" class="form-control" formControlName="foodName" id="foodName" value="" />
25                 <div *ngIf="editFood.controls['foodName'].invalid && editFood.controls['foodName'].touched">
26                   <div *ngIf="editFood.controls['foodName'].errors!.required"
27                     class="badge shadow-0 bg-danger">
28                     Dish Name is required.
29                   </div>
30                 </div>
31               </div>
32             </div>
33
34             <div class="cell form-group row">
35               <label class="col-3 col-form-label text-end" for="cuisine">Cuisine: </label>
36               <div class="col-9">
37                 <select class="form-select" formControlName="cuisine" id="cuisine">
38                   <option>ARAB</option>
39                   <option>MEXICAN</option>
40                   <option>ITALIAN</option>
41                   <option>FRENCH</option>
42                   <option>SPAIN</option>
43                 </select>
44                 <div *ngIf="editFood.controls['cuisine'].invalid && editFood.controls['cuisine'].touched">
45                   <div *ngIf="editFood.controls['cuisine'].errors!.required"
46                     class="badge shadow-0 bg-danger">
47                     Cuisine is required.
48                   </div>
49                 </div>

```

File - C:\Users\giann\Documents\COD\sl-capstoneproject-foodbox\src\main\resources\frontend\foodbox-app\src\app\search\edit.component.html

```
50     </div>
51   </div>
52
53   <div class="cell form-group row">
54     <label class="col-3 col-form-label text-end" for="description">Description: </label>
55     <div class="col-9">
56       <textarea class="form-control" formControlName="description" id="description" rows="3"></textarea>
57       <div *ngIf="editFood.controls['description'].invalid && editFood.controls['description'].touched">
58         <div *ngIf="editFood.controls['description'].errors!.required"
59           class="badge shadow-0 bg-danger">
60           Holy guacamole! Description is required.
61         </div>
62       </div>
63     </div>
64   </div>
65
66   <div class="cell form-group row">
67     <label class="col-3 col-form-label text-end">is Enabled: </label>
68     <div class="col-9">
69
70
71     <div class="form-check">
72       <label class="form-check-label" for="Y"> Yes </label>
73       <input class="form-check-input" type="radio" id="Y" value="Y" name="isEnabled" formControlName="
isEnabled"
74       checked>
75     </div>
76     <div class="form-check">
77       <label class="form-check-label" for="N"> No </label>
78       <input class="form-check-input" type="radio" id="N" value="N" name="isEnabled" formControlName="
isEnabled"
79     >
80   </div>
81
82   </div>
83 </div>
84
85   <div class="cell form-group row">
86     <label class="col-3 col-form-label text-end" for="price">Price: </label>
87     <div class="col-9">
88       <input type="number" class="form-control" formControlName="price" id="price" value="" />
89       <div *ngIf="editFood.controls['price'].invalid && editFood.controls['price'].touched">
90         <div *ngIf="editFood.controls['price'].errors!.required" class="badge shadow-0 bg-danger">
91           <div *ngIf="editFood.controls['price'].hasError('max')" class="badge shadow-0 bg-danger">-->
92           Price is required.
93         </div>
94       </div>
95     </div>
96   </div>
```

```

97 </div>
98 <div class="cell form-group row">
99 <label class="col-3 col-form-label text-end" for="imageUrl">Image:</label>
100 <div class="col-9">
101 <input type="text" class="form-control" formControlName="imageUrl" id="imageUrl" value=""
102 [(ngModel)]="myImageUrl"/>
103 <div *ngIf="editFood.controls['imageUrl'].invalid && editFood.controls['imageUrl'].touched">
104 <div *ngIf="editFood.controls['imageUrl'].errors!.required"
105 class="badge shadow-0 bg-danger">
106 Image is required.
107 </div>
108 </div>
109 </div>
110 </div>
111 </div>
112 <div class="cell col-5 d-flex align-items-center gapBottom2em">
113
114
115 <div class="cell-blue col-12">
116 <div class="cell row">
117 <div class="cell col-1"></div>
118 <div class="cell col-10">
119 
120 </div>
121 <div class="cell col-1"></div>
122 </div>
123 <div class="cell row">
124 <div class="cell d-flex justify-content-center">
125 <button type="submit" class="gapTop2em btn btn-primary">
126 Update Dish
127 </button>
128 </div>
129 </div>
130 </div>
131
132
133 </div>
134
135 </div>
136 </form>
137 </div>
138 </div>
139 </div>
140 <div class="col-3"></div>
141 </div>
142

```

File - C:\Users\giann\Documents\COD\Isl-capstoneproject-foodbox\src\main\resources\frontend\foodbox-app\src\app\search\new.service.spec.ts

```
1 import { TestBed } from '@angular/core/testing';
2
3 import { NewService } from './new.service';
4
5 describe('NewService', () => {
6   let service: NewService;
7
8   beforeEach(() => {
9     TestBed.configureTestingModule({});
10    service = TestBed.inject(NewService);
11  });
12
13  it('should be created', () => {
14    expect(service).toBeTruthy();
15  });
16 });
17
```

File - C:\Users\giann\Documents\CODE\is-capstoneproject-foodbox\src\main\resources\frontend\foodbox-app\src\app\search\search.component.ts

```
1 import {Component, OnDestroy, OnInit} from '@angular/core';
2 import {IFood} from "../model/food";
3 import {SearchService} from "../search.service";
4 import {Subscription} from "rxjs";
5 import {LoginService} from "../login/login.service";
6 import {Router} from "@angular/router";
7
8 @Component({
9   templateUrl: './search.component.html',
10  styleUrls: ['./search.component.css']
11 })
12 export class SearchComponent implements OnInit, OnDestroy {
13
14   foodsFiltered: IFood[] = [];
15   foodsOriginal: IFood[] = [];
16   foods: IFood[] = [];
17   cart: IFood[] = [];
18   listFilter: string = '';
19   _dishFilter: string = '';
20   _cuisineFilter: string = '';
21   _descriptionFilter: string = '';
22   _priceFilter!: number;
23   sub!: Subscription;
24
25   constructor(private searchService: SearchService,
26               private loginService: LoginService,
27               private router: Router) {
28
29   }
30
31   getLoginService() {
32     return this.loginService;
33   }
34
35   ngOnInit(): void {
36     this.sub = this.searchService.getFoods().subscribe({
37       next: data => {
38         this.foods = data;
39         this.foodsFiltered = this.foods;
40         this.foodsOriginal = this.foods;
41         return this.foodsFiltered;
42       },
43       error: err => alert(err)
44     });
45
46   }
47
48 }
49
```

File - C:\Users\giann\Documents\COD\Isl-capstoneproject-foodbox\src\main\resources\frontend\foodbox-app\src\app\search\search.component.ts

```
50 get dishFilter(): string {
51
52     return this._dishFilter;
53 }
54
55 set dishFilter(value: string) {
56     this._dishFilter = value;
57     this.foodsFiltered = this.foodsOriginal.filter(
58         food => food.foodName.toLowerCase().includes(value.toLowerCase())
59         && food.cuisine.toLowerCase().includes(this._cuisineFilter)
60         && food.description.toLowerCase().includes(this._descriptionFilter)
61         && (this._priceFilter === null || this._priceFilter === undefined ? food.price > 0 : food.price <= this._priceFilter)
62     )
63 }
64
65 get cuisineFilter(): string {
66
67     return this._cuisineFilter;
68 }
69
70 set cuisineFilter(value: string) {
71     this._cuisineFilter = value;
72     this.foodsFiltered = this.foodsOriginal.filter(
73         food => food.cuisine.toLowerCase().includes(value.toLowerCase())
74         && food.foodName.toLowerCase().includes(this._dishFilter)
75         && food.description.toLowerCase().includes(this._descriptionFilter)
76         && (this._priceFilter === null || this._priceFilter === undefined ? food.price > 0 : food.price <= this._priceFilter)
77     )
78 }
79
80 get descriptionFilter(): string {
81
82     return this._descriptionFilter;
83 }
84
85 set descriptionFilter(value: string) {
86     this._descriptionFilter = value;
87     this.foodsFiltered = this.foodsOriginal.filter(
88         food => food.description.toLowerCase().includes(value.toLowerCase())
89         && food.foodName.toLowerCase().includes(this._dishFilter)
90         && food.cuisine.toLowerCase().includes(this._cuisineFilter)
91         && (this._priceFilter === null || this._priceFilter === undefined ? food.price > 0 : food.price <= this._priceFilter)
92     )
93 }
94
95 get priceFilter(): number {
96
97     return this._priceFilter;
98 }
```

```
99
100 set priceFilter(value: number) {
101
102     this._priceFilter = value;
103     this.foodsFiltered = this.foodsOriginal.filter(
104         food => (value === null ? food.price > 0 : food.price <= value)
105             && food.foodName.toLowerCase().includes(this._dishFilter)
106             && food.cuisine.toLowerCase().includes(this._cuisineFilter)
107             && food.description.toLowerCase().includes(this._descriptionFilter)
108     )
109 }
110
111 ngOnDestroy(): void {
112     //this.sub.unsubscribe();
113     //food => food.foodName.toLowerCase()
114     // .includes(this.listFilter.toLowerCase())
115 }
116
117 select(food: IFood) {
118     this.searchService.addToCart(food);
119 }
120
121 edit(food: IFood): void {
122     this.router.navigate(['/editFood', food.id])
123 }
124
125
126
127 RandomTilt() {
128     let tilt: number = 0;
129     tilt = Math.floor(Math.random() * 5)+1;
130     return "deg-" + tilt
131 }
132
133 delete(food: IFood) {
134     this.router.navigate(['/deleteFood', food.id])
135 }
136
137
138 newDish() {
139     this.router.navigate(['/newFood'])
140 }
141
142 }
143
```

File - C:\Users\giann\Documents\COD\isl-capstoneproject-foodbox\src\main\resources\frontend\foodbox-app\src\app\search\edit.service.spec.ts

```
1 import { TestBed } from '@angular/core/testing';
2
3 import { EditService } from './edit.service';
4
5 describe('EditService', () => {
6   let service: EditService;
7
8   beforeEach(() => {
9     TestBed.configureTestingModule({});
10    service = TestBed.inject(EditService);
11  });
12
13  it('should be created', () => {
14    expect(service).toBeTruthy();
15  });
16 });
17
```


File - C:\Users\giann\Documents\COD\Isl-capstoneproject-foodbox\src\main\resources\frontend\foodbox-app\src\app\search\delete.component.html

```
1 <div class="row">
2   <div class="col-1"></div>
3   <div class="col-8">
4     <div class="card shadow-1">
5       <div class="card-header">
6         Delete Dish
7       </div>
8       <div class="card-body">
9         <form (ngSubmit)="onSubmit()" [formGroup]="deleteFood" class="gapTop1em">
10          <h5 class="card-title gapTop1em">On this screen, you delete a Dish.</h5>
11          <div class="cell row">
12            <div class="cell col-7 ">
13
14            <div class="cell form-group row">
15              <label class="col-3 col-form-label text-end" for="foodName">ID: </label>
16              <div class="col-9">
17                <input [readOnly]="true" type="number" class="form-control" formControlName="id" id="id" value="" />
18              </div>
19            </div>
20
21            <div class="cell form-group row">
22              <label class="col-3 col-form-label text-end" for="foodName">Dish Name: </label>
23              <div class="col-9">
24                <input [readOnly]="true" type="text" class="form-control" formControlName="foodName" id="foodName"
25                  value="" />
26              </div>
27            <div *ngIf="deleteFood.controls['foodName'].invalid && deleteFood.controls['foodName'].errors!.required"
28              class="badge shadow-0 bg-danger">
29              Dish Name is required.
30            </div>
31          </div>
32        </div>
33
34        <div class="cell form-group row">
35          <label class="col-3 col-form-label text-end" for="cuisine">Cuisine: </label>
36          <div class="col-9">
37            <input [readOnly]="true" type="text" class="form-control" formControlName="cuisine" id="cuisine"
38              value="" />
39          </div>
40          <div *ngIf="deleteFood.controls['cuisine'].invalid && deleteFood.controls['cuisine'].errors!.required"
41            class="badge shadow-0 bg-danger">
42            Cuisine is required.
43          </div>
44        </div>
45      </div>
46    </div>
47  </div>
```

File - C:\Users\giann\Documents\COD\sl-capstoneproject-foodbox\src\main\resources\frontend\foodbox-app\src\app\search\delete.component.html

```
48 <div class="cell form-group row">
49   <label class="col-3 col-form-label text-end" for="description">Description: </label>
50   <div class="col-9">
51     <textarea [readOnly]="true" class="form-control" formControlName="description" id="description"
rows="3"></textarea>
52   <div *ngIf="deleteFood.controls['description'].invalid && deleteFood.controls['description'].touched">
53     <div *ngIf="deleteFood.controls['description'].errors!.required"
54       class="badge shadow-0 bg-danger">
55       Holy guacamole! Description is required.
56     </div>
57   </div>
58 </div>
59 </div>
60
61 <div class="cell form-group row">
62   <label class="col-3 col-form-label text-end">is Enabled: </label>
63   <div class="col-9">
64
65     <input [readOnly]="true" type="text" class="form-control" formControlName="isEnabled" id="isEnabled"
value="" />
66
67
68
69
70
71
72 <div class="cell form-group row">
73   <label class="col-3 col-form-label text-end" for="price">Price: </label>
74   <div class="col-9">
75     <input [readOnly]="true" type="number" class="form-control" formControlName="price" id="price" value
=""/>
76
77   <div *ngIf="deleteFood.controls['price'].invalid && deleteFood.controls['price'].touched">
78     <div *ngIf="deleteFood.controls['price'].errors!.required" class="badge shadow-0 bg-danger">
79       <!--
80       ('max ') " class="badge shadow-0 bg-danger">-->
81       Price is required.
82       <!--
83     </div>
84   </div>
85 </div>
86 <div class="cell form-group row">
87   <label class="col-3 col-form-label text-end" for="imageUrl">Image:</label>
88   <div class="col-9">
89     <input [readOnly]="true" type="text" class="form-control" formControlName="imageUrl" id="imageUrl"
value=""
90     [(ngModel)]="myImageUrl" />
91   <div *ngIf="deleteFood.controls['imageUrl'].invalid && deleteFood.controls['imageUrl'].touched">
92     <div *ngIf="deleteFood.controls['imageUrl'].errors!.required"
```

```
92     class="badge shadow-0 bg-danger">
93         Image is required.
94     </div>
95 </div>
96 </div>
97 </div>
98 <div class="cell col-5 d-flex align-items-center gapBottom2em">
99
100
101
102     <div class="cell-blue col-12">
103         <div class="cell row">
104             <div class="cell col-1"></div>
105             <div class="cell col-10">
106                 
107             </div>
108             <div class="cell col-1"></div>
109         </div>
110         <div class="cell row">
111             <div class="cell d-flex justify-content-center">
112                 <button type="submit" class="gapTop2em btn btn-primary">
113                     Delete Dish
114                 </button>
115             </div>
116         </div>
117     </div>
118
119
120 </div>
121
122 </div>
123 </form>
124 </div>
125 </div>
126 </div>
127 <div class="col-3"></div>
128 </div>
129
```

File - C:\Users\giann\Documents\COD\Elis-capstoneproject-foodbox\src\main\resources\frontend\foodbox-app\src\app\search\new.component.spec.ts

```
1 import { ComponentFixture, TestBed } from '@angular/core/testing';
2
3 import { NewComponent } from './new.component';
4
5 describe('NewComponent', () => {
6   let component: NewComponent;
7   let fixture: ComponentFixture<NewComponent>;
8
9   beforeEach(async () => {
10     await TestBed.configureTestingModule({
11       declarations: [ NewComponent ]
12     })
13     .compileComponents();
14
15     fixture = TestBed.createComponent(NewComponent);
16     component = fixture.componentInstance;
17     fixture.detectChanges();
18   });
19
20   it('should create', () => {
21     expect(component).toBeTruthy();
22   });
23 });
24
```

File - C:\Users\giann\Documents\CODE\sl-capstoneproject-foodbox\src\main\resources\frontend\foodbox-app\src\app\search\search.component.html

[illegible]

File - C:\Users\giann\Documents\COD\is1-capstoneproject-foodbox\src\main\resources\frontend\foodbox-app\src\app\search\search.component.html

```
50 <div class="cell col-1" style="display:inline-block">
51   <input [(ngModel)]="priceFilter" class="cell form-control" placeholder="Price"
52     type="number">
53 </div>
54 <div class="cell col-1 text-center">&nbsp;</div>
55 </div>
56 <div *ngFor="let food of foodsFiltered" class="cell row border rounded p-1">
57   <div class="cell col-1 d-flex align-items-center">
58     <button *ngIf="!getLoginService().admin" (click)="select(food)"
59       class="cell btn btn-outline-primary btn-sm alert-secondary ">
60       select
61     </button>
62     <button *ngIf="getLoginService().admin" (click)="edit(food)"
63       class="cell btn btn-outline-primary btn-sm alert-secondary adminBorder">
64       <i aria-hidden="true" class='fa fa-pencil'></i> edit
65     </button>
66   </div>
67   <span hidden>{{food.id}}</span>
68   <div class="cell col-2 d-flex align-items-center">{{food.foodName}}</div>
69   <div class="cell col-2 d-flex align-items-center">
70     
71   </div>
72   <div class="cell col-1 d-flex align-items-center"><div class="cell text-centered">{{food.cuisine}}</div></div>
73   <div class="cell col-4 d-flex align-items-center">{{food.description}}</div>
74   <div class="cell col-1 d-flex align-items-center"><div class="cell text-centered">{{food.price|currency}}</div></div>
75 >
76
77
78   <button *ngIf="getLoginService().admin" (click)="delete(food)"
79     class="cell btn btn-outline-primary btn-sm alert-secondary adminBorder">
80     <i aria-hidden="true" class='fa fa-eraser'></i> delete
81   </button>
82 </div>
83 </div>
84 </div>
85 </div>
86 </div>
```

File - C:\Users\giann\Documents\COD\IsI-capstoneproject-foodbox\src\main\resources\frontend\foodbox-app\src\app\search\delete.service.spec.ts

```
1 import { TestBed } from '@angular/core/testing';
2
3 import { DeleteService } from './delete.service';
4
5 describe('DeleteService', () => {
6   let service: DeleteService;
7
8   beforeEach(() => {
9     TestBed.configureTestingModule({});
10    service = TestBed.inject(DeleteService);
11  });
12
13  it('should be created', () => {
14    expect(service).toBeTruthy();
15  });
16 });
17
```

File - C:\Users\giann\Documents\COD\Isl-capstoneproject-foodbox\src\main\resources\frontend\foodbox-app\src\app\search\edit.component.spec.ts

```
1 import { ComponentFixture, TestBed } from '@angular/core/testing';
2
3 import { EditComponent } from './edit.component';
4
5 describe('EditComponent', () => {
6   let component: EditComponent;
7   let fixture: ComponentFixture<EditComponent>;
8
9   beforeEach(async () => {
10     await TestBed.configureTestingModule({
11       declarations: [ EditComponent ]
12     })
13     .compileComponents();
14
15     fixture = TestBed.createComponent(EditComponent);
16     component = fixture.componentInstance;
17     fixture.detectChanges();
18   });
19
20   it('should create', () => {
21     expect(component).toBeTruthy();
22   });
23 });
24
```


File - C:\Users\giann\Documents\COD\sl-capstoneproject-foodbox\src\main\resources\frontend\foodbox-app\src\app\search\delete.component.spec.ts

```
1 import { ComponentFixture, TestBed } from '@angular/core/testing';
2
3 import { DeleteComponent } from './delete.component';
4
5 describe('DeleteComponent', () => {
6   let component: DeleteComponent;
7   let fixture: ComponentFixture<DeleteComponent>;
8
9   beforeEach(async () => {
10     await TestBed.configureTestingModule({
11       declarations: [ DeleteComponent ]
12     })
13     .compileComponents();
14
15     fixture = TestBed.createComponent(DeleteComponent);
16     component = fixture.componentInstance;
17     fixture.detectChanges();
18   });
19
20   it('should create', () => {
21     expect(component).toBeTruthy();
22   });
23 });
24
```

File - C:\Users\giann\Documents\CODE\isl-capstoneproject-foodbox\src\main\resources\frontend\foodbox-app\src\app\checkout\checkout.component.ts

```
1 import {Component, OnInit} from '@angular/core';
2 import {PurchaseService} from "../purchase/purchase.service";
3 import {SearchService} from "../search/search.service";
4 import {IFood} from "../model/food";
5 import {UntypedFormBuilder, Validators} from "@angular/forms";
6 import {Router} from "@angular/router";
7 import {IOrden} from "../model/orden";
8 import {LoginService} from "../login/login.service";
9
10 @Component({
11   selector: 'ks-checkout',
12   templateUrl: './checkout.component.html',
13   styleUrls: ['./checkout.component.css']
14 })
15
16 export class CheckoutComponent implements OnInit {
17   cart: IFood[] = []
18   cartTotal = this.searchService.absoluteTotal
19   checkoutForm = this.formBuilder.group({
20     id: [this.loginService.id, Validators.required],
21     name: [this.loginService.name, Validators.required],
22     contact: [this.loginService.contact, Validators.required],
23     credit: [this.loginService.credit, Validators.required]
24   });
25
26   constructor(private loginService: LoginService,
27     private searchService: SearchService,
28     private purchaseService: PurchaseService,
29     private formBuilder: UntypedFormBuilder,
30     private router: Router) {
31
32   }
33   ngOnInit(): void {
34     this.cart = this.searchService.getCart()
35   }
36
37   onSubmit(): void {
38
39
40     this.loginService.name = this.checkoutForm.get("name")?.value;
41     this.loginService.contact = this.checkoutForm.get("contact")?.value;
42     this.loginService.credit = this.checkoutForm.get("credit")?.value;
43     this.purchaseService.postPurchase(this.createOrder()).subscribe({
44       next: (data: any) => {
45
46         let retVal = JSON.parse(JSON.stringify(data))
47         if (JSON.stringify(data) === ("1")) {
48           this.searchService.cart = [];
49           this.router.navigate(["ordens"])
```

File - C:\Users\giann\Documents\COD\isl-capstoneproject-foodbox\src\main\resources\frontend\foodbox-app\src\app\checkout\checkout.component.ts

```
50     }
51   },
52   error: err => {
53     console.error(err.message)
54     alert("onSubmit error: " + err.message);
55   }
56 });
57 }
58
59 createOrder(): IOrden {
60   this.searchService.orden = {user: this.checkoutForm.value, cart: this.searchService.cart, status: "PENDING"}
61   return this.searchService.orden;
62 }
63 }
64 }
65 }
```

File - C:\Users\giann\Documents\COD\sl-capstoneproject-foodbox\src\main\resources\frontend\foodbox-app\src\app\checkout\checkout.component.css

```
1 .cell {  
2   border: 1px solid red;  
3 }
```

File - C:\Users\giann\Documents\COD\isl-capstoneproject-foodbox\src\main\resources\frontend\foodbox-app\src\app\checkout\checkout.component.html

```
1 <div class="row">
2   <div class="col-6">
3
4
5   <div class="card">
6     <div class="card-header">
7       Shipping
8     </div>
9     <div class="card-body">
10      <h5 class="card-title">Customer's information</h5>
11      <p class="card-title mb-4">Update your information if needed.</p>
12
13      <form (ngSubmit)="onSubmit()" [formGroup]="checkoutForm">
14
15        <div class="mb-2">
16          <label class="form-label mb-1" for="id">ID</label>
17          <input class="form-control" formControlName="id" id="id" placeholder="id" type="text" [readOnly]="true">
18        </div>
19        <div class="mb-2">
20          <label class="form-label mb-1" for="name">Name</label>
21          <input class="form-control" formControlName="name" id="name" placeholder="name" type="text" [readOnly]="false">
22        </div>
23        <div class="mb-2">
24          <label class="form-label mb-1" for="contact">Contact</label>
25          <input class="form-control" formControlName="contact" id="contact" placeholder="contact"
26            type="text" [readOnly]="false">
27        </div>
28        <div class="mb-4">
29          <label class="form-label mb-1" for="credit">Credit Card</label>
30          <input class="form-control"
31            formControlName="credit"
32            id="credit"
33            placeholder="No worries, it's on the house!"
34            type="text">
35        </div>
36      </div>
37      <button *ngIf="cart.length==0" [disabled]="true" class="btn btn-primary" type="submit">Place Your Order</button>
38      <button *ngIf="cart.length>0" class="btn btn-primary" type="submit">Place Your Order</button>
39      
40    </div>
41  </form>
42 </div>
43 </div>&nbsp;   
44
45 </div>
46 <div class="col-6">
47
48   <div class="card">
49     <div class="card-header">
```

File - C:\Users\giann\Documents\COD\is1-capstoneproject-foodbox\src\main\resources\frontend\foodbox-app\src\app\checkout\checkout.component.html

```
50     Total: {{cartTotal|currency}}
51   </div>
52   <div class="card-body">
53     <h5 class="card-title">Cart Content</h5>
54     <p *ngIf="cart.length<=0" class="card-text"> - empty - </p>
55   </div>
56
57   <ul *ngFor="let food of cart" class="list-group list-group-flush">
58     <li class="list-group-item">{{food.price|currency}} {{food.foodName}}</li>
59   </ul>
60 </div>
61 </div>
62 </div>
```

File - C:\Users\giann\Documents\COD\src\capstoneproject\foodbox\src\main\resources\frontend\foodbox-app\src\app\purchase\purchase.service.ts

```
1 import {Injectable} from '@angular/core';
2 import {IFood} from "../model/food";
3 import {HttpClient, HttpResponse, Headers} from "@angular/common/http";
4 import {Observable, throwError} from "rxjs";
5 import {catchError, tap} from "rxjs/operators";
6 import {IOrden} from "../model/orden";
7
8 @Injectable({
9   providedIn: 'root'
10 })
11 export class PurchaseService {
12   // baseUrl: string = 'api/foods.json';
13   baseUrl: string = "http://foodboxapp-env.eba-5ye8smp8.us-west-2.elasticbeanstalk.com"
14   foodUrl: string = this.baseUrl + '/food/getAllFoods';
15   postPurchaseUrl: string = this.baseUrl + '/cart/postPurchase';
16   cart: IFood[] = [];
17   absoluteTotal: number = 0;
18
19   constructor(private http: HttpClient) {
20   }
21
22   postPurchase(orden: IOrden): Observable<any> {
23
24     const httpOptions = {
25       headers: new HttpHeaders({ 'Accept': 'application/json', 'Content-Type': 'application/json' })
26     };
27
28     return this.http.post<any>(<this>.postPurchaseUrl, JSON.stringify(orden), httpOptions).pipe(
29       tap(data => (data)),
30       catchError(<this>.handleError)
31     );
32   }
33
34   handleError(error: HttpResponse) {
35     let errorMessage
36     if (error instanceof ErrorEvent) {
37       errorMessage = `An error occurred: ${error.error.message}`;
38     } else {
39       errorMessage = `Server returned code: ${error.status}, error message is: ${error.message}`;
40     }
41     console.error(errorMessage);
42     return throwError(errorMessage);
43   }
44 }
```

File - C:\Users\giann\Documents\CODE\sl-capstoneproject-foodbox\src\main\resources\frontend\foodbox-app\src\app\purchase\purchase.component.ts

```
1 import {Component, OnInit} from '@angular/core';
2 import {Router} from "@angular/router";
3 import {SearchService} from "../search/search.service";
4 import {IUser} from "../model/user";
5 import {IFood} from "../model/food";
6
7 @Component({
8   selector: 'ks-purchase',
9   templateUrl: './purchase.component.html',
10  styleUrls: ['./purchase.component.css']
11 })
12 export class PurchaseComponent implements OnInit {
13
14   user!: IUser;
15   cart!: IFood[];
16   absoluteTotal: number = 0;
17
18   constructor(private router: Router,
19               private searchService: SearchService) {
20   }
21
22   ngOnInit(): void {
23     this.user = this.searchService.orden.user
24     this.cart = this.searchService.orden.cart
25     this.absoluteTotal = this.searchService.absoluteTotal
26   }
27
28   jumpToSearch() {
29     this.router.navigate(['/search'])
30   }
31 }
32
```


File - C:\Users\giann\Documents\COD\sl-capstoneproject-foodbox\src\main\resources\frontend\foodbox-app\src\app\purchase\purchase.component.css

```
1 .cell {  
2   border: 1px solid red;  
3 }
```

File - C:\Users\giann\Documents\CODE\is1-capstoneproject-foodbox\src\main\resources\frontend\foodbox-app\src\app\purchase\purchase.component.html

```
1 <div class="row">
2   <div class="col-1"></div>
3   <div class="col-9">
4     <div class="card">
5       <div class="card-header">
6         Breakdown of Your Order
7       </div>
8       <div class="card-body">
9         <h5 class="card-title mb-3">Thank you for your order!</h5>
10        <div>
11          <button (click)="jumpToSearch()" class="btn btn-primary mb-3">Back to Search</button>
12        </div>
13        <div class="row">
14          <div class="col-6">
15
16            <div class="card" style="width: 18rem;">
17              <div class="card-header">
18                Client's Info
19              </div>
20              <ul class="list-group list-group-flush">
21                <li class="list-group-item">Name: {{user.username}}</li>
22                <li class="list-group-item">Address: {{user.contact}}</li>
23                <li class="list-group-item">Credit Card: </li>
24              </ul>
25            </div>
26          </div>
27          <div class="col-6">
28
29
30        <div class="row bg-light border rounded p-2">
31          <div class="col-9 fw-bold">Food Name</div>
32          <div class="col-3 fw-bold">Price</div>
33        </div>
34        <div *ngFor="let food of cart" class="row border rounded p-2">
35          <div class="col-9">{{food.foodName}}</div>
36          <div class="col-3">{{food.price|currency}}</div>
37        </div>
38        <div *ngIf="cart.length>0" class="row bg-light border rounded p-2">
39          <div class="col-9 fw-bold">Total</div>
40          <div class="col-3 fw-bold">{{absoluteTotal|currency}}</div>
41        </div>
42        <p *ngIf="cart.length<=0" class="card-text text-center mt-4"> - empty - </p>
43
44
45      </div>
46    </div>
47  </div>
48 </div>
49
```

File - C:\Users\giann\Documents\COD\sl-capstoneproject-foodbox\src\main\resources\frontend\foodbox-app\src\app\purchase\purchase.component.html

```
50 </div>
51 <div class="col-2"></div>
52 </div>
```

File - C:\Users\giann\Documents\CODE\sl-capstoneproject-foodbox\src\main\resources\frontend\foodbox-app\src\app\inventory\inventory.service.ts

```
1 import {Injectable} from '@angular/core';
2 import {HttpClient, HttpResponse} from "@angular/common/http";
3 import {Observable, throwError} from "rxjs";
4 import {catchError, tap} from "rxjs/operators";
5 import {IFood} from "../model/food";
6
7 @Injectable({
8   providedIn: 'root'
9 })
10 export class InventoryService {
11   urlSave: string = "https://62e8570a249bb1284ead379a.mockapi.io/api/v1/foods";
12   urlDelete: string = "https://62e8570a249bb1284ead379a.mockapi.io/api/v1/foods";
13   urlGet: string = "https://62e8570a249bb1284ead379a.mockapi.io/api/v1/foods";
14   admin: boolean = false;
15
16   constructor(private http: HttpClient) {
17   }
18
19   saveNewItem(inventoryGroup: any): Observable<any> {
20     return this.http.post(this.urlSave, inventoryGroup).pipe(
21       tap(data => (data)),
22       catchError(this.handleError)
23     );
24   }
25
26   deleteItem(foodId: number): Observable<any> {
27     return this.http.delete(this.urlDelete + "/" + foodId).pipe(
28       tap(data => (data)),
29       catchError(this.handleError)
30     );
31   }
32
33   getFoods(): Observable<IFood[]> {
34     return this.http.get<IFood[]>(this.urlGet).pipe(
35       tap(data => data),
36       catchError(this.handleError)
37     )
38   }
39
40   handleError(err: HttpResponse) {
41     let errorMessage = '';
42     if (err.error instanceof ErrorEvent) {
43       errorMessage = `An error occurred: ${err.error.message}`;
44     } else {
45       errorMessage = `Server returned code: ${err.status}, error message is: ${err.message}`;
46     }
47     console.error(errorMessage);
48     return throwError(() => errorMessage);
49   }
```

File - C:\Users\giann\Documents\COD\sl-capstoneproject-foodbox\src\main\resources\frontend\foodbox-app\src\app\inventory\inventory.service.ts
50 }
51

File - C:\Users\giann\Documents\CODE\is1-capstoneproject-foodbox\src\main\resources\frontend\foodbox-app\src\app\inventory\inventory.component.ts

```
1 import {Component, OnInit} from '@angular/core';
2 import {UntypedFormBuilder} from "@angular/forms";
3 import {IFood} from "../model/food";
4 import {Subscription} from "rxjs";
5 import {InventoryService} from "../inventory.service";
6 import {Router} from "@angular/router";
7 import {NotificationService} from "../notification/notification.service";
8 import {LoginService} from "../login/login.service";
9
10 @Component({
11   selector: 'fb-inventory',
12   templateUrl: '',
13   styleUrls: ['./inventory.component.css']
14 })
15 export class InventoryComponent implements OnInit {
16   inventoryGroup = this.formBuilder.group({
17     foodId: '',
18     foodName: '',
19     price: ''
20   })
21
22   sub!: Subscription;
23   errMsg!: string;
24   foods: IFood[] = [];
25   result: any;
26   admin: boolean = this.inventoryService.admin;
27
28   constructor(private FormBuilder: UntypedFormBuilder,
29     private inventoryService: InventoryService,
30     private router: Router,
31     private loginService: LoginService,
32     private notificationService: NotificationService
33   ) {
34   }
35
36   onSubmit(): void {
37     this.inventoryService.saveNewItem(this.inventoryGroup.value).subscribe({
38       next: (data: any) => {
39         const parsed = JSON.parse(JSON.stringify(data))
40         let retVal: string = parsed.return
41         if (retVal === "OK") {
42           alert("Item saved successfully.")
43           this.refreshList();
44         } else {
45           alert("error: ");
46         }
47       },
48       error: (error: any) => {
49         alert("error: " + error.message)
```

File - C:\Users\giann\Documents\COD\sl-capstoneproject-foodbox\src\main\resources\frontend\foodbox-app\src\app\inventory\inventory.component.ts

```
50     }
51     });
52 }
53
54 onDelete(item: IFood) {
55     this.inventoryService.deleteItem(item.id).subscribe({
56         next: (data) => {
57             const deleteOp = JSON.parse(JSON.stringify(data))
58             let retVal = deleteOp.return
59             if (retVal === "OK") {
60                 alert("Item deleted successfully.")
61                 this.refreshList();
62             } else {
63                 alert("error:");
64             }
65         },
66         error: (error: any) => {
67             alert("error: " + error.message)
68         }
69     });
70 }
71
72 ngOnInit(): void {
73     this.refreshList();
74 }
75
76 refreshList(): void {
77     this.sub = this.inventoryService.getFoods().subscribe({
78         next: data => {
79             this.foods = data
80         },
81         error: error => this.errMsg = error
82     });
83 }
84
85 ngOnDestroy(): void {
86     this.sub.unsubscribe();
87 }
88
89 jumpToLogin() {
90     this.router.navigate(["/login"])
91 }
92 }
93 }
```

File - C:\Users\giann\Documents\COD\esi-capstoneproject-foodbox\src\main\resources\frontend\foodbox-app\src\app\inventory\inventory.component.css

```
1 .cell {  
2   /*border: 1px solid red;*/  
3 }
```


File - C:\Users\giann\Documents\COD\lsl-capstoneproject-foodbox\src\main\resources\frontend\foodbox-app\src\app\notification\notification.message.ts

```
1 export class NotificationMessage {
2   text: string = "";
3   type: NotificationType = 0;
4 }
5
6 export enum NotificationType {
7   success = 0,
8   warning = 1,
9   error = 2,
10  info = 3
11 }
12
13
```

File - C:\Users\giann\Documents\CODE\sl-capstoneproject-foodbox\src\main\resources\frontend\foodbox-app\src\app\notification\notification.service.ts

```
1 import {Injectable} from '@angular/core';
2 import {NotificationMessage, NotificationType} from './notification.message';
3 import {ToastService} from "ngx-toastr";
4 import {Subject} from "rxjs";
5
6 @Injectable({
7
8   providedIn: 'root'
9 })
10 export class NotificationService {
11
12   notificationSubject: Subject<NotificationMessage> = new Subject<NotificationMessage>();
13
14   constructor(private toastrService: ToastService) {
15
16
17     this.notificationSubject.subscribe(message => {
18       switch (message.type) {
19         case NotificationType.success:
20           this.toastrService.success(message.text);
21           break;
22         case NotificationType.error:
23           this.toastrService.error(message.text);
24           break;
25         case NotificationType.warning:
26           this.toastrService.warning(message.text);
27           break;
28         case NotificationType.info:
29           this.toastrService.info(message.text);
30           break;
31       }
32     });
33
34
35   }
36
37   sendMessage(message: NotificationMessage): void {
38     this.notificationSubject.next(message);
39
40
41   }
42
43 }
44
```