File - C:\Users\giann\Documents\CODE\sl-project4-kitchen-story\src\main.ts

```ts
import { enableProdMode } from '@angular/core';
import { platformBrowserDynamic } from '@angular/platform-browser-dynamic';

import { AppModule } from './app/app.module';
import { environment } from './environments/environment';

if (environment.production) {
  enableProdMode();
}

platformBrowserDynamic().bootstrapModule(AppModule)
  .catch(err => console.error(err));

```

File - C:\Users\giann\Documents\CODE\sl-project4-kitchen-story\src\test.ts

```ts
// This file is required by karma.conf.js and loads recursively all the .spec and framework files

import 'zone.js/testing';
import { getTestBed } from '@angular/core/testing';
import {
  BrowserDynamicTestingModule,
  platformBrowserDynamicTesting
} from '@angular/platform-browser-dynamic/testing';

declare const require: {
  context(path: string, deep?: boolean, filter?: RegExp): {
    keys(): string[];
    <T>(id: string): T;
  };
};

// First, initialize the Angular testing environment.
getTestBed().initTestEnvironment(
  BrowserDynamicTestingModule,
  platformBrowserDynamicTesting(),
  { teardown: { destroyAfterEach: true }},
);

// Then we find all the tests.
const context = require.context('./', true, /\.spec\.ts$/);
// And load the modules.
context.keys().map(context);

```

```
1
2 // TO DO
3
4 - Checkout
5
6 -login
```

File - C:\Users\giann\Documents\CODE\sl-project4-kitchen-story\src\index.html

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>SimplilearnKitchenStory</title>
  <base href="/">
  <meta name="viewport" content="width=device-width, initial-scale=1" >
  <link rel="icon" type="image/x-icon"  href="favicon.ico">
</head>
<body>
  <ks-root></ks-root>
</body>
</html>

```

File - C:\Users\giann\Documents\CODE\sl-project4-kitchen-story\src\styles.css

```
1 /* You can add global styles to this file, and also import other style files */
2 @import "~bootstrap/dist/css/bootstrap.min.css";
3 @import "~font-awesome/css/font-awesome.min.css";
4
```

```
1  /**
2   * This file includes polyfills needed by Angular and is loaded before the app.
3   * You can add your own extra polyfills to this file.
4   *
5   * This file is divided into 2 sections:
6   *   1. Browser polyfills. These are applied before loading ZoneJS and are sorted by
    browsers.
7   *   2. Application imports. Files imported after ZoneJS that should be loaded before
    your main
8   *      file.
9   *
10  * The current setup is for so-called "evergreen" browsers; the last versions of browsers
     that
11  * automatically update themselves. This includes Safari >= 10, Chrome >= 55 (including
    Opera),
12  * Edge >= 13 on the desktop, and iOS 10 and Chrome on mobile.
13  *
14  * Learn more in https://angular.io/guide/browser-support
15  */
16
17 /***************************************************************************************
    ************
18  * BROWSER POLYFILLS
19  */
20
21 /**
22  * IE11 requires the following for NgClass support on SVG elements
23  */
24 // import 'classlist.js';  // Run `npm install --save classlist.js`.
25
26 /**
27  * Web Animations `@angular/platform-browser/animations`
28  * Only required if AnimationBuilder is used within the application and using IE/Edge or
    Safari.
29  * Standard animation support in Angular DOES NOT require any polyfills (as of Angular 6.
    0).
30  */
31 // import 'web-animations-js';  // Run `npm install --save web-animations-js`.
32
33 /**
34  * By default, zone.js will patch all possible macroTask and DomEvents
35  * user can disable parts of macroTask/DomEvents patch by setting following flags
36  * because those flags need to be set before `zone.js` being loaded, and webpack
37  * will put import in the top of bundle, so user need to create a separate file
38  * in this directory (for example: zone-flags.ts), and put the following flags
39  * into that file, and then add the following code before importing zone.js.
40  * import './zone-flags';
41  *
42  * The flags allowed in zone-flags.ts are listed here.
43  *
44  * The following flags will work for all browsers.
45  *
46  * (window as any).__Zone_disable_requestAnimationFrame = true; // disable patch
    requestAnimationFrame
47  * (window as any).__Zone_disable_on_property = true; // disable patch onProperty such as
     onclick
48  * (window as any).__zone_symbol__UNPATCHED_EVENTS = ['scroll', 'mousemove']; // disable
    patch specified eventNames
49  *
50  *  in IE/Edge developer tools, the addEventListener will also be wrapped by zone.js
51  *  with the following flag, it will bypass `zone.js` patch for IE/Edge
52  *
53  *  (window as any).__Zone_enable_cross_context_check = true;
54  *
```

```
 */

/***************************************************************************************************
 * Zone JS is required by default for Angular itself.
 */
import 'zone.js';  // Included with Angular CLI.


/***************************************************************************************************
 * APPLICATION IMPORTS
 */

```

File - C:\Users\giann\Documents\CODE\sl-project4-kitchen-story\src\app\app.module.ts

```ts
import {NgModule} from '@angular/core';
import {BrowserModule} from '@angular/platform-browser';
import {RouterModule} from '@angular/router';
import {AppComponent} from './app.component';
import {LoginComponent} from './login/login.component';
import {SearchComponent} from './search/search.component';
import {FormsModule, ReactiveFormsModule} from "@angular/forms";
import {HttpClientModule} from "@angular/common/http";
import {CartComponent} from './cart/cart.component';
import {CheckoutComponent} from './cart/checkout.component';
import {InventoryComponent} from './inventory/inventory.component';
import {PurchaseComponent} from './cart/purchase.component';
import {ChangeComponent} from './login/change.component';

@NgModule({
    declarations: [
        AppComponent,
        LoginComponent,
        SearchComponent,
        CartComponent,
        CheckoutComponent,
        InventoryComponent,
        PurchaseComponent,
        ChangeComponent
    ],
    imports: [
        BrowserModule,
        HttpClientModule,
        RouterModule.forRoot([
            {path: '', component: SearchComponent},
            {path: 'search', component: SearchComponent},
            {path: 'login', component: LoginComponent},
            {path: 'inventory', component: InventoryComponent},
            {path: 'cart', component: CartComponent},
            {path: 'checkout', component: CheckoutComponent},
            {path: 'purchase', component: PurchaseComponent},
            {path: 'changePassword', component: ChangeComponent},
        ]),
        FormsModule,
        ReactiveFormsModule
    ],
    providers: [],
    bootstrap: [AppComponent]
})
export class AppModule {
}
```

File - C:\Users\giann\Documents\CODE\sl-project4-kitchen-story\src\app\app.component.ts

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

@Component({
  selector: 'ks-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.css']
})
export class AppComponent {
  pageTitle = 'Kitchen Story';

  activeTab = 'search';

  search(activeTab: string){
    this.activeTab = activeTab;
  }

  result(activeTab: string){
    this.activeTab = activeTab;
  }
}

```

File - C:\Users\giann\Documents\CODE\sl-project4-kitchen-story\src\app\app.component.css

```css

.cell {
    border: 1px solid red;
}

.cell-gray {
    border: 1px solid lightgray;
    background-color: white;
}

.btn-primary, .btn-primary:hover, .btn-primary:active, .btn-primary:visited {

}
```

File - C:\Users\giann\Documents\CODE\sl-project4-kitchen-story\src\app\app.component.html

```
<div class="alert alert-info mb-1 p-1"  role="alert">
    <div class="row">
        <div class="col-4">

        </div>
        <div class="col-4">
            <h1 class='text-center'>
                <span class="badge bg-info">Kitchen
                    <span class="badge bg-warning">Story</span></span>
            </h1>

        </div>

        <div class="col-4 align-middle">
            <div class="container d-flex justify-content-end" >
                <div class="p-2 align-baseline">
                <span style="color:#A0A0A0;">Java Full Stack Bootcamp<br/>
                Gianni Fontanot - Angular</span>
                </div>
            </div>
        </div>
    </div>
</div>

<div class="row" style="min-height: 100vh">
    <div class="col-2 d-flex justify-content-end">
        <div aria-orientation="vertical" class="nav flex-column nav-pills"  id="v-pills-tab" role="tablist">
            <a [routerLink]="['/search']" class='nav-link cell-gray'  routerLinkActive='active'>
                <i aria-hidden="true" class="fa fa-search"></i>  Search</a>
            <a [routerLink]="['/cart']" class='nav-link cell-gray'  routerLinkActive='active'>
                <i aria-hidden="true" class="fa fa-shopping-cart"></i>  Cart  
                <!--
                <span class="badge bg-danger">4</span>
                -->
            </a>
            <a [routerLink]="['/checkout']" class='nav-link cell-gray'  routerLinkActive='active'>
                <i aria-hidden="true" class="fa fa-credit-card"></i>   Checkout</a>
            <a [routerLink]="['/login']" class='nav-link cell-gray'  routerLinkActive='active'>
                <i aria-hidden="true" class="fa fa-sign-in"></i>  Admin login</a>
            <a [routerLink]="['/inventory']" class='nav-link cell-gray'  routerLinkActive='active'>
                <i aria-hidden="true" class="fa fa-list"></i>  Inventory</a>
        </div>
    </div>


    <div class="col-10">
        <router-outlet></router-outlet>
    </div>
</div>


```

File - C:\Users\giann\Documents\CODE\sl-project4-kitchen-story\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
22   removeItem(food: IFood) {
23     this.searchService.removeItem(food)
24     this.total = this.searchService.getTotal()
25   }
26 }
27
```

File - C:\Users\giann\Documents\CODE\sl-project4-kitchen-story\src\app\cart\cart.component.html

```
<div class="card">
    <div class="card-body">
        <div class="row cell">
            <div class="cell col-2"></div>
            <div class="cell col-7">
                <div class="row cell bg-light border rounded p-1" >
                    <div class="cell col-2 bg-gray"></div>
                    <div class="cell col-1">ID</div>
                    <div class="cell col-6">Food Name</div>
                    <div class="cell col-2">Price</div>
                    <div class="cell col-1"></div>
                </div>
                <div *ngFor="let food of cart" class="row cell border rounded p-1">
                    <div class="cell col-2 bg-gray">
                        <button (click)="removeItem(food)" class="btn btn-outline-
danger btn-sm">
                            delete
                        </button>
                    </div>
                    <div class="cell col-1">{{food.foodId}}</div>
                    <div class="cell col-6">{{food.foodName}}</div>
                    <div class="cell col-2">{{food.price|currency}}</div>
                    <div class="cell col-1"></div>
                </div>
                <p *ngIf="cart.length<=0" class="card-text text-center mt-4"> - empty
 - </p>
            </div>
            <div class="cell col-3"></div>
        </div>
    </div>
</div>

```

File - C:\Users\giann\Documents\CODE\sl-project4-kitchen-story\src\app\cart\checkout.component.ts

```
1  import {Component, OnInit} from '@angular/core';
2  import {SearchService} from "../search/search.service";
3  import {IFood} from "../model/food";
4  import {FormBuilder} from "@angular/forms";
5  import {Router} from "@angular/router";
6  import {IOrder} from "../model/order";
7
8  @Component({
9      selector: 'ks-checkout',
10     templateUrl: './checkout.component.html',
11     styleUrls: ['./checkout.component.css']
12 })
13 export class CheckoutComponent implements OnInit {
14     cart: IFood[] = []
15     cartTotal = this.searchService.absoluteTotal
16     checkoutForm = this.formBuilder.group({
17         name: '',
18         address: '',
19         credit: ''
20     });
21
22     constructor(private searchService: SearchService,
23                 private formBuilder: FormBuilder,
24                 private router: Router) {
25     }
26
27     ngOnInit(): void {
28         this.cart = this.searchService.getCart()
29     }
30
31     onSubmit(): void {
32         this.searchService.postPurchase(this.createOrder()).subscribe({
33             next: (data: any) => {
34                 const parsed = JSON.parse(JSON.stringify(data))
35
36
37                 let retVal = parsed.return
38                 console.log(">>> " + retVal)
39                 if (retVal === ("OK")) {
40
41                     this.router.navigate(["/purchase"])
42                 }
43             },
44             error: err => {
45                 console.error(err.message)
46                 alert("error: " + err.message);
47             }
48         });
49     }
50
51     createOrder(): IOrder {
52         this.searchService.order = {client: this.checkoutForm.value, cart: this.
   searchService.cart}
53         return this.searchService.order;
54     }
55 }
```

File - C:\Users\giann\Documents\CODE\sl-project4-kitchen-story\src\app\cart\purchase.component.ts

```ts
import {Component, OnInit} from '@angular/core';
import {Router} from "@angular/router";
import {SearchService} from "../search/search.service";
import {IClient} from "../model/client";
import {IFood} from "../model/food";

@Component({
    selector: 'ks-purchase',
    templateUrl: './purchase.component.html',
    styleUrls: ['./purchase.component.css']
})
export class PurchaseComponent implements OnInit {

    client!: IClient;
    cart!: IFood[];
    absoluteTotal: number = 0;

    constructor(private router: Router,
                private searchService: SearchService) {
    }

    ngOnInit(): void {
        this.client = this.searchService.order.client
        this.cart = this.searchService.order.cart
        this.absoluteTotal = this.searchService.absoluteTotal
    }

    jumpToSearch() {
        this.router.navigate(['/search'])
    }
}

```

File - C:\Users\giann\Documents\CODE\sl-project4-kitchen-story\src\app\cart\checkout.component.css

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

File - C:\Users\giann\Documents\CODE\sl-project4-kitchen-story\src\app\cart\purchase.component.css

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

```
<div class="row">
    <div class="col-6">


        <div class="card">
            <div class="card-header">
                Shipping
            </div>
            <div class="card-body">
                <h5 class="card-title">Customer's information</h5>

                <form (ngSubmit)="onSubmit()" [formGroup]="checkoutForm">

                    <div class="mb-2">
                        <label class="form-label mb-1" for="name">Name</label>
                        <input class="form-control" formControlName="name" id="name" placeholder="name" type="text">
                    </div>
                    <div class="mb-2">
                        <label class="form-label mb-1" for="address">Address</label>
                        <input class="form-control" formControlName="address" id="address" placeholder="address"
                               type="text">
                    </div>
                    <div class="mb-4">
                        <label class="form-label mb-1" for="credit">Credit Card</label>
                        <input class="form-control"
                               formControlName="credit"
                               id="credit"
                               placeholder="0000-0000-0000-0000"
                               type="text">
                    </div>

                    <div>
                        <button class="btn btn-primary" type="submit">Purchase</button>
                        <img alt="credit card" src="assets/images/we_accept.png" width="70%"/>
                    </div>
                </form>
            </div>
        </div>
         

    </div>
    <div class="col-6">

        <div class="card">
            <div class="card-header">
                Total: {{cartTotal|currency}}
            </div>
            <div class="card-body">
                <h5 class="card-title">Cart Content</h5>
                <p *ngIf="cart.length<=0" class="card-text"> - empty - </p>

            </div>


            <ul *ngFor="let food of cart" class="list-group list-group-flush">
                <li class="list-group-item">{{food.price|currency}} {{food.foodName}}</li>

            </ul>
            <!--            <div *ngFor="let food of cart" class="row border rounded p-1">-->
```

```
60              <!--                -->
61              <!--                <div class="row">-->
62              <!--                    <div class="col-2 text-center">{{food.foodId}}</div
    >-->
63              <!--                    <div class="col-8">{{food.foodName}}</div>-->
64              <!--                    <div class="col-2">{{food.price|currency}}</div>-->
65              <!--                </div>-->
66              <!--            </div>-->
67          </div>
68      </div>
69  </div>
```

File - C:\Users\giann\Documents\CODE\sl-project4-kitchen-story\src\app\cart\purchase.component.html

```
<div class="row">
    <div class="col-1"></div>
    <div class="col-9">
        <div class="card">
            <div class="card-header">
                Breakdown of Your Order
            </div>
            <div class="card-body">
                <h5 class="card-title mb-3">Thank you for your purchase!</h5>
                <div>
                    <button (click)="jumpToSearch()" class="btn btn-primary mb-3">Back to Search</button>
                </div>
                <div class="row">
                    <div class="col-6">

                        <div class="card" style="width: 18rem;">
                            <div class="card-header">
                                Client's Info
                            </div>
                            <ul class="list-group list-group-flush">
                                <li class="list-group-item">Name: {{client.name}}</li>
                                <li class="list-group-item">Address: {{client.address}}</li>
                                <li class="list-group-item">Credit Card: {{client.credit}}</li>
                            </ul>
                        </div>
                    </div>
                    <div class="col-6">


                        <div class="row bg-light border rounded p-2">
                            <div class="col-9 fw-bold">Food Name</div>
                            <div class="col-3 fw-bold">Price</div>
                        </div>
                        <div *ngFor="let food of cart" class="row border rounded p-2">
                            <div class="col-9">{{food.foodName}}</div>
                            <div class="col-3">{{food.price|currency}}</div>
                        </div>
                        <div *ngIf="cart.length>0" class="row bg-light border rounded p-2">
                            <div class="col-9 fw-bold">Total</div>
                            <div class="col-3 fw-bold">{{absoluteTotal|currency}}</div>
                        </div>
                        <p *ngIf="cart.length<=0" class="card-text text-center mt-4">
 - empty - </p>


                    </div>
                </div>

            </div>
        </div>
    </div>
    <div class="col-2"></div>
</div>
```

File - C:\Users\giann\Documents\CODE\sl-project4-kitchen-story\src\app\login\login.service.ts

```ts
import {Injectable} from '@angular/core';
import {HttpClient, HttpErrorResponse} from "@angular/common/http";
import {Observable, throwError} from "rxjs";
import {catchError, tap} from "rxjs/operators";

@Injectable({
    providedIn: 'root'
})
export class LoginService {
    userUrl: string = 'https://62e8570a249bb1284ead379a.mockapi.io/api/v1/users' ;
    putUrl: string = 'https://62e8570a249bb1284ead379a.mockapi.io/api/v1/users/' ;

    constructor(private http: HttpClient) {
    }

    submitLogin(formValue: any): Observable<any> {
        return this.http.post(this.userUrl, formValue).pipe(
            tap(data => data),
            catchError(err => this.handleError(err))
        )

    }

    changePassword(formValue: any): Observable<any> {
        return this.http.put(this.putUrl + "/" + formValue.idx, formValue).pipe(
            tap(data => data),
            catchError(err => this.handleError(err))
        );
    }

    handleError(error: HttpErrorResponse) {
        let errorMessage
        if (error.error instanceof ErrorEvent) {
            errorMessage = `An error occurred: ${error.error.message}`;
        } else {
            errorMessage = `Server returned code: ${error.status}, error message is: ${
error.message}`;
        }
        console.error(errorMessage);
        return throwError(errorMessage);
    }

}

```

File - C:\Users\giann\Documents\CODE\sl-project4-kitchen-story\src\app\login\login.component.ts

```ts
import {Component, OnInit} from '@angular/core';
import {FormBuilder} from "@angular/forms";
import {LoginService} from "./login.service";
import {Router} from "@angular/router";
import {InventoryService} from "../inventory/inventory.service";


@Component({
    templateUrl: './login.component.html',
    styleUrls: ['./login.component.css']
})
export class LoginComponent implements OnInit {

    loginForm = this.formBuilder.group({
        id: '1',
        username: 'admin',
        password: 'xadminw'

    })

    constructor(private formBuilder: FormBuilder,
                private loginService: LoginService,
                private router: Router,
                private inventoryService: InventoryService) {
    }

    ngOnInit(): void {
    }

    onSubmit() {
        this.loginService.submitLogin(this.loginForm.value).subscribe(
            {
                next: (data) => {
                    console.log(data)
                    this.inventoryService.admin = true;
                    this.router.navigate(["/inventory"])
                },
                error: (err => console.log(err))
            }
        )
    }

    onChangePassword() {
        this.router.navigate(["/changePassword"])

    }
}

```

File - C:\Users\giann\Documents\CODE\sl-project4-kitchen-story\src\app\login\change.component.ts

```ts
import {Component, OnInit} from '@angular/core';
import {FormBuilder} from "@angular/forms";
import {LoginService} from "./login.service";
import {Router} from "@angular/router";

@Component({
    selector: 'ks-change',
    templateUrl: './change.component.html',
    styleUrls: ['./change.component.css']
})
export class ChangeComponent implements OnInit {

    loginGroup = this.formBuilder.group({
        idx: '1',
        usernamex: 'admin',
        oldPassword: 'admin',
        newPassword: '',
        newPassword2: ''
    })


    constructor(private formBuilder: FormBuilder,
                private loginService: LoginService,
                private router: Router) {
    }

    ngOnInit(): void {
    }


    onSubmit(): void {

        if (this.loginGroup.value.newPassword !== this.loginGroup.value.newPassword2) {
            alert("New Password does not match. Please try again." )
        } else {
            this.loginService.changePassword(this.loginGroup.value).subscribe({
                next: (data: any) => {
                    const parsed = JSON.parse(JSON.stringify(data))
                    let retVal = parsed.return
                    console.log(">>> " + retVal)
                    if (retVal === ("OK")) {
                        alert("Password successfully changed")
                        this.router.navigate(["/inventory"])
                    }
                },
                error: err => {
                    console.error(">>> " + err)
                    alert("error: >>>" + err);
                }
            })
        }
    }
}

```

File - C:\Users\giann\Documents\CODE\sl-project4-kitchen-story\src\app\login\login.component.html

```
<div class="row">
    <div class="col-2"></div>
    <div class="col-6">
        <div class="card">
            <div class="card-header">
                Welcome!
            </div>
            <div class="card-body">
                <h5 class="card-title">Admin Login</h5>

                <form (ngSubmit)="onSubmit()" [formGroup]="loginForm">
                    <div class="mb-3">
                        <label class="form-label" for="id">ID</label>
                        <input class="form-control" formControlName="id" id="id"
placeholder="ID"
                               type="text">
                    </div>
                    <div class="mb-3">
                        <label class="form-label" for="username">Username</label>
                        <input class="form-control" formControlName="username" id="
username" placeholder="username"
                               type="text">
                    </div>
                    <div class="mb-3">
                        <label class="form-label" for="password">Password</label>
                        <input class="form-control" formControlName="password" id="
password" placeholder="password"
                               type="password">
                    </div> 

                    <div class="row">
                        <div class="col-6 d-flex justify-content-start">

                            <button class="btn btn-primary" type="submit">Login</button
>
                        </div>
                        <div class="col-6 d-flex justify-content-end">
                            <button (click)="onChangePassword()" class="btn btn-
secondary">Change Password</button>

                        </div>
                    </div>
                </form>
            </div>
        </div>
    </div>
    <div class="col-4"></div>
</div>
```

File - C:\Users\giann\Documents\CODE\sl-project4-kitchen-story\src\app\login\change.component.html

```
<div class="row">
    <div class="col-2"></div>
    <div class="col-6">
        <div class="card">
            <div class="card-header">
                Password Change
            </div>
            <div class="card-body">
                <h5 class="card-title">Change Admin Login</h5>

                <form (ngSubmit)="onSubmit()" [formGroup]="loginGroup">
                    <div class="mb-3">
                        <label class="form-label" for="idx">ID</label>
                        <input class="form-control" formControlName="idx" id="idx"
placeholder="ID"
                               type="text">
                    </div>
                    <div class="mb-3">
                        <label class="form-label" for="usernamex">Username</label>
                        <input class="form-control" formControlName="usernamex" id="
usernamex"
                               placeholder="username"
                               type="text">
                    </div>
                    <div class="mb-3">
                        <label class="form-label" for="oldPassword">Old Password</
label>
                        <input class="form-control" formControlName="oldPassword" id="
oldPassword"
                               placeholder="old password"
                               type="password">
                    </div>
                    <div class="mb-3">
                        <label class="form-label" for="newPassword2">New Password</
label>
                        <input class="form-control" formControlName="newPassword" id="
newPassword"
                               placeholder="type your new password"
                               type="password">
                    </div>
                    <div class="mb-4">
                        <label class="form-label" for="newPassword2">New Password (
again)</label>
                        <input class="form-control" formControlName="newPassword2" id
="newPassword2"
                               placeholder="type again your new password"
                               type="password">
                    </div>

                    <button class="btn btn-primary" type="submit">Change Password</
button>


                </form>
            </div>
        </div>
    </div>
    <div class="col-4"></div>
</div>
```

File - C:\Users\giann\Documents\CODE\sl-project4-kitchen-story\src\app\model\food.ts

```
1 export interface IFood {
2   foodId: number;
3   foodName: string;
4   price: number;
5 }
6
7
8
```

File - C:\Users\giann\Documents\CODE\sl-project4-kitchen-story\src\app\model\user.ts

```
1 export interface IUser{
2 name: string;
3 password: string;
4 }
```

File - C:\Users\giann\Documents\CODE\sl-project4-kitchen-story\src\app\model\order.ts

```ts
import {IFood} from "./food";
import {IClient} from "./client";

export interface IOrder {
    client: IClient;
    cart: IFood[];
}

```

File - C:\Users\giann\Documents\CODE\sl-project4-kitchen-story\src\app\model\client.ts

```ts
export interface IClient {
    name: string;
    address: string;
    credit: string;
}
```

File - C:\Users\giann\Documents\CODE\sl-project4-kitchen-story\src\app\search\search.service.ts

```ts
import {Injectable} from '@angular/core';
import {IFood} from "../model/food";
import {HttpClient, HttpErrorResponse} from "@angular/common/http";
import {Observable, throwError} from "rxjs";
import {catchError, tap} from "rxjs/operators";
import {IOrder} from "../model/order";


@Injectable({
    providedIn: 'root'
})
export class SearchService {
    // foodUrl: string = 'api/foods.json';
    foodUrl: string = 'https://62e8570a249bb1284ead379a.mockapi.io/api/v1/foods' ;
    postPurchaseUrl: string = 'https://62e8570a249bb1284ead379a.mockapi.io/api/v1/purchase';
    cart: IFood[] = [];
    order!: IOrder;
    absoluteTotal: number = 0;

    constructor(private http: HttpClient) {
    }

    getFoods(): Observable<IFood[]> {
        return this.http.get<IFood[]>(this.foodUrl).pipe(
            tap(data => console.log('All', JSON.stringify(data))),
            catchError(this.handleError)
        );
    }

    postPurchase(order: any): Observable<any> {
        return this.http.post<any>(this.postPurchaseUrl, order).pipe(
            tap(data => console.log(JSON.stringify(data))),
            catchError(this.handleError)
        );
    }

    handleError(error: HttpErrorResponse) {
        let errorMessage
        if (error.error instanceof ErrorEvent) {
            errorMessage = `An error occurred: ${error.error.message}`;
        } else {
            errorMessage = `Server returned code: ${error.status}, error message is: ${error.message}`;
        }
        console.error(errorMessage);
        return throwError(errorMessage);
    }

    addToCart(food: IFood) {
        this.cart.push(food);
    }

    getCart(): IFood[] {
        return this.cart;
    }

    removeItem(food: IFood) {

        let foodIdToDelete = (element: { foodId: number; }) => food.foodId === element.foodId;

        let index = this.cart.findIndex(foodIdToDelete)
        this.cart.splice(index, 1)
```

```
62      }
63
64      getTotal(): number {
65          this.absoluteTotal = 0;
66          this.cart.forEach(item => this.absoluteTotal += item.price)
67          return this.absoluteTotal;
68      }
69  }
70
71
```

File - C:\Users\giann\Documents\CODE\sl-project4-kitchen-story\src\app\search\search.component.ts

```ts
import {Component, OnDestroy, OnInit} from '@angular/core';
import {IFood} from "../model/food";
import {SearchService} from "./search.service";
import {Subscription} from "rxjs";

@Component({
    templateUrl: './search.component.html',
    styleUrls: ['./search.component.css']
})
export class SearchComponent implements OnInit, OnDestroy {

    foodsFiltered: IFood[] = [];
    foods: IFood[] = [];
    cart: IFood [] = []
    listFilter: string = '';
    sub!: Subscription;

    constructor(private searchService: SearchService) {
    }

    ngOnInit(): void {
        this.sub = this.searchService.getFoods().subscribe({
            next: data => {
                this.foods = data;
                this.foodsFiltered = this.foods;
                return this.foodsFiltered;
            },
            error: err => alert(err)
        });

    }

    search() {
        this.sub = this.searchService.getFoods().subscribe({
            next: data => {
                this.foods = data
                this.foodsFiltered = this.foods.filter(
                    food => food.foodName.toLowerCase().includes(
                        this.listFilter.toLowerCase()
                    )
                )
            },
            error: err => alert(err),

        });
    }

    ngOnDestroy(): void {
        //this.sub.unsubscribe();
        //food => food.foodName.toLowerCase()
        // .includes(this.listFilter.toLowerCase()
    }

    select(food: IFood) {
        this.searchService.addToCart(food);
    }
}

```

File - C:\Users\giann\Documents\CODE\sl-project4-kitchen-story\src\app\search\search.component.css

```css
body {
    background: #0D0;
}

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

File - C:\Users\giann\Documents\CODE\sl-project4-kitchen-story\src\app\search\search.component.html

```
<div class="card">
    <div class="card-header">
        Our Products
    </div>
    <div class="card-body">
        <h5 class="card-title">This is the list of foods we offer.</h5>
        <p class="card-text">Select the foods you would like to order. When you are done, click the Cart menu to see
            a list of the items. You can come back to this page to add more items to your cart if
            you want.</p>
    </div>
    <div class="row pb-3 pt-3">
        <div class="col-3"></div>
        <div class="col-5 ">
            <input [(ngModel)]="listFilter" class="form-control" placeholder="What are you looking for?" type="text">
        </div>
        <div class="col-3 ps-0 ms-0">
            <button (click)="search()" class="btn btn-primary">search</button>

        </div>
    </div>
    <div class="container">
        <div class="row">
            <div class="col-2 "></div>
            <div class="col-7 ">
                <div class="row bg-light border rounded p-1">
                    <div class="col-2 bg-gray"></div>
                    <div class="col-2 text-center">ID</div>
                    <div class="col-5">Food Name</div>
                    <div class="col-2">Price</div>
                    <div class="col-1"></div>
                </div>
                <div *ngFor="let food of foodsFiltered" class="row border rounded p-1">
                    <div class="col-2 bg-gray">
                        <button (click)="select(food)" class="btn btn-outline-primary btn-sm">select</button>
                    </div>
                    <div class="col-2 text-center">{{food.foodId}}</div>
                    <div class="col-5">{{food.foodName}}</div>
                    <div class="col-2">{{food.price|currency}}</div>
                    <div class="col-1"></div>
                </div>
            </div>
            <div class="col-3 "></div>
        </div>
    </div>
</div>
```

File - C:\Users\giann\Documents\CODE\sl-project4-kitchen-story\src\app\inventory\inventory.service.ts

```ts
import {Injectable} from '@angular/core';
import {HttpClient, HttpErrorResponse} from "@angular/common/http";
import {Observable, throwError} from "rxjs";
import {catchError, tap} from "rxjs/operators";
import {IFood} from "../model/food";

@Injectable({
    providedIn: 'root'
})
export class InventoryService {
    urlSave: string = "https://62e8570a249bb1284ead379a.mockapi.io/api/v1/foods";
    urlDelete: string = "https://62e8570a249bb1284ead379a.mockapi.io/api/v1/foods";
    urlGet: string = 'https://62e8570a249bb1284ead379a.mockapi.io/api/v1/foods';
    admin: boolean = false;

    constructor(private http: HttpClient) {
    }

    saveNewItem(inventoryGroup: any): Observable<any> {
        return this.http.post(this.urlSave, inventoryGroup).pipe(
            tap(data => console.log(JSON.stringify(data))),
            catchError(this.handleError)
        );
    }

    deleteItem(foodId: number): Observable<any> {
        return this.http.delete(this.urlDelete + "/" + foodId).pipe(
            tap(data => console.log("DELETE: " + JSON.stringify(data))),
            catchError(this.handleError)
        );
    }

    getFoods(): Observable<IFood[]> {
        return this.http.get<IFood[]>(this.urlGet).pipe(
            tap(data => console.log(JSON.stringify(data))),
            catchError(this.handleError)
        )
    }

    handleError(err: HttpErrorResponse) {
        let errorMessage = '';
        if (err.error instanceof ErrorEvent) {
            errorMessage = `An error occurred: ${err.error.message}`;
        } else {
            errorMessage = `Server returned code: ${err.status}, error message is: ${
err.message}`;
        }
        console.error(errorMessage);
        return throwError(() => errorMessage);
    }
}

```

File - C:\Users\giann\Documents\CODE\sl-project4-kitchen-story\src\app\inventory\inventory.component.ts

```
import {Component, OnInit} from '@angular/core';
import {FormBuilder} from "@angular/forms";
import {IFood} from "../model/food";
import {Subscription} from "rxjs";
import {InventoryService} from "./inventory.service";
import {Router} from "@angular/router";

@Component({
    selector: 'ks-inventory',
    templateUrl: './inventory.component.html',
    styleUrls: ['./inventory.component.css']
})
export class InventoryComponent implements OnInit {
    inventoryGroup = this.formBuilder.group({
        foodId: '',
        foodName: '',
        price: ''
    })

    sub!: Subscription;
    errMsg!: string;
    foods: IFood[] = [];
    result: any;
    admin: boolean = this.inventoryService.admin;

    constructor(private formBuilder: FormBuilder,
                private inventoryService: InventoryService,
                private router: Router) {
    }

    onSubmit(): void {
        this.inventoryService.saveNewItem(this.inventoryGroup.value).subscribe({
            next: (data: any) => {
                const parsed = JSON.parse(JSON.stringify(data))
                let retVal: string = parsed.return
                if (retVal === ("OK")) {
                    alert("Item saved successfully.")
                    this.refreshList();
                } else {
                    alert("error: ");
                }
            },
            error: (error: any) => {
                alert("error: " + error.message)
            }
        });
    }

    onDelete(item: IFood) {
        this.inventoryService.deleteItem(item.foodId).subscribe({
            next: (data) => {
                const deleteOp = JSON.parse(JSON.stringify(data))
                let retVal = deleteOp.return
                if (retVal === ("OK")) {
                    alert("Item deleted successfully.")
                    this.refreshList();
                } else {
                    alert("error:")
                }
            },
            error: (error: any) => {
                alert("error: " + error.message)
            }
        });
```

```
65      }
66
67      ngOnInit(): void {
68          this.refreshList();
69      }
70
71      refreshList(): void {
72          this.sub = this.inventoryService.getFoods().subscribe({
73              next: data => {
74                  this.foods = data
75              },
76              error: error => this.errMsg = error
77          });
78      }
79
80      ngOnDestroy(): void {
81          this.sub.unsubscribe();
82      }
83
84      jumpToLogin() {
85          this.router.navigate(["/login"])
86      }
87  }
88
```

File - C:\Users\giann\Documents\CODE\sl-project4-kitchen-story\src\app\inventory\inventory.component.css

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

File - C:\Users\giann\Documents\CODE\sl-project4-kitchen-story\src\app\inventory\inventory.component.html

```
<div *ngIf="admin==true;else other_content">
    <div class="card-body">
        <div class="row cell">
            <div class="cell col-2"></div>
            <div class="cell col-7">
                <div class="row cell bg-light border rounded p-1" >
                    <div class="cell col-2 bg-gray"></div>
                    <div class="cell col-1"></div>
                    <div class="cell col-6">Food Name</div>
                    <div class="cell col-2">Price</div>
                    <div class="cell col-1"></div>
                </div>
                <form (ngSubmit)="onSubmit()" [formGroup]="inventoryGroup">
                    <div class="row cell p-2">
                        <div class="cell col-2"></div>
                        <div class="cell col-1"></div>
                        <div class="cell col-6"><input class="form-control"
                                                       formControlName="foodName"
                                                       id="foodName"
                                                       placeholder="add a new food..."
                                                       size="25" type="text">
                        </div>
                        <div class="cell col-2"><input class="form-control"
                                                       formControlName="price"
                                                       id="price"
                                                       placeholder="0.00"
                                                       size="2"
                                                       type="text"></div>
                        <div class="cell col-1">
                            <button class="btn btn-primary" type="submit">Save</button>
                        </div>
                    </div>
                </form>

                <div *ngFor="let item of foods" class="row cell border rounded p-1">
                    <div class="cell col-2">
                        <button (click)="onDelete(item)" class="btn btn-outline-danger
btn-sm">Delete</button>
                    </div>
                    <div class="cell col-1">{{item.foodId}}</div>
                    <div class="cell col-6">{{item.foodName}}</div>
                    <div class="cell col-2">{{item.price|currency}}</div>
                    <div class="cell col-1"></div>
                </div>

            </div>
            <div class="cell col-3"></div>
        </div>
    </div>
</div>
<ng-template #other_content>

    <div class="card">
        <h5 class="card-header">Not allowed</h5>
        <div class="card-body">
            <h5 class="card-title">You need to login first</h5>
            <p class="card-text">In order to see the inventory and add or delete items,
first you must be looged in the
                application.</p>
            <a (click)="jumpToLogin()" class="btn btn-primary">Go to Login</a>
        </div>
    </div>
</ng-template>

```

File - C:\Users\giann\Documents\CODE\sl-project4-kitchen-story\src\app\inventory\inventory.component.spec.ts

```
1 import { ComponentFixture, TestBed } from '@angular/core/testing';
2
3 import { InventoryComponent } from './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\CODE\sl-project4-kitchen-story\src\environments\environment.ts

```ts
// This file can be replaced during build by using the `fileReplacements` array.
// `ng build` replaces `environment.ts` with `environment.prod.ts`.
// The list of file replacements can be found in `angular.json`.

export const environment = {
  production: false
};

/*
 * For easier debugging in development mode, you can import the following file
 * to ignore zone related error stack frames such as `zone.run`, `zoneDelegate.invokeTask`.
 *
 * This import should be commented out in production mode because it will have a negative impact
 * on performance if an error is thrown.
 */
// import 'zone.js/plugins/zone-error';  // Included with Angular CLI.

```

File - C:\Users\giann\Documents\CODE\sl-project4-kitchen-story\src\environments\environment.prod.ts

```
1 export const environment = {
2   production: true
3 };
4
```