Source code

1: Sample Data

```typescript
import { Food } from "../Models/food";
import { Tag } from "../Models/tag";

export const sample_foods: Food[] = [
    {
        id: '1',
        name: 'Milk',
        price: 25,
        imageUrl: 'assets/foodItem1.jpg',
        tags: ['Veg'],
        company: ''
    },
    {
        id: '2',
        name: 'Bread',
        price: 20,
        imageUrl: 'assets/foodItem2.jpg',
        tags: ['Veg'],
        company: ''

    },
    {
        id: '3',
        name: 'Eggs',
        price: 10,
        imageUrl: 'assets/foodItem3.jpg',
```

2: Models

```typescript
export class Tag {

    name!:string;
    count!:number;

}
```

```typescript
import { CartItem } from "./cartItem";

export class Order  {

    id !: number;
    items !: CartItem[];
    totalPrice !: number;
    name !: string;
    address !: string;
    contact !: number;
    paymentId !: string;
    createdAt !: string;
    status !: string;

}
```

Tabs: `cart.ts` | `cartItem.ts` | `tags.component.css` | `app-routing.module.ts` | `header.component.ts` | `tag.ts` | **`food.ts`** ✕

`src > app > Models > food.ts > Food > tags`

```typescript
export class Food {

    id!:string;
    name!:string;
    price!:number;
    tags?: string[];
    imageUrl!: string;
    company!: string;

}
```

Tabs: `payment.component.css` | `payment.component.html M` | `payment.component.ts` | `order.ts` | `cart.ts` | **`cartItem.ts`** ✕

`src > app > Models > cartItem.ts > CartItem`

```typescript
import { Food } from "./food";

export class CartItem {
    constructor(public food: Food) { }
    quantity: number = 1;
    price: number = this.food.price;
}
```

Tabs: `t.component.ts` | `payment.component.css` | `payment.component.html M` | `payment.component.ts` | `order.ts` | **`cart.ts`** ✕

`src > app > Models > cart.ts > Cart`

```typescript
import { CartItem } from "./cartItem";

export class Cart {
    items: CartItem[] = [];
    totalPrice: number = 0;
    totalCount: number = 0;
}
```

3: Services

```typescript
import { Injectable } from '@angular/core';

import { Order } from '../Models/order';

@Injectable({
  providedIn: 'root'
})
export class OrderService {

  constructor() { }

  order!: Order;

  create(order : Order){
    return this.order = order;
  }

  getNewOrder(): Order {
    return this.order;
  }

}
```

```typescript
import { Injectable } from '@angular/core';

import { Food } from '../Models/food';
import { Tag } from '../Models/tag';
import { sample_foods, sample_tags } from '../Sample_data/data';

@Injectable({
  providedIn: 'root'
})
export class FoodService {

  constructor() { }

  getAll(): Food[] {
    return sample_foods;
  }

  getAllTags(): Tag[] {
    return sample_tags;
  }

  getAllFoodsByTag(tag: string) : Food[] {
    return tag == "All" ? this.getAll() : this.getAll().filter(food => food.tags?.includes(tag));
  }

  getAllFoodsBySearchTerm(searchTerm: string) {
```

PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL

```
√ Compiled successfully.
PS C:\Users\anushad\OneDrive - Cisco\Desktop\Training\Java_ESD\Java_Projects\Project4>
```

```typescript
import { Injectable } from '@angular/core';
import { BehaviorSubject, Observable } from 'rxjs';

import { Cart } from '../Models/cart';
import { CartItem } from '../Models/cartItem';
import { Food } from '../Models/food';

@Injectable({
  providedIn: 'root'
})
export class CartService {

  private cart: Cart = new Cart();
  private cartSubject: BehaviorSubject<Cart> = new BehaviorSubject(this.cart);

  constructor() { }

  addToCart(food: Food): void {
    let cartItem = this.cart.items.find(item => item.food.id === food.id);
    if (cartItem)
      return;

    this.cart.items.push(new CartItem(food));
    this.setCartToLocalStorage();
  }
```

4: Project Structure