

Kitchen Story(Write Up)

Developed By: *G Rajya Lakshmi Suvarna Himavalli*

CART(css)

```
.cart-page{
  margin-top: 40px;
  margin-bottom: 50px;
}
.cart-page h1{
  text-align: center;
}
.details {
}
.details img{
  height: 100px;
}
.details ul{
  padding: 0;
  margin: 0;
  list-style-type: none;
  display: flex;
  justify-content: space-between;
  border: 1px solid #ddd;
  padding: 15px;
  margin-bottom: 15px;
}
.summary h3{
  text-align: center;
}
.summary ul {
  border: 1px solid #ddd;
  list-style-type: none;
  padding: 15px;
}
.summary ul li{
  display: flex;
  justify-content: space-between;
```

CART(HTML):

```
<div class="cart-page">
<h1>
|   Cart Page
</h1>
<div class="row">
<div class="col-sm-8 details">
    <ul *ngFor="let cart of cartData">
        <li>
            
        </li>
        <li>
            <span> {{cart.name}}</span>
        </li>
        <li>
            <span> Price : {{cart.price}}</span>
        </li>
        <li>
            <span> Quantity : {{cart.quantity}}</span>
        </li>
        <li>
            <button class="form-button"(click)="removeToCart(cart.id)" >Remove To Cart</button>
        </li>
    </ul>
</div>
<div class="col-sm-4 summary">
<h3>Summary</h3>
<ul>
    <li> <span>Amount:</span><span>{{priceSummary.price}}</span></li>
    <li> <span>Tax:</span><span>{{priceSummary.tax}}</span></li>
    <li> <span>Delivery:</span><span>{{priceSummary.delivery}}</span></li>
    <li> <span>Discount:</span><span>{{priceSummary.discount}}</span></li>
    <li> <span><h4>Total</h4></span> <span><h4>{{priceSummary.total}}</h4></span></li>
    <button (click)="checkout()" class="form-button max-width" >Checkout</button>
</ul>
</div>
</div>
</div>
```

CART(TS):

```
1 import { Component, OnInit } from '@angular/core';
2 import { Router } from '@angular/router';
3 import { cart, priceSummary } from '../data-type';
4 import { ProductService } from '../services/product.service';
5 > @Component({ ...
9 })
10 export class CartPageComponent implements OnInit {
11   cartData: cart[] | undefined;
12   priceSummary: priceSummary = {
13     price: 0,
14     discount: 0,
15     tax: 0,
16     delivery: 0,
17     total: 0
18   }
19   constructor(private product: ProductService, private router: Router) { }
20   ngOnInit(): void {
21     this.loadDetails()
22   }
23   removeToCart(cartId:number|undefined){
24     cartId && this.cartData && this.product.removeToCart(cartId)
25       .subscribe((result)=>{
26         this.loadDetails();
27       })
28   }
29   loadDetails(){
30     this.product.currentCart().subscribe((result) => {
31       this.cartData = result;
32       console.warn(this.cartData);
33       let price = 0;
34       result.forEach((item) => {
35         if (item.quantity) {
36           price = price + (item.price * item.quantity)
```

```

        let price = 0;
        result.forEach((item) => {
            if (item.quantity) {
                price = price + (+item.price * +item.quantity)
            }
        })
        this.priceSummary.price = price;
        this.priceSummary.discount = price / 10;
        this.priceSummary.tax = price / 10;
        this.priceSummary.delivery = 100;
        this.priceSummary.total = price + (price / 10) + 100 - (price / 10);

        if(!this.cartData.length){
            this.router.navigate([''])
        }
    })
}

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

```

CHECKOUT(CSS):

```

at_Kitchen_story > src > app > checkout > # checkout.css
.checkout h1{
    text-align:center;
    margin-top: 50px;
    margin-bottom: 50px;
}

.checkout h3{
    text-align: center;
    margin-top: 15px;
    margin-bottom: 15px;
}

```

CHECKOUT(HTML):

```

<div class="checkout">
  <h1>Checkout</h1>
  <div class="row">
    <div class="col-sm-7">
      <h3>Add shipping address</h3>
      <p class="message-p">{{orderMsg}}</p>
      <form #orderData="ngForm" class="common-form" (ngSubmit)="orderNow(orderData.value)">
        <input
          type="text"
          class="form-input"
          name="email"
          placeholder="Enter User Email"
          ngModel
        />
        <input
          type="text"
          class="form-input"
          name="address"
          placeholder="Enter User Address"
          ngModel
        />
        <input
          type="text"
          class="form-input"
          name="contact"
          placeholder="Enter contact details"
          ngModel
        />
        <button class="form-button">Order Now</button>
      </form>
    </div>
    <div class="col-sm-5">
      <h3>Total Amount: {{totalPrice}}</h3>
      <h4>Payment Method : <b>Pay on delivery</b></h4>
    </div>
  </div>
</div>

```

CHECKOUT(TS):

```

1 import { Component, OnInit } from '@angular/core';
2 import { Router } from '@angular/router';
3 import { cart, order } from '../data-type';
4 import { ProductService } from '../services/product.service';
5
6 @Component({
7   selector: 'app-checkout',
8   templateUrl: './checkout.component.html',
9   styleUrls: ['./checkout.component.css']
10 })
11 export class CheckoutComponent implements OnInit {
12
13   totalPrice: number | undefined;
14   cartData: cart[] | undefined;
15   orderMsg: string | undefined;
16   constructor(private product: ProductService, private router: Router) { }
17
18   ngOnInit(): void {
19     this.product.currentCart().subscribe((result) => {
20
21       let price = 0;
22       this.cartData = result;
23       result.forEach((item) => {
24         if (item.quantity) {
25           price = price + (+item.price * +item.quantity)
26         }
27       })
28       this.totalPrice = price + (price / 10) + 100 - (price / 10);
29
30       console.warn(this.totalPrice);
31     })
32   }
33
34   }
35   orderNow(data: { email: string, address: string, contact: string }) {
36     let user = localStorage.getItem('user');
37     let userId = user && JSON.parse(user).id;
38     if (this.totalPrice) {
39       let orderData: order = {
40         ...data,

```

```

        console.warn(this.totalPrice);
    })
}
orderNow(data: { email: string, address: string, contact: string }) {
    let user = localStorage.getItem('user');
    let userId = user && JSON.parse(user).id;
    if (this.totalPrice) {
        let orderData: order = {
            ...data,
            totalPrice: this.totalPrice,
            userId,
            id: undefined
        }

        this.cartData?.forEach((item) => {
            setTimeout(() => {
                item.id && this.product.deleteCartItems(item.id);
            }, 700)
        })

        this.product.orderNow(orderData).subscribe((result) => {
            if (result) {
                this.orderMsg = "Order has been placed";
                setTimeout(() => {
                    this.orderMsg = undefined;
                    this.router.navigate(['/my-orders'])
                }, 4000);
            }
        })
    }
}
}

```

FOOTER(CSS):

judai_kitchen_story / src / app / footer / # footer.component

```
1  ul{
2      margin: 0;
3      padding: 0;
4  }
5  .footer {
6      padding: 10px;
7      text-align: center;
8      position: fixed;
9      bottom: 0;
10     right: 0;
11     left: 0;
12     background-color: #fff;
13 }
14 .footer ul{
15     list-style-type: none;
16     display: flex;
17     justify-content: space-between;
18 }
19 .footer ul li a{
20     color: blueviolet;
21     text-decoration: none;
22 }
```


FOOTER(HTML):

```
1 <div class="footer">
2   <ul>
3     <li>
4       <a href="">Become a Seller</a>
5     </li>
6     <li>
7       <a href="">Policies</a>
8     </li>
9     <li>
10      <a href="">Terms and Conditions</a>
11    </li>
12    <li>
13      <a href="">Help Center</a>
14    </li>
15    <li>
16      <a href="">News</a>
17    </li>
18    <li>
19      <a href="">Facebook</a>
20    </li>
21    <li>
22      <a href="">Youtube</a>
23    </li>
24    <li>
25      <a href="">Sitemap</a>
26    </li>
27    <li>
28      <a href="">FAQ</a>
29    </li>
30  </ul>
31  <p>© 2022 Kitchen Story Project</p>
32
33 </div>
```

FOOTER(TS):

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

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

  constructor() { }

  ngOnInit(): void {
  }

}
```

HEADER(CSS):

```
.nav {  
  display: flex;  
  justify-content: space-between;  
}  
  
.nav h1 {  
  color: blueviolet;  
  margin: 0;  
  font-size: 40px;  
}  
  
.nav .nav-search {  
  flex: 1;  
  display: flex;  
  margin-left: 5%;  
  margin-right: 5%;  
  margin-top: 5px;  
}  
  
.nav .nav-search input {  
  flex: 1;  
  color: blueviolet;  
  border: 1px solid blueviolet;  
  height: 39px;  
  border-right: none;  
}  
  
.nav .nav-search button{  
  color: blueviolet;  
  border: 1px solid blueviolet;  
  height: 39px;  
}  
  
.nav ul {  
  display: inline-flex;  
  margin: 0;  
  padding: 0;  
  list-style-type: none;  
}
```

```
.nav ul li{
  padding-left: 15px;
  padding-top: 15px;
}

.nav ul a {
  text-decoration: none;
  color: blueviolet;
  cursor: pointer;
}

h1 a{
  color: blueviolet;
  text-decoration: none;
}

.suggested-search{
  position: absolute;
  top: 40px;
  z-index: 10;
  display: list-item !important;
  border: 1px solid #ddd;
  background-color: #fff;
  width: 300px;
}

.suggested-search li {
  padding: 5px;
  border: 1px solid #ddd;
}

.suggested-search li a{
  cursor: pointer;
  color: blueviolet;
}
```

HEADER(HTML):

Go to component

```
<nav class="nav">
  <h1><a routerLink="">Kitchen Story</a> </h1>
  <div *ngIf="menuType!=='seller'" class="nav-search">
    <input type="text"
      #searchInput
      (keyup)="searchProduct($event)" (blur)="hideSearch()" placeholder="Search Products" />
    <button (click)="submitSearch(searchInput.value)" >Search</button>
    <ul class="suggested-search" *ngIf="searchResult">
      <li *ngFor="let item of searchResult">
        <a (mousedown)="redirectToDetails(item.id)" >{{item.name}}</a>
      </li>
    </ul>
  </div>
  <div [ngSwitch]="menuType">
    <ul *ngSwitchCase="'default'" >
      <li><a routerLink="seller-auth">Seller</a></li>
      <li><a routerLink="">Home</a></li>
      <li><a routerLink="user-auth">Login/Sign-up</a></li>
      <li><a href="#">Cart({{cartItems}})</a></li>
    </ul>
    <ul *ngSwitchCase="'user'" >
      <li><a routerLink="seller-auth">Seller</a></li>
      <li><a routerLink="">Home</a></li>
      <li><a>{{userName | titlecase}}</a></li>
      <li><a routerLink="my-orders" >My Orders</a></li>
      <li><a (click)="userLogout()" >Logout</a></li>
      <li><a [routerLink]="cartItems>0?'cart-page':null">Cart({{cartItems}})</a></li>
    </ul>
    <ul *ngSwitchCase="'seller'" >
      <li><a routerLink="seller-add-product">Add Product</a></li>
      <li><a routerLink="seller-home">List</a></li>
      <li><a (click)="logout()">Logout</a></li>
      <li><a href="#">{{sellerName | titlecase}}</a></li>
    </ul>
  </div>
</nav>
```

HEADER(TS):

```
import { Component, OnInit } from '@angular/core';
import { Router } from '@angular/router';
import { product } from '../data-type';
import { ProductService } from '../services/product.service';
@Component({
  selector: 'app-header',
  templateUrl: './header.component.html',
  styleUrls: ['./header.component.css'],
})
export class HeaderComponent implements OnInit {
  menuType: string = 'default';
  sellerName:string="";
  userName:string="";
  searchResult:undefined|product[];
  cartItems=0;
  constructor(private route: Router, private product:ProductService) {}

  ngOnInit(): void {
    this.route.events.subscribe((val: any) => {
      if (val.url) {
        if (localStorage.getItem('seller') && val.url.includes('seller')) {
          let sellerStore=localStorage.getItem('seller');
          let sellerData =sellerStore && JSON.parse(sellerStore)[0];
          this.sellerName=sellerData.name;
          this.menuType = 'seller';
        }
        else if(localStorage.getItem('user')){
          let userStore = localStorage.getItem('user');
          let userData = userStore && JSON.parse(userStore);
          this.userName= userData.name;
          this.menuType='user';
          this.product.getCartList(userData.id);
        }
        else {
          this.menuType = 'default';
        }
      }
    });
    let cartData= localStorage.getItem('localCart');
```

```

    |   this.cartItems= JSON.parse(cartData).length
    | }
    | this.product.cartData.subscribe((items)=>{
    |   this.cartItems= items.length
    | })
  }
  logout(){
    |   localStorage.removeItem('seller');
    |   this.route.navigate(['/'])
  }

  userLogout(){
    |   localStorage.removeItem('user');
    |   this.route.navigate(['/user-auth'])
    |   this.product.cartData.emit([])
  }

  searchProduct(query:KeyboardEvent){
    |   if(query){
    |     const element = query.target as HTMLInputElement;
    |     this.product.searchProduct(element.value).subscribe((result)=>{
    |       |   if(result.length>5){
    |       |   result.length=length
    |       |   }
    |       |   this.searchResult=result;
    |     })
    |   }
  }
  hideSearch(){
    |   this.searchResult=undefined
  }
  redirectToDetails(id:number){
    |   this.route.navigate(['/details/'+id])
  }
  submitSearch(val:string){
    |   console.warn(val)
    |   this.route.navigate(['search/${val}']);
  }
}

```

HOME(CSS):

```
.product-header-image{
  object-fit: contain;
  width: 100%;
  height: 600px;
}

.product-item{
  margin-bottom: 40px;
  border: 1px solid #ccc;
}

.product-item img{
  width: 100%;
  height: 300px;
  object-fit: contain;
}

.product-item .product-details {
  border-top: 1px solid #ccc;
  border-bottom: 1px solid #ccc;
  padding: 15px;
}

.product-item .name-head{
  font-size: 20px;
  font-weight: 600;
}

.product-item h6{
  text-align: center;
}

.product-item .product-footer{
  display: flex;
  justify-content: space-between;
  padding: 10px;
}
```

HOME(HTML):

```
<ngb-carousel *ngIf="popularProducts">
  <ng-template *ngFor="let item of popularProducts" ngbSlide>
    <div class="picsum-img-wrapper">
      <a routerLink="/details/{{item.id}}"> 
    </a>
    </div>
    <div class="carousel-caption">
      <h3>{{item.name}}</h3>
      <p>{{item.description}}</p>
    </div>
  </ng-template>
</ngb-carousel>
<div class="product-list-wrapper">
  <h1>Trendy Products</h1>
  <div class="row">
    <div class="col-sm-3" *ngFor="let item of trendyProducts">
      <div class="product-item">
        
        <div class="product-details">
          <h6 class="name-head">{{item.name}}</h6>
          <div class="product-price">
            <h6>Price:{{item.price}}</h6>
          </div>
          <!-- <div class="product-color">
            <h6>Color:{{item.color}}</h6>
          </div-->
        </div>
        <div class="product-footer">
          <a routerLink="/details/{{item.id}}">View Details</a>
          <a href="">Add to Cart</a>
        </div>
      </div>
    </div>
  </div>
</div>
</div>
```


HOME(TS):

```
1 import { Component, OnInit } from '@angular/core';
2 import { product } from '../data-type';
3 import { ProductService } from '../services/product.service';
4
5 @Component({
6   selector: 'app-home',
7   templateUrl: './home.component.html',
8   styleUrls: ['./home.component.css'],
9 })
10 export class HomeComponent implements OnInit {
11   popularProducts: undefined | product[];
12   trendyProducts: undefined | product[];
13   constructor(private product: ProductService) {}
14
15   ngOnInit(): void {
16     this.product.popularProducts().subscribe((data) => {
17       this.popularProducts = data;
18     });
19
20     this.product.trendyProducts().subscribe((data) => {
21       this.trendyProducts = data;
22     });
23   }
24 }
25
```

MY_ORDERS(CSS):

```

.my-orders{
    margin-bottom: 100px;
}

.my-orders h1{
    margin-top: 50px;
    margin-bottom: 50px;
    text-align: center;
}

.my-orders .order-img{
    width: 100px;
}

.my-orders .row{
    margin-top: 15px;
    margin-bottom: 15px;
    padding-bottom: 10px;
    border-bottom: 1px solid #ddd;
}

```

MY_ORDERS(HTML):

```

<div class="my-orders">
  <h1>My Orders</h1>
  <div class="row">
    <div class="col-sm-2">
      <span>Orders</span>
    </div>
    <div class="col-sm-2">
      <span>Order Id</span>
    </div>
    <div class="col-sm-2">
      <span>Price</span>
    </div>
    <div class="col-sm-2">
      <span>Status</span>
    </div>
    <div class="col-sm-2">
      <span>Cancel Order</span>
    </div>
  </div>
  <div class="row" *ngFor="let order of orderData">
    <div class="col-sm-2">
      
    </div>
    <div class="col-sm-2">
      <span>#{order.id}</span>
    </div>
    <div class="col-sm-2">
      <span>{{order.totalPrice}}</span>
    </div>
    <div class="col-sm-2">
      <span>In Progress</span>
    </div>
    <div class="col-sm-2">
      <button class="form-button" (click)="cancelOrder(order.id)">Cancel Order</button>
    </div>
  </div>
</div>

```

MYORDERS(TS):

```

import { Component, OnInit } from '@angular/core';
import { order } from '../data-type';
import { ProductService } from '../services/product.service';

@Component({
  selector: 'app-my-orders',
  templateUrl: './my-orders.component.html',
  styleUrls: ['./my-orders.component.css']
})
export class MyOrdersComponent implements OnInit {

  orderData:order[]|undefined
  constructor(private product:ProductService) { }

  ngOnInit(): void {
    this.getOrderList()
  }
  cancelOrder(orderId:number|undefined){
    orderId && this.product.cancelOrder(orderId).subscribe((result)=>{
      if(result){
        this.getOrderList();
      }
    })
  }
  getOrderList(){
    this.product.orderList().subscribe((result)=>{
      this.orderData=result;
    })
  }
}

```

PRODUCTDETAILS(CSS):

```

.product-details{
  margin-top: 50px;
  margin-bottom: 80px;
}
.product-details .product-img{
  width: 100%;
  height: 400px;
  object-fit: contain;
}
.product-color{
  border: 3px solid ■ #000;
  width: 40px;
  height: 40px;
  border-radius: 50%;
}
.quantity-group{
  margin-top: 20px;
}
.quantity-group button{
  padding: 10px;
}
.quantity-group input{
  padding: 10px;
  width: 50px;
}

```

PRODUCTDETAILS(HTML):

dr_kitchen_story / src / app / product-details / > product-details.component.html / > div.row.product-details

Go to component

```
<div class="row product-details" *ngIf="productData">
  <div class="col-sm-6">
    
  </div>
  <div class="col-sm-6">
    <div class="details">
      <h1>{{productData.name}}</h1>
      <h3>Price : {{productData.price}}</h3>
      <!-- <div>Color :<h3 [style.backgroundColor]=productData.color" class="product-color"></h3></div>-->
      <h6>Category : {{productData.category}}</h6>
      <h6>Description : {{productData.description}}</h6>
      <button class="form-button">Buy Now</button>
      <button *ngIf="removeCart" class="form-button" (click)=" removeToCart(productData.id)">Remove to Cart</button>
      <button *ngIf=!removeCart" class="form-button" (click)="addToCart()">Add to Cart</button>
      <div class="quantity-group">
        <button (click)="handleQuantity('min') ">-</button>
        <input type="text" value="{{productQuantity}}" />
        <button (click)="handleQuantity('plus') ">+</button>
      </div>
    </div>
  </div>
</div>
</div>
```

PRODUCTDETAILS(TS):

id_Nutrien_story > src > app > product-details > product-details.component.ts > ...

```
import { Component, OnInit } from '@angular/core';
import { ActivatedRoute } from '@angular/router';
import { cart, product } from '../data-type';
import { ProductService } from '../services/product.service';

@Component({
  selector: 'app-product-details',
  templateUrl: './product-details.component.html',
  styleUrls: ['./product-details.component.css']
})
export class ProductDetailsComponent implements OnInit {
  productData: undefined | product;
  productQuantity: number = 1;
  removeCart = false;
  cartData: product | undefined;
  constructor(private activeRoute: ActivatedRoute, private product: ProductService) { }

  ngOnInit(): void {
    let productId = this.activeRoute.snapshot.paramMap.get('productId');
    console.warn(productId);
    productId && this.product.getProduct(productId).subscribe((result) => {
      this.productData = result;
      let cartData = localStorage.getItem('localCart');
      if (productId && cartData) {
        let items = JSON.parse(cartData);
        items = items.filter((item: product) => productId === item.id.toString());
        if (items.length) {
          this.removeCart = true;
        } else {
          this.removeCart = false;
        }
      }
    });

    let user = localStorage.getItem('user');
    if (user) {
      let userId = user && JSON.parse(user).id;
      this.product.getCartList(userId);

      this.product.cartData.subscribe((result) => {
        let item = result.filter((item: product) => productId?.toString() === item.productId?.toString())
      });
    }
  }
}
```

```

        if(item.length){
            this.cartData=item[0];
            this.removeCart=true;
        }
    })
}

}))

}

handleQuantity(val:string){
    if(this.productQuantity<20 && val==='plus'){
        this.productQuantity+=1;
    }else if(this.productQuantity>1 && val==='min'){
        this.productQuantity-=1;
    }
}

addToCart(){
    if(this.productData){
        this.productData.quantity = this.productQuantity;
        if(!localStorage.getItem('user')){
            this.product.localAddToCart(this.productData);
            this.removeCart=true
        }else{
            let user = localStorage.getItem('user');
            let userId= user && JSON.parse(user).id;
            let cartData:cart={
                ...this.productData,
                productId:this.productData.id,
                userId
            }
            delete cartData.id;
            this.product.addToCart(cartData).subscribe((result)=>{
                if(result){
                    this.product.getCartList(userId);
                    this.removeCart=true
                }
            })
        }
    }
}

```

```

        // console.log
        this.product.getCartList(userId);
        this.removeCart=true
    }
    })
}
}
}
removeToCart(productId:number){
    if(!localStorage.getItem('user')){
this.product.removeItemFromCart(productId)
    }else{
        console.warn("cartData", this.cartData);

        this.cartData && this.product.removeToCart(this.cartData.id)
        .subscribe((result)=>{
            let user = localStorage.getItem('user');
            let userId= user && JSON.parse(user).id;
            this.product.getCartList(userId)
        })
    }
    this.removeCart=false
}
}
}

```

SEARCH(CSS):

```

.search-item{
    width: 800px;
    margin: 25px auto;
    border-top: 1px solid #ddd;
    padding-top: 15px;
}
.search-item img{
    width: 100%;
    object-fit: contain;
    height: 150px;
}
.search-item h6{
    color: blueviolet;
    font-size: 24px;
}
.search-item p{
    margin-bottom: 5px;
}
.search-item span{margin-right: 10px;}

```

SEARCH(HTML):

Go to component

```
<div class="row search-item" *ngFor="let item of searchResult">
  <div class="col-sm-3">
    <a routerLink="/details/{{item.id}}"> </a>
  </div>
  <div class="col-sm-8 details">
    <h6> <a routerLink="/details/{{item.id}}"> {{item.name}} </a></h6>
    <p><b>Price:{{item.price}}</b></p>
    <!-- <p><span>Color:{{item.color}}</span><span>Category: {{item.category}}</span></p>-->
    <p>{{item.description}}</p>
  </div>
</div>
```

SEARCH(TS):

```
import { Component, OnInit } from '@angular/core';
import { ActivatedRoute } from '@angular/router';
import { product } from '../data-type';
import { ProductService } from '../services/product.service';
@Component({
  selector: 'app-search',
  templateUrl: './search.component.html',
  styleUrls: ['./search.component.css']
})
export class SearchComponent implements OnInit {

  searchResult:undefined|product[]
  constructor(private activeRoute: ActivatedRoute, private product:ProductService) { }

  ngOnInit(): void {
    let query = this.activeRoute.snapshot.paramMap.get('query');
    console.warn(query);
    query && this.product.searchProduct(query).subscribe((result)=>{
      this.searchResult=result;
    })
  }

}
```

SELLER-ADD-PRODUCT(CSS):

```
h1{
  color: blueviolet;
  text-align: center;
}
.add-product{
  text-align: center;
}
```


SELLER-ADD-PRODUCT(HTML):

Go to component

```
<div class="add-product">
  <h1>Add New Product</h1>
  <p class="message-p">{{addProductMessage}}</p>
  <form #addProduct="ngForm"
    (ngSubmit)="submit(addProduct.value)"
    class="common-form">
    <input
      type="text"
      class="form-input"
      placeholder="Enter Product Name"
      name="name"
      ngModel
    />
    <input
      type="text"
      class="form-input"
      placeholder="Enter Product Price"
      name="price"
      ngModel
    />
    <input
      type="text"
      class="form-input"
      placeholder="Enter Product Category"
      name="category"
      ngModel
    />
    <!--<input
      type="text"
      class="form-input"
      placeholder="Enter Product Color"
      name="color"
      ngModel
    />-->
    <textarea
      type="text"
      class="form-input"
      placeholder="Enter Product Description"
      name="description"
    />
    <!--<textarea
      type="text"
      class="form-input"
      placeholder="Enter Product Description"
      name="description"
    />-->
    <input
      type="text"
      class="form-input"
      placeholder="Enter Product Image URL"
      name="image"
      ngModel
    />
    <button class="form-button">Add Product</button>
  </form>
</div>
```

SELLER-ADD-PRODUCT(TS):

```
angular_kitchen_story > src > app > seller-add-product > ts seller-add-product.component.ts > ...  
import { Component, OnInit } from '@angular/core';  
import { product } from '../data-type';  
import { ProductService } from '../services/product.service';  
  
@Component({  
  selector: 'app-seller-add-product',  
  templateUrl: './seller-add-product.component.html',  
  styleUrls: ['./seller-add-product.component.css'],  
})  
export class SellerAddProductComponent implements OnInit {  
  addProductMessage: string | undefined;  
  constructor(private product: ProductService) {}  
  
  ngOnInit(): void {}  
  
  submit(data: product) {  
    this.product.addProduct(data).subscribe((result) => {  
      console.warn(result);  
      if (result) {  
        this.addProductMessage = 'Product is added successfully';  
      }  
    });  
  
    setTimeout(() => {  
      this.addProductMessage=undefined  
    }, 3000);  
  }  
}
```

SELLER-AUTH(CSS):

```
.sign-up, .login{  
  text-align: center;  
}  
.sign-up h1, .login h1{  
  margin-top: 50px;  
  margin-bottom: 80px;  
  color: blueviolet;  
}  
.auth-toggle-link{  
  color: blueviolet;  
  font-size: 14px;  
  margin-top: 20px;  
  text-align: left;  
}
```

SELLER-AUTH(HTML):

```
<div class="seller-auth">
  <div *ngIf="!showLogin" class="sign-up">
    <h1>Seller sign-up</h1>
    <form #sellerSignUp="ngForm" (ngSubmit)="signUp(sellerSignUp.value)" class="common-form">
      <input
        type="text"
        class="form-input"
        placeholder="Enter Name"
        name="name"
        ngModel
      />
      <input
        type="password"
        class="form-input"
        placeholder="Enter Password"
        name="password"
        ngModel
      />
      <input
        type="text"
        class="form-input"
        placeholder="Enter Email"
        name="email"
        ngModel
      />
      <button class="form-button">Sign Up</button>
      <a class="auth-toggle-link" (click)="openLogin()">Already have account? Click for Login</a>
    </form>
  </div>

  <div *ngIf="showLogin" class="login">
    <h1>Seller Login</h1>
    <p class="error-p">{{authError}}</p>
    <form #sellerLogin="ngForm" (ngSubmit)="login(sellerLogin.value)" class="common-form">
      <input
        type="text"
        class="form-input"
        placeholder="Enter Email"
        name="email"
        ngModel
      />
      <input
        type="password"
        class="form-input"
        placeholder="Enter Password"
        name="password"
        ngModel
      />
      <button class="form-button">Login</button>
      <a class="auth-toggle-link" (click)="openSignUp()">Don't have account? Click for sign-up</a>
    </form>
  </div>
</div>

./div>
```

SELLER-AUTH(TS):

```
ng_angular_angular / src / app / seller-auth / ts / seller-auth.component.ts / ...
import { Component, OnInit } from '@angular/core';
import { signUp } from '../data-type';
import { SellerService } from '../services/seller.service';

@Component({
  selector: 'app-seller-auth',
  templateUrl: './seller-auth.component.html',
  styleUrls: ['./seller-auth.component.css'],
})
export class SellerAuthComponent implements OnInit {
  showLogin=false;
  authError:String='';
  constructor(private seller: SellerService) {}

  ngOnInit(): void {
    this.seller.reloadSeller()
  }
  signUp(data: signUp): void {
    console.warn(data);
    this.seller.userSignUp(data);
  }
  login(data: signUp): void {
    this.seller.userLogin(data);
    this.seller.isLoginError.subscribe((isError)=>{
      if(isError){
        this.authError="Email or password is not correct";
      }
    })
  }
  openLogin(){
    this.showLogin=true
  }
  openSignUp(){
    this.showLogin=false
  }
}
```

SELLER-HOME(CSS):

```

table{
  text-align: left;
  color: blueviolet;
  border-collapse: collapse;
}
img{
  height: 100px;
  width: 100px;
}
table td , table th{
  border: 1px solid blueviolet;
  padding: 10px;
  min-width: 160pxs;
}
.adjust-update-link{
  margin-left: 10px;
}

```

SELLER-HOME(HTML):

```

<div class="product-list">
  <h1>Product List</h1>
  <p class="message-p">{{ productMessage }}</p>
  <table>
    <thead>
      <tr>
        <th>Image</th>
        <th>Name</th>
        <th>Price</th>
        <!-- <th>Color</th>-->
        <th>Category</th>
        <th>Description</th>
        <th>Action</th>
      </tr>
    </thead>
    <tbody>
      <tr *ngFor="let item of productList">
        <td></td>
        <td>{{ item.name }}</td>
        <td>{{ item.price }}</td>
        <!-- <td>{{ item.color }}</td>-->
        <td>{{ item.category }}</td>
        <td>{{ item.description }}</td>
        <td>
          <fa-icon (click)="deleteProduct(item.id)" [icon]="icon"></fa-icon>
          <a class="adjust-update-link" routerLink="/seller-update-product/{{item.id}}"> <fa-icon [icon]="iconEdit"></fa-icon></a>
        </td>
      </tr>
    </tbody>
  </table>
</div>

```

SELLER-HOME(TS):

```

import { Component, OnInit } from '@angular/core';
import { product } from '../data-type';
import { ProductService } from '../services/product.service';
import { faTrash, faEdit } from '@fortawesome/free-solid-svg-icons';
@Component({
  selector: 'app-seller-home',
  templateUrl: './seller-home.component.html',
  styleUrls: ['./seller-home.component.css'],
})
export class SellerHomeComponent implements OnInit {
  productList: undefined | product[];
  productMessage: undefined | string;
  icon = faTrash;
  iconEdit=faEdit;
  constructor(private product: ProductService) {}

  ngOnInit(): void {
    this.list();
  }

  deleteProduct(id: number) {
    this.product.deleteProduct(id).subscribe((result) => {
      if (result) {
        this.productMessage = 'Product is deleted';
        this.list();
      }
    });
    setTimeout(() => {
      this.productMessage = undefined;
    }, 3000);
  }

  list() {
    this.product.productList().subscribe((result) => {
      if (result) {
        this.productlist = result;
      }
    });
  }
}

```

SELLER-UPDATE-PRODUCT(CSS):

```

.update-product{
  text-align: center;
}
h1{
  color: blueviolet;
}

```

SELLER-UPDATE-PRODUCT(HTML):

Go to component

```
<div class="update-product">
  <h1>Update Product</h1>
  <p class="message-p">{{productMessage}}</p>
  <form #addProduct="ngForm"
    (ngSubmit)="submit(addProduct.value)"
    class="common-form">
    <input
      type="text"
      class="form-input"
      placeholder="Enter Product Name"
      name="name"
      ngModel="{{productData?.name}}"
    />
    <input
      type="text"
      class="form-input"
      placeholder="Enter Product Price"
      name="price"
      ngModel="{{productData?.price}}"
    />
    <input
      type="text"
      class="form-input"
      placeholder="Enter Product Category"
      name="category"
      ngModel="{{productData?.category}}"
    />
    <!--<input
      type="text"
      class="form-input"
      placeholder="Enter Product Color"
      name="color"
      ngModel="{{productData?.color}}"
    />-->
    <textarea
      type="text"
      class="form-input"
      placeholder="Enter Product Description"
      name="description"
```

```

<textarea
type="text"
class="form-input"
placeholder="Enter Product Description"
name="description"
ngModel="{{productData?.description}}"
></textarea>

<input
type="text"
class="form-input"
placeholder="Enter Product Image URL"
name="image"
ngModel="{{productData?.image}}"
/>
<button class="form-button">Update Product</button>

</form>
</div>

```

SELLER-UPDATE-PRODUCT(TS):

```

import { Component, OnInit } from '@angular/core';
import { ActivatedRoute } from '@angular/router';
import { product } from '../data-type';
import { ProductService } from '../services/product.service';

@Component({
  selector: 'app-seller-update-product',
  templateUrl: './seller-update-product.component.html',
  styleUrls: ['./seller-update-product.component.css'],
})
export class SellerUpdateProductComponent implements OnInit {
  productData: undefined | product;
  productMessage: undefined | string;
  constructor(private route: ActivatedRoute, private product: ProductService) {}

  ngOnInit(): void {
    let productId = this.route.snapshot.paramMap.get('id');
    console.warn(productId);
    productId &&
      this.product.getProduct(productId).subscribe((data) => {
        console.warn(data);
        this.productData = data;
      });
  }

  submit(data: any) {
    if (this.productData) {
      data.id = this.productData.id;
    }
    this.product.updateProduct(data).subscribe((result) => {
      if (result) {
        this.productMessage = 'Product has updated';
      }
    });
    setTimeout(() => {
      this.productMessage = undefined;
    }, 3000);
    console.warn(data);
  }
}

```


PRODUCTSERVICE(TS):

```
    });
  }
  popularProducts() {
    return this.http.get<product[]>(<'http://localhost:3000/products?_limit=3'>);
  }

  trendyProducts() {
    return this.http.get<product[]>(<'http://localhost:3000/products?_limit=8'>);
  }

  searchProduct(query: string) {
    return this.http.get<product[]>(<'http://localhost:3000/products?q=${query}'>);
  }

  localAddToCart(data: product) {
    let cartData = [];
    let localCart = localStorage.getItem('localCart');
    if (!localCart) {
      localStorage.setItem('localCart', JSON.stringify([data]));
      this.cartData.emit([data]);
    } else {
      cartData = JSON.parse(localCart);
      cartData.push(data);
      localStorage.setItem('localCart', JSON.stringify(cartData));
      this.cartData.emit(cartData);
    }
  }

  removeItemFromCart(productId: number) {
    let cartData = localStorage.getItem('localCart');
    if (cartData) {
      let items: product[] = JSON.parse(cartData);
      items = items.filter((item: product) => productId !== item.id);
      localStorage.setItem('localCart', JSON.stringify(items));
      this.cartData.emit(items);
    }
  }
}
```

```
addtoCart(cartData: cart) {
  return this.http.post(<'http://localhost:3000/cart'>, cartData);
}

getCartList(userId: number) {
  return this.http
    .get<product[]>(<'http://localhost:3000/cart?userId=' + userId, {
      observe: 'response',
    }>))
    .subscribe((result) => {
      if (result && result.body) {
        this.cartData.emit(result.body);
      }
    });
}

removeToCart(cartId: number) {
  return this.http.delete(<'http://localhost:3000/cart/' + cartId>);
}

currentCart() {
  let userStore = localStorage.getItem('user');
  let userData = userStore && JSON.parse(userStore);
  return this.http.get<cart[]>(<'http://localhost:3000/cart?userId=' + userData.id>);
}

orderNow(data: order) {
  return this.http.post(<'http://localhost:3000/orders'>, data);
}

orderList() {
  let userStore = localStorage.getItem('user');
  let userData = userStore && JSON.parse(userStore);
  return this.http.get<order[]>(<'http://localhost:3000/orders?userId=' + userData.id>);
}

deleteCartItems(cartId: number) {
  return this.http.delete(<'http://localhost:3000/cart/' + cartId>).subscribe((result) => {
    this.cartData.emit([]);
  });
}

cancelOrder(orderId: number) {
  return this.http.delete(<'http://localhost:3000/orders/' + orderId>);
}
```

SELLER SERVICE(TS):

```
import { EventEmitter, Injectable } from '@angular/core';
import { HttpClient } from '@angular/common/http'
import { login, signUp } from '../data-type';
import { BehaviorSubject } from 'rxjs';
import { Router } from '@angular/router';
@Injectable({
  providedIn: 'root'
})
export class SellerService {
  isSellerLoggedIn= new BehaviorSubject<boolean>(false);
  isLoginError= new EventEmitter<boolean>(false)

  constructor(private http:HttpClient, private router:Router) { }
  userSignUp(data:signUp){
    this.http.post('http://localhost:3000/seller',
      data,
      {observe:'response'}).subscribe((result)=>{
        console.warn(result)
        if(result){
          localStorage.setItem('seller',JSON.stringify(result.body))
          this.router.navigate(['seller-home'])
        }
      })
  }
  reloadSeller(){
    if(localStorage.getItem('seller')){
      this.isSellerLoggedIn.next(true)
      this.router.navigate(['seller-home'])
    }
  }
  userLogin(data:login){
    this.http.get(`http://localhost:3000/seller?email=${data.email}&password=${data.password}`,
      {observe:'response'}).subscribe((result:any)=>{
        console.warn(result)
        if(result && result.body && result.body.length===1){
          this.isLoginError.emit(false)
          localStorage.setItem('seller',JSON.stringify(result.body))
          this.router.navigate(['seller-home'])
        }else{
          console.warn("login failed");
        }
      })
  }
}
```

```

    {observe: 'response'}).subscribe((result)=>{
      console.warn(result)
      if(result){
        localStorage.setItem('seller', JSON.stringify(result.body))
        this.router.navigate(['seller-home'])
      }
    })
  }
  reloadSeller(){
    if(localStorage.getItem('seller')){
      this.isSellerLoggedIn.next(true)
      this.router.navigate(['seller-home'])
    }
  }
  userLogin(data:login){
    this.http.get(`http://localhost:3000/seller?email=${data.email}&password=${data.password}`,
    {observe: 'response'}).subscribe((result:any)=>{
      console.warn(result)
      if(result && result.body && result.body.length===1){
        this.isLoginError.emit(false)
        localStorage.setItem('seller', JSON.stringify(result.body))
        this.router.navigate(['seller-home'])
      }else{
        console.warn("login failed");
        this.isLoginError.emit(true)
      }
    })
  }
}

```

USER SERVICE(TS):

```

import { HttpClient } from '@angular/common/http';
import { EventEmitter, Injectable } from '@angular/core';
import { Router } from '@angular/router';
import { login, signUp } from '../data-type';

@Injectable({
  providedIn: 'root'
})
export class UserService {
  invalidUserAuth= new EventEmitter<boolean>(false)
  constructor(private http: HttpClient, private router:Router) { }
  userSignUp(user:signUp){
    this.http.post('http://localhost:3000/users',user,{observe: 'response'})
    .subscribe((result)=>{
      if(result){
        localStorage.setItem('user', JSON.stringify(result.body));
        this.router.navigate(['/']);
      }
    })
  }
  userLogin(data:login){
    this.http.get<signUp[]>(`http://localhost:3000/users?email=${data.email}&password=${data.password}`,
    {observe: 'response'})
    .subscribe((result)=>{
      if(result && result.body?.length){
        localStorage.setItem('user', JSON.stringify(result.body[0]));
        this.router.navigate(['/']);
        this.invalidUserAuth.emit(false)
      }else{
        this.invalidUserAuth.emit(true)
      }
    })
  }
  userAuthReload(){
    if(localStorage.getItem('user')){
      this.router.navigate(['/']);
    }
  }
}

```

USER-AUTH(CSS):

```
.user-auth h1{
  margin-top: 50px;
  margin-bottom: 80px;
  color: blueviolet;
  text-align: center;
}

.user-auth {
  margin-bottom: 100px;
}

.user-auth a{
  color: blueviolet;
  cursor: pointer;
}
```

USER-AUTH(HTML):

```
<div class="user-auth">
  <div *ngIf="!showLogin" class="sign-up">
    <h1>User Signup</h1>
    <form #userSignUp="ngForm" class="common-form" (ngSubmit)="signUp(userSignUp.value)">
      <input
        type="text"
        class="form-input"
        name="name"
        placeholder="Enter User Name"
        ngModel
      />
      <input
        type="text"
        class="form-input"
        name="email"
        placeholder="Enter User Email"
        ngModel
      />
      <input
        type="password"
        class="form-input"
        name="password"
        placeholder="Enter User Password"
        ngModel
      />
      <button class="form-button">Sign Up</button>
      <p>Already Have Account ? <a (click)="openLogin()" >Click Here</a></p>
    </form>
  </div>
  <div *ngIf="showLogin" class="login">
    <h1>User Login</h1>
    <form #userLogin="ngForm" class="common-form" (ngSubmit)="login(userLogin.value)">
      <p class="error-p">{{authError}}</p>
      <input
        type="text"
        class="form-input"
        name="email"
        placeholder="Enter User Email"
        ngModel
        email
      />
    </form>
  </div>
</div>
```

```

52 <p class="error-p">{{authError}}</p>
53 <input
54   type="text"
55   class="form-input"
56   name="email"
57   placeholder="Enter User Email"
58   ngModel
59   email
60   #email="ngModel"
61   required
62 />
63 <p class="invalid-input"><span *ngIf="email.invalid && email.touched " >Please enter valid input</span></p>
64 <input
65   type="password"
66   class="form-input"
67   name="password"
68   placeholder="Enter User Password"
69   ngModel
70   minlength="2"
71   maxlength="20"
72   #password="ngModel"
73   required
74 />
75 <p class="invalid-input"><span *ngIf="password.invalid && password.touched " >Please enter valid input</span></p>
76 <button [disabled]="userLogin.invalid" class="form-button">Login</button>
77 <p>Want to Create Account ? <a (click)="openSignUp()" >Click Here</a></p>
78 </form>
79 </div>
80 </div>
81

```



USER-AUTH(TS):

```

import { Component, OnInit } from '@angular/core';
import { cart, login, product, signUp } from '../data-type';
import { ProductService } from '../services/product.service';
import { UserService } from '../services/user.service';

@Component({
  selector: 'app-user-auth',
  templateUrl: './user-auth.component.html',
  styleUrls: ['./user-auth.component.css'],
})
export class UserAuthComponent implements OnInit {
  showLogin: boolean = true;
  authError: string = "";
  constructor(private user: UserService, private product: ProductService) {}

  ngOnInit(): void {
    this.user.userAuthReload();
  }

  signUp(data: signUp) {
    this.user.userSignUp(data);
  }

  login(data: login) {
    this.user.userLogin(data)
    this.user.invalidUserAuth.subscribe((result) => {
      console.warn(result);
      if (result) {
        this.authError = "User not found"
      } else {
        this.localCartToRemoteCart();
      }
    })
  }

  openSignUp() {
    this.showLogin = false;
  }

  openLogin() {
    this.showLogin = true;
  }
}

```

```

openLogin(){
this.showLogin=true;
}

localCartToRemoteCart(){
let data = localStorage.getItem('localCart');
let user = localStorage.getItem('user');
let userId= user && JSON.parse(user).id;
if(data){
let cartDataList:product[]= JSON.parse(data);

cartDataList.forEach((product:product, index)=>{
let cartData:cart={
...product,
productId:product.id,
userId
}
delete cartData.id;
setTimeout(() => {
this.product.addToCart(cartData).subscribe((result)=>{
if(result){
console.warn("data is stored in DB");
}
}
}), 500);
if(cartDataList.length===index+1){
localStorage.removeItem('localCart')
}
})
}

setTimeout(() => {
this.product.getCartList(userId)
}, 2000);
}
}

```

APP-ROUTING-MODULE.TS

```

1 import { NgModule } from '@angular/core';
2 import { RouterModule, Routes } from '@angular/router';
3 import { AuthGuard } from './auth.guard';
4 import { CartPageComponent } from './cart-page/cart-page.component';
5 import { CheckoutComponent } from './checkout/checkout.component';
6 import { HomeComponent } from './home/home.component';
7 import { MyOrdersComponent } from './my-orders/my-orders.component';
8 import { ProductDetailsComponent } from './product-details/product-details.component';
9 import { SearchComponent } from './search/search.component';
10 import { SellerAddProductComponent } from './seller-add-product/seller-add-product.component';
11 import { SellerAuthComponent } from './seller-auth/seller-auth.component';
12 import { SellerHomeComponent } from './seller-home/seller-home.component';
13 import { SellerUpdateProductComponent } from './seller-update-product/seller-update-product.component';
14 import { UserAuthComponent } from './user-auth/user-auth.component';
15
16 const routes: Routes = [
17   {
18     component: HomeComponent,
19     path: '',
20   },
21   {
22     component: SellerAuthComponent,
23     path: 'seller-auth',
24   },
25   {
26     component: SellerHomeComponent,
27     path: 'seller-home',
28     canActivate:[AuthGuard]
29   },
30   {
31     component: SellerAddProductComponent,
32     path: 'seller-add-product',
33     canActivate:[AuthGuard]
34   },
35 ]

```

```

    ], {
      component: SellerUpdateProductComponent,
      path: 'seller-update-product/:id',
      canActivate: [AuthGuard]
    },
    {
      component: SearchComponent,
      path: 'search/:query'
    }, {
      component: ProductDetailsComponent,
      path: 'details/:productId'
    }, {
      component: UserAuthComponent,
      path: 'user-auth'
    }, {
      component: CartPageComponent,
      path: 'cart-page'
    }, {
      component: CheckoutComponent,
      path: 'checkout'
    }, {
      component: MyOrdersComponent,
      path: 'my-orders'
    }
  ];

  @NgModule({
    imports: [RouterModule.forRoot(routes)],
    exports: [RouterModule],
  })
  export class AppRoutingModule {}

```

APP-COMPONENT.HTML:

```

<div class="container">
  <app-header></app-header>
  <router-outlet></router-outlet>
  <app-footer></app-footer>
</div>

```

APP-COMPONENT.TS:

```
run_kitchen_story / src / app / ts appcomponent.html / ts AppComponent.ts
import { Component } from '@angular/core';

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

APP-MODULE.TS


```
import { HttpClientModule } from '@angular/common/http';
import { NgModule } from '@angular/core';
import { FormsModule } from '@angular/forms';
import { BrowserModule } from '@angular/platform-browser';

import { AppRoutingModule } from './app-routing.module';
import { AppComponent } from './app.component';
import { HeaderComponent } from './header/header.component';
import { HomeComponent } from './home/home.component';
import { SellerAuthComponent } from './seller-auth/seller-auth.component';
import { SellerHomeComponent } from './seller-home/seller-home.component';
import { SellerAddProductComponent } from './seller-add-product/seller-add-product.component';
import { FontAwesomeModule } from '@fortawesome/angular-fontawesome';
import { SellerUpdateProductComponent } from './seller-update-product/seller-update-product.comp
import { NgbModule } from '@ng-bootstrap/ng-bootstrap';
import { FooterComponent } from './footer/footer.component';
import { SearchComponent } from './search/search.component';
import { ProductDetailsComponent } from './product-details/product-details.component';
import { UserAuthComponent } from './user-auth/user-auth.component';
import { CartPageComponent } from './cart-page/cart-page.component';
import { CheckoutComponent } from './checkout/checkout.component';
import { MyOrdersComponent } from './my-orders/my-orders.component';

@NgModule({
  declarations: [
    AppComponent,
    HeaderComponent,
    HomeComponent,
    SellerAuthComponent,
    SellerHomeComponent,
    SellerAddProductComponent,
    SellerUpdateProductComponent,
    FooterComponent,
    SearchComponent,
    ProductDetailsComponent,
    UserAuthComponent,
    CartPageComponent,
    CheckoutComponent,
    MyOrdersComponent
  ],
```

```

        HeaderComponent,
        HomeComponent,
        SellerAuthComponent,
        SellerHomeComponent,
        SellerAddProductComponent,
        SellerUpdateProductComponent,
        FooterComponent,
        SearchComponent,
        ProductDetailsComponent,
        UserAuthComponent,
        CartPageComponent,
        CheckoutComponent,
        MyOrdersComponent
    ],
    imports: [
        BrowserModule,
        AppRoutingModule,
        FormsModule,
        HttpClientModule,
        FontAwesomeModule,
        NgbModule
    ],
    providers: [],
    bootstrap: [AppComponent]
  })
  export class AppModule { }

```

AUTHGUARD.TS:

```

1  import { Injectable } from '@angular/core';
2  import { ActivatedRouteSnapshot, CanActivate, RouterStateSnapshot, UrlTree } from '@angular/router';
3  import { Observable } from 'rxjs';
4  import { SellerService } from '../services/seller.service';
5
6  @Injectable({
7    providedIn: 'root'
8  })
9  export class AuthGuard implements CanActivate {
10     constructor(private sellerService: SellerService) {}
11     canActivate(
12       route: ActivatedRouteSnapshot,
13       state: RouterStateSnapshot): Observable<boolean | UrlTree> | Promise<boolean | UrlTree> | boolean | UrlTree {
14       if(localStorage.getItem('seller')){
15         return true;
16       }
17       return this.sellerService.isSellerLoggedIn;
18     }
19   }
20 }

```

DATATYPE.TS:

src_Nutanix_study / src / app / ts / data-types.ts / ~\$ signUp

```
export interface signUp {  
  name: string;  
  email: string;  
  password: string;  
}  
  
export interface login {  
  email: String;  
  password: String;  
}  
  
export interface product{  
  name:string,  
  price:number,  
  category:string,  
  
  image:string,  
  description:string,  
  id:number,  
  quantity:undefined | number,  
  productId:undefined|number  
}  
  
export interface cart{  
  name:string,  
  price:number,  
  category:string,  
  
  image:string,  
  description:string,  
  id:number| undefined,  
  quantity:undefined | number,  
  productId:number,  
  userId:number  
}  
  
export interface priceSummary{  
  price:number,  
  discount:number,  
  tax:number,  
  delivery:number,  
  total:number
```

```
  description:string,  
  id:number| undefined,  
  quantity:undefined | number,  
  productId:number,  
  userId:number  
}  
}
```

```
export interface priceSummary{  
  price:number,  
  discount:number,  
  tax:number,  
  delivery:number,  
  total:number  
}  
}
```

```
export interface order {  
  email:string,  
  address:string,  
  contact:string,  
  totalPrice:number,  
  userId:string,  
  id:number|undefined  
}  
}
```

INDEX.HTML:

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

DB.JSON:

ar_Kitchen_Story > {} dbjson > {} seller > {} 2

```
{
  "seller": [
    {
      "name": "sam sidhu",
      "email": "sam@seller.com",
      "password": "abc123",
      "id": 2
    },
    {
      "name": "peter",
      "password": "abc123",
      "email": "peter@test.com",
      "id": 3
    },
    {
      "name": "bruce",
      "password": "abc123",
      "email": "bruce@test.com",
      "id": 4
    },
    {
      "name": "anil",
      "password": "abc123",
      "email": "anil@test.com",
      "id": 5
    },
    {
      "name": "alena",
      "password": "abc123",
      "email": "alena@test.com",
      "id": 6
    },
    {
      "name": "sam",
      "password": "abc123",
      "email": "sam@test.com",
      "id": 7
    }
  ]
}
```

```

},
"products": [
  {
    "name": "Idly",
    "price": "20",
    "category": "Breakfast",
    "description": "Savory steamed Rice cake",
    "image": "https://www.thulamfoods.com/product-images/Plain+Idly.png/956287000029846012/600x600",
    "id": 1
  },
  {
    "name": "Vada",
    "price": "30",
    "category": "Breakfast",
    "description": "Savory Fried Patties",
    "image": "https://www.gitsfood.com/wp-content/uploads/2021/04/04-19.jpg",
    "id": 2
  },
  {
    "name": "Dosa",
    "price": "35",
    "category": "Breakfast",
    "description": "Savory crepe",
    "image": "https://food.fnr.sndimg.com/content/dam/images/food/fullset/2019/10/9/DV3016_masala-dosa_s4x3.jpg.rend.hgtvcom.616.462.suffix/",
    "id": 3
  },
  {
    "name": "South Indian Meals",
    "price": "150",
    "category": "Lunch",
    "description": "Curries, Rice, Ghee, Papad, Curd",
    "image": "https://bangalore247.in/wp-content/uploads/2021/02/South-Indian-cuisine-In-Bangalore.jpg",
    "id": 4
  },
  {
    "name": "Veg Manchurian",
    "price": "100",
    "category": "Starter",
    "description": "Indian Chinese dish made by roughly chopping and deep-frying Veggies",
    "image": "https://vegecravings.com/wp-content/uploads/2017/03/veg-manchurian-dry-recipe-step-by-step-instructions-10.jpg",

```

Ln 18, Col 33 Spaces: 2 UTI

```

"users": [
  {
    "name": "anil",
    "email": "anil@user.com",
    "password": "abc123",
    "id": 1
  },
  {
    "name": "peter",
    "email": "peter@user.com",
    "password": "abc123",
    "id": 2
  },
  {
    "name": "hyden",
    "email": "hyden@user.com",
    "password": "abc123",
    "id": 3
  },
  {
    "name": "tony",
    "email": "tony@user.com",
    "password": "abc123",
    "id": 4
  }
],
"cart": [],
"orders": [
  {
    "email": "anil@user.com",
    "address": "noida sector x",
    "contact": "9999",
    "totalPrice": 104750,
    "userId": 1,
    "id": 1
  }
]

```