

Advanced JavaScript Frameworks (Angular)

CIA-2

- **Project Title:** *Responsive Portfolio Website Detailing Restaurant Menu and Ordering System.*
- **Submitted By:**
 - o *2462306 Adithyan CR*
adithyan.cr@btech.christuniversity.in
 - o *2462308 Adwaith Praveen*
adwaith.praveen@btech.christuniversity.in
 - o *2462331 Jerolin Mathew*
jerolin.mathew@btech.christuniversity.in
 - o *2462332 Joel Anthony Dsilva*
joel.anthony@btech.christuniversity.in

- **Course:** *Advanced JavaScript Frameworks (Angular)*
- **Instructor Name:** Mr. Narendra
- **Institution:** *Christ University*
- **Date of Submission:** 22/01/2026
- **GitHub Repo:** <https://github.com/joel-dsilva/Restaurant-Menu-and-Ordering-System>
- **Published Output:** <https://joel-dsilva.github.io/Restaurant-Menu-and-Ordering-System/>

2. Abstract

This project presents a **Restaurant Menu and Ordering System** developed using **Angular** as part of the *Advanced JavaScript Frameworks* course. The application allows users to browse a restaurant menu, view food item details, add items to a cart, and place orders through a simple checkout process. The system consists of multiple views such as Home, Menu, Cart, and

Checkout, all connected using Angular routing. Core Angular concepts including component-based architecture, data binding, and services are used to manage application state and user interactions. The project simulates a real-world online food ordering workflow with a clean and responsive user interface.

3. Objectives

- Design a user-friendly and responsive restaurant ordering application
- Implement Angular components for modular development
- Use Angular routing for navigation between pages
- Manage cart and order data using Angular services
- Apply modern UI/UX principles suitable for real-world applications

4. Scope of the Project

- Front-end web application developed using Angular
- Includes menu browsing, cart management, and checkout flow
- No backend database or payment gateway integration
- Designed for desktop and responsive layouts
- Intended for academic and learning purposes

5. Tools & Technologies Used

Tool/Technology	Purpose
HTML5	Markup and content structure
CSS3	Styling and layout management
Angular	Front-end framework
TypeScript	Application logic
VS Code	Code editor
Angular CLI	Project scaffolding and build
Node.js	Runtime environment required for Angular
Chrome DevTools	Testing and debugging

6. Application Structure Overview

- Component-based architecture (Home, Menu, Cart, Checkout)

- Reusable components for menu items and cart items
- Angular services used for cart state management
- Routing module for seamless navigation
- Separation of concerns between UI and logic

7. Styling and UI Strategy

- Component-level CSS for styling
- Card-based layout for menu display
- Consistent color scheme and typography
- Responsive layout using flexible containers
- User feedback through alerts and buttons

8. Key Features

Feature	Description
Home Page	Introduction and navigation options
Menu Page	Displays food items with images and prices
Add to Cart	Adds selected items to the cart
Cart Page	View, remove items, and calculate total
Checkout	Collects customer details and confirms order
Alerts	Displays confirmation messages to users

9. Challenges Faced & Solutions

Challenge	Solution
Managing shared cart data	Implemented Angular service
Page navigation handling	Used Angular Router
Maintaining UI consistency	Applied reusable components
State management across pages	Centralized cart logic

10. Outcome

- Successfully developed a functional restaurant ordering system

- Implemented core Angular concepts effectively
- Achieved a clean, responsive, and user-friendly UI
- Gained practical experience with Angular architecture

11. Future Enhancements

- Backend integration with database support
- Payment gateway integration
- Quantity update functionality in cart
- User authentication and order history
- Admin panel for menu management

12. Sample Code

File Structure:

```
✓ RESTAURANT MENU AND ORDERING SYSTEM USING ANGULAR AND TYPESCRIPT
  ✓ restaurant
    ✓ src
      ✓ app
        ✓ components
          > cart
          > home
          > menu-detail
          > menu-list
          > navbar
          ✓ order-form
            TS order-form.component.ts
          > models
          > services
          TS app.component.ts
          TS app.config.ts
          # app.css
          <> app.html
          TS app.module.ts
          TS app.routes.ts
```



```
src > app > components > cart > cart.component.ts > ...
1 import { Component } from '@angular/core';
2 import { CommonModule } from '@angular/common';
3 import { RouterModule } from '@angular/router';
4 import { CartService } from '../../../../../services/cart.service';
5
6 @Component({
7   selector: 'app-cart',
8   standalone: true,
9   imports: [CommonModule, RouterModule],
10  template: `
11    <div class="cart-container">
12      <h2>Your Cart</h2>
13
14      <div *ngIf="cartService.cartItems.length === 0" class="empty-cart">
15        <p>Your cart is empty</p>
16        <a routerLink="/menu">Go to Menu</a>
17      </div>
18
19      <div *ngIf="cartService.cartItems.length > 0">
20        <div *ngFor="let item of cartService.cartItems" class="cart-item">
21          <h4>{{ item.menuItem.name }}</h4>
22
23          <p>
24            Price:
25            {{ item.menuItem.price | currency:'USD' }}
26          </p>
27
28          <p>
29            Quantity: {{ item.quantity }}</p>
30
31          <p>
32            Total:
33            {{ item.menuItem.price * item.quantity | currency:'USD' }}
34          </p>
35
36          <button (click)="removeItem(item.menuItem.id)">Remove</button>
37        </div>
38      </div>
39    </div>
40  </div>
41
```

```
src > app > components > cart > ts cart.component.ts ...
  text-align: center;
  padding: 40px;
}
.cart-item {
  border: 1px solid #ddd;
  padding: 15px;
  margin-bottom: 10px;
  border-radius: 4px;
}
.cart-summary {
  margin-top: 30px;
  padding: 20px;
  background: #f5f5f5;
  border-radius: 4px;
}
button {
  margin: 5px;
  padding: 8px 16px;
  border: none;
  border-radius: 4px;
  cursor: pointer;
}
.checkout-btn {
  background: #4caf50;
  color: white;
}
}

export class CartComponent {
  constructor(public cartService: CartService) {}

  removeItem(itemID: number) {
    this.cartService.removeItem(itemID);
  }

  clearCart() {
    this.cartService.clearCart();
  }
}
```

File Edit Selection View Go Run Terminal Help ↶ ↷ Q restaurant

EXPLORER

OPEN EDITORS

RESTAURANT

- Angular
- .vscode
- node_modules
- public
- src
 - app
 - components
 - cart
 - home

home.component.ts

```

src > app > components > home > TS home.component.ts > ...
1 import { Component } from '@angular/core';
2 import { RouterModule } from '@angular/router';
3
4 @Component({
5     selector: 'app-home',
6     standalone: true,
7     imports: [RouterModule],
8     template: `
9         <div class="home">
10             <!-- Header with background image -->
11             <div class="hero">
12                 <div class="hero-content">
13                     <h1>Restaurant App</h1>
14                     <p class="tagline">Delicious food made just for you</p>
15                 </div>
16             </div>
17
18             <!-- Main content -->
19             <main class="content">
20                 <div class="welcome-section">
21                     <h2>Welcome to Our Restaurant</h2>
22                     <p>We serve fresh, delicious meals made from high-quality ingredients. From appetizers to dessert, we have something for everyone.</p>
23                 </div>
24
25             <!-- Quick navigation -->
26             <div class="quick-links">
27                 <div class="link-card">
28                     <div class="link-icon">☰</div>
29                     <h3>Browse Menu</h3>
30                     <p>Check out our delicious offerings</p>
31                     <a routerLink="/menu" class="link-btn">View Menu</a>
32                 </div>
33
34                 <div class="link-card">
35                     <div class="link-icon">🛒</div>
36                     <h3>Your Cart</h3>
37                     <p>Review your selected items</p>
38             </div>
39         </div>
40     </main>
41
42     <footer>
43         <p>Footer content here</p>
44     </footer>
45
46     <style>
47         .hero {
48             margin-top: 60px;
49         }
50
51         .hero p {
52             margin: 8px 0;
53             opacity: 0.8;
54             font-size: 0.9rem;
55         }
56
57         /* Responsive */
58         @media (max-width: 768px) {
59             .hero {
60                 padding: 40px 20px;
61             }
62
63             .hero h1 {
64                 font-size: 2rem;
65             }
66
67             .quick-links {
68                 grid-template-columns: 1fr;
69             }
70
71             .cta-buttons {
72                 flex-direction: column;
73                 align-items: center;
74             }
75
76             .btn {
77                 width: 200px;
78                 text-align: center;
79             }
80
81         }
82     </style>
83
84     export class HomeComponent {}
```

Ln 1, Col 1 Spaces: 2 UTF-8 CRLF

File Edit Selection View Go Run Terminal Help ↶ ↷ Q restaurant

EXPLORER

OPEN EDITORS

RESTAURANT

- Angular
- .vscode
- node_modules
- public
- src
 - app
 - components
 - cart
 - home

home.component.ts

```

src > app > components > home > TS home.component.ts > ...
248     margin-top: 60px;
249 }
250
251 .footer p {
252     margin: 8px 0;
253     opacity: 0.8;
254     font-size: 0.9rem;
255 }
256
257 /* Responsive */
258 @media (max-width: 768px) {
259     .hero {
260         padding: 40px 20px;
261     }
262
263     .hero h1 {
264         font-size: 2rem;
265     }
266
267     .quick-links {
268         grid-template-columns: 1fr;
269     }
270
271     .cta-buttons {
272         flex-direction: column;
273         align-items: center;
274     }
275
276     .btn {
277         width: 200px;
278         text-align: center;
279     }
280
281 }
282
283 export class HomeComponent {}
```

Ln 1, Col 1 Spaces: 2 UTF-8 CRLF



File Edit Selection View Go Run Terminal Help ↵ → 🔍 restaurant

EXPLORER

OPEN EDITORS

RESTAURANT

- > .angular
- > .vscode
- > node_modules
- > public
- > src
 - > app
 - > components
 - > cart
 - > home
 - > menu-detail
 - TS menu-detail.component.ts

menu-detail.component.ts

```
src > app > components > menu-detail > TS menu-detail.component.ts ...  
1 import { Component, OnInit } from '@angular/core';  
2 import { CommonModule } from '@angular/common';  
3 import { ActivatedRoute, RouterModule } from '@angular/router';  
4 import { MenuService } from '../../../../../services/menu.service';  
5 import { CartService } from '../../../../../services/cart.service';  
6 import { MenuItem } from '../../../../../models/menu-item.model';  
7 import { ToastService } from '../../../../../services/toast.service';  
8  
9 @Component({  
10   selector: 'app-menu-detail',  
11   standalone: true,  
12   imports: [CommonModule, RouterModule],  
13   template: '  
14     <div class="detail-container" *ngIf="menuItem">  
15       <a routerLink="/menu" class="back-link">< Back to Menu</a>  
16       <d (element) img: HTMLElement  
17         <img  
18           [src]="menuItem.imageUrl"  
19           [alt]="menuItem.name"  
20           class="detail-image"  
21         >  
22       </d>  
23       <h2>{{ menuItem.name }}</h2>  
24       <p class="description">{{ menuItem.description }}</p>  
25  
26       <div class="info-row">  
27         <span class="price">  
28           {{ menuItem.price | currency:'USD' }}  
29         </span>  
30         <span class="category">{{ menuItem.category }}</span>  
31       </div>  
32  
33       <div class="tags">  
34         <span *ngIf="menuItem.isVegetarian" class="tag veg">  
35           🥦 Vegetarian  
36         </span>  
37       </div>  
38     </div>  
39   
```

Ln 1, Col 1 Spaces: 2 UTF-8 CRLF

File Edit Selection View Go Run Terminal Help ↵ → 🔍 restaurant

EXPLORER

OPEN EDITORS

RESTAURANT

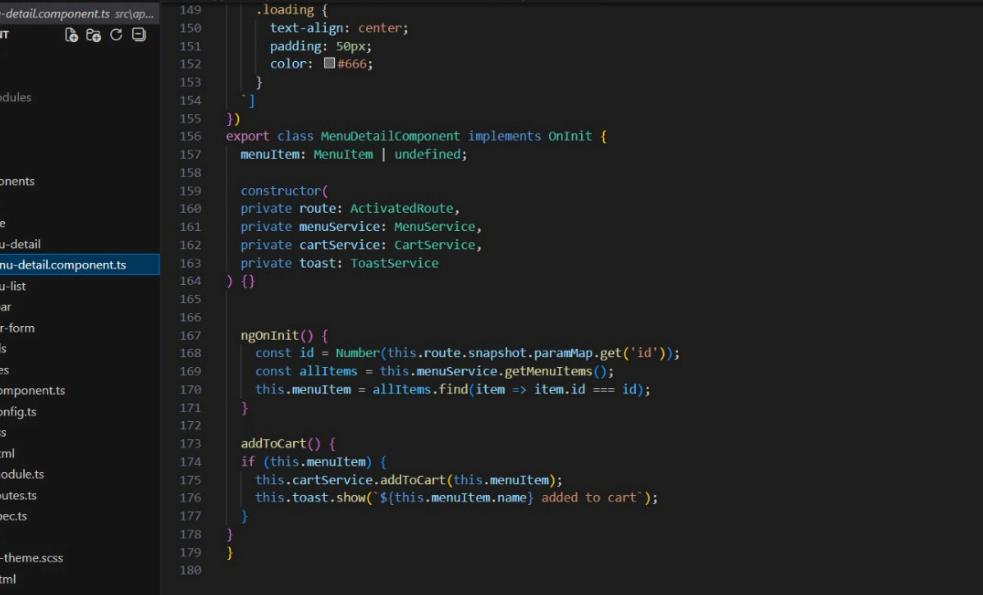
- > .angular
- > .vscode
- > node_modules
- > public
- > src
 - > app
 - > components
 - > cart
 - > home
 - > menu-detail
 - TS menu-detail.component.ts

menu-detail.component.ts

```
src > app > components > menu-detail > TS menu-detail.component.ts ...  
73   color: #3f51b5;  
74   text-decoration: none;  
75 }  
76 .detail-card {  
77   border: 1px solid #ddd;  
78   padding: 25px;  
79   border-radius: 8px;  
80   box-shadow: 0 2px 8px rgba(0,0,0,0.1);  
81 }  
82 .detail-image {  
83   width: 100%;  
84   height: 300px;  
85   object-fit: cover;  
86   border-radius: 4px;  
87   margin-bottom: 20px;  
88 }  
89 .description {  
90   font-size: 1.1em;  
91   line-height: 1.6;  
92   color: #555;  
93   margin: 15px 0;  
94 }  
95 .info-row {  
96   display: flex;  
97   justify-content: space-between;  
98   align-items: center;  
99   margin: 20px 0;  
100 }  
101 .price {  
102   font-size: 1.8em;  
103   font-weight: bold;  
104   color: #3f51b5;  
105 }  
106 .category {  
107   background: #f0f0f0;  
108   padding: 8px 16px;  
109   border-radius: 20px;
```

Ln 1, Col 1 Spaces: 2 UTF-8





```
src > app > components > menu-detail > TS menu-detail.component.ts > ...
149     .loading {
150         text-align: center;
151         padding: 50px;
152         color: #666;
153     }
154 }
155 })
156 export class MenuDetailComponent implements OnInit {
157     menuItem: MenuItem | undefined;
158
159     constructor(
160         private route: ActivatedRoute,
161         private menuService: Menuservice,
162         private cartService: CartService,
163         private toast: ToastService
164     ) {}
165
166
167     ngOnInit() {
168         const id = Number(this.route.snapshot.paramMap.get('id'));
169         const allItems = this.menuService.getMenuItems();
170         this.menuItem = allItems.find(item => item.id === id);
171     }
172
173     addToCart() {
174         if (this.menuItem) {
175             this.cartService.addToCart(this.menuItem);
176             this.toast.show(`${this.menuItem.name} added to cart`);
177         }
178     }
179 }
180
```

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer (Left):** Shows the project structure under "RESTAURANT". The "menu-list.component.ts" file is currently selected.
- Editor (Center):** Displays the code for "menu-list.component.ts". The code defines a component for a menu list, utilizing Angular's CommonModule, RouterModule, and MenuService.
- Terminal (Bottom Left):** Shows the command "ng serve" being run.
- Status Bar (Bottom):** Shows the status "Up to date" and the file path "src/app/components/menu-list/menu-list.component.ts".

```
src > app > components > menu-list > ts menu-list.component.ts ...  
1 import { Component, OnInit } from '@angular/core';  
2 import { CommonModule } from '@angular/common';  
3 import { RouterModule } from '@angular/router';  
4 import { MenuService } from '../../../../../services/menu.service';  
5 import { CartService } from '../../../../../services/cart.service';  
6 import { MenuItem } from '../../../../../models/menu-item.model';  
7  
8 @Component({  
9   selector: 'app-menu-list',  
10  standalone: true,  
11  imports: [CommonModule, RouterModule],  
12  template: '  
13    <div class="menu-container">  
14      <h2>Our Menu</h2>  
15  
16      <div class="menu-grid">  
17        <div *ngFor="let item of menuItems" class="menu-card">  
18          <img  
19            [src] = "item.imageUrl"  
20            [alt] = "item.name"  
21            class = "menu-image"  
22          >  
23  
24          <h3>{{ item.name }}</h3>  
25          <p>{{ item.description }}</p>  
26  
27          <p class="price">  
28            {{ item.price | currency:'USD' }}  
29          </p>  
30  
31          <p class="category">{{ item.category }}</p>  
32  
33          <div class="tags">  
34            <span *ngIf = "item.isVegetarian" class = "tag veg">  
35              Veg  
36            </span>  
37            <span *ngIf = "item.isSpicy" class = "tag spicy">  
38              Spicy  
39            </span>  
40          </div>  
41        </div>  
42      </div>  
43    </div>  
44  </div>  
45</div>
```



```
File Edit Selection View Go Run Terminal Help ← → 🔍 restaurant ⌂ ...  
EXPLORER OPEN EDITORS RESTAURANT .angular .vscode node_modules public src app components cart home menu-detail menu-list TS menu-list.component.ts ...  
src > app > components > menu-list > TS menu-list.component.ts ...  
1 import { Component, OnInit } from '@angular/core';  
2 import { CommonModule } from '@angular/common';  
3 import { RouterModule } from '@angular/router';  
4 import { MenuService } from '../../../../../services/menu.service';  
5 import { CartService } from '../../../../../services/cart.service';  
6 import { MenuItem } from '../../../../../models/menu-item.model';  
7  
8 @Component({  
9   selector: 'app-menu-list',  
10  standalone: true,  
11  imports: [CommonModule, RouterModule],  
12  template: '  
13    <div class="menu-container">  
14      <h2>Our Menu</h2>  
15  
16      <div class="menu-grid">  
17        <div *ngFor="let item of menuItems" class="menu-card">  
18          <img  
19            [src]="item.imageUrl"  
20            [alt]="item.name"  
21            class="menu-image"  
22          >  
23  
24          <h3>{{ item.name }}</h3>  
25          <p>{{ item.description }}</p>  
26  
27          <p class="price">  
28            {{ item.price | currency:'USD' }}  
29          </p>  
30  
31          <p class="category">{{ item.category }}</p>  
32  
33          <div class="tags">  
34            <span *ngIf="item.isVegetarian" class="tag veg">  
35              Veg  
36            </span>  
37            <span *ngIf="item.isSpicy" class="tag spicy">
```

```
src > app > components > menu-list > TS menu-list.component.ts > ...
105     justify-content: space-between;
106     margin-top: 15px;
107   }
108   button, a {
109     padding: 8px 16px;
110     border: none;
111     border-radius: 4px;
112     cursor: pointer;
113     text-decoration: none;
114   }
115   .add-btn {
116     background: #3f51b5;
117     color: white;
118   }
119   a {
120     background: #f5f5f5;
121     color: #333;
122   }
123 }
124 }
125 export class MenulistComponent implements OnInit {
126   menuItems: MenuItem[] = [];
127
128   constructor(
129     private menuService: MenuService,
130     private cartService: CartService
131   ) {}
132
133   ngOnInit() {
134     this.menuItems = this.menuService.getMenuItems();
135   }
136
137   addToCart(item: MenuItem) {
138     this.cartService.addToCart(item);
139     alert(`${item.name} added to cart!`);
140   }
141 }
```



File Edit Selection View Go Run Terminal Help

restaurant

EXPLORER

OPEN EDITORS

RESTAURANT

.angular
.vscode
node_modules
public
src
app
components
cart
home
menu-detail
menu-list
navbar

TS navbar.component.ts

```
src > app > components > navbar > TS navbar.component.ts > ...
1 import { Component } from '@angular/core';
2 import { RouterModule } from '@angular/router';
3 import { CartService } from '../../../../../services/cart.service';
4
5 @Component({
6   selector: 'app-navbar',
7   standalone: true,
8   imports: [RouterModule],
9   template:
10     `<nav class="navbar">
11       <h1>Restaurant App</h1>
12       <div class="nav-links">
13         <a routerLink="/menu">Menu</a>
14         <a routerLink="/cart">Cart ({ cartService.getItemCount() })</a>
15       </div>
16     </nav>
17   `,
18   styles: [
19     .navbar {
20       background: #3f51b5;
21       color: white;
22       padding: 1rem;
23       display: flex;
24       justify-content: space-between;
25       align-items: center;
26     }
27     .nav-links a {
28       color: white;
29       margin-left: 20px;
30       text-decoration: none;
31     }
32     .nav-links a:hover {
33       text-decoration: underline;
34     }
35   ]
36 }
37
38 export class NavbarComponent {
```

Ln 1, Col 1 Spaces: 2 UTF-8 CRLF

File Edit Selection View Go Run Terminal Help

restaurant

EXPLORER

OPEN EDITORS

RESTAURANT

.angular
.vscode
node_modules
public
src
app
components
cart
home
menu-detail
menu-list
navbar
order-form

TS order-form.component.ts

```
src > app > components > order-form > TS order-form.component.ts > ...
1 import { Component } from '@angular/core';
2 import { CommonModule } from '@angular/common';
3 import { FormsModule } from '@angular/forms';
4 import { Router, RouterModule } from '@angular/router';
5 import { CartService } from '../../../../../services/cart.service';
6
7 @Component({
8   selector: 'app-order-form',
9   standalone: true,
10  imports: [CommonModule, FormsModule, RouterModule],
11  template:
12    `<div class="checkout-container">
13      <h2>Checkout</h2>
14
15      <div *ngIf="cartService.cartItems.length === 0" class="empty-cart">
16        <p>Your cart is empty</p>
17        <button routerLink="/menu">Go to Menu</button>
18      </div>
19
20      <div *ngIf="cartService.cartItems.length > 0">
21        <div class="order-summary">
22          <h3>Order Summary</h3>
23
24          <div *ngFor="let item of cartService.cartItems" class="order-item">
25            {{ item.menuItem.name }} x {{ item.quantity }} =
26            {{ item.menuItem.price * item.quantity | currency:'USD' }}
27          </div>
28
29          <p class="total">
30            Total: {{ cartService.getTotal() | currency:'USD' }}
31          </p>
32        </div>
33
34        <form (ngSubmit)="placeOrder()" class="checkout-form">
35          <h3>Customer Information</h3>
36
37          <div class="form-group">
```

Ln 1, Col 1 Spaces: 2 UTF-8 CRLF



File Edit Selection View Go Run Terminal Help ↶ ↷

RESTAURANT .angular .vscode node_modules public src app components cart home menu-detail menu-list navbar order-form

order-form.component.ts

```
src > app > components > order-form > TS order-form.component.ts > ...
103 }
104 button {
105   padding: 10px 20px;
106   border: none;
107   border-radius: 4px;
108   cursor: pointer;
109 }
110 .submit-btn {
111   background: #4CAF50;
112   color: white;
113 }
114 ]
115 )
116 export class OrderFormComponent {
117   customerName = '';
118   customerEmail = '';
119   customerPhone = '';
120
121   constructor(
122     public cartService: CartService,
123     private router: Router
124   ) {}
125
126   placeOrder() {
127     if (this.customerName && this.customerEmail && this.customerPhone) {
128       alert(
129         'Order placed successfully!
130         Total: ${this.cartService.getTotal()}'
131       );
132       thank you, ${this.customerName}!';
133     }
134     this.cartService.clearCart();
135     this.router.navigate(['/menu']);
136   } else {
137     alert('Please fill in all fields');
138   }
139 }
```

Ln 1, Col 1 Spaces: 2 UTF-8 CRLF

File Edit Selection View Go Run Terminal Help ↶ ↷

RESTAURANT .angular .vscode node_modules public src app components cart home menu-detail menu-list navbar order-form models

cart-item.model.ts

```
src > app > models > TS cart-item.model.ts > ...
1 import { MenuItem } from './menu-item.model';
2
3 export interface CartItem {
4   menuItem: MenuItem;
5   quantity: number;
6 }
```

Ln 1, Col 1 Spaces: 2 UTF-8 CRLF



File Edit Selection View Go Run Terminal Help ↵ → 🔍 restaurant

EXPLORER OPEN EDITORS RESTAURANT .angular .vscode node_modules public src app components cart home menu-detail menu-list navbar order-form models cart-item.model.ts TS menu-item.model.ts services app.component.ts app.config.ts # app.css app.html app.module.ts app.routes.ts app.spec.ts app.ts custom-theme.scss index.html main.ts

OUTLINE TIMELINE

TS menu-item.model.ts

```
src > app > models > TS menu-item.model.ts > ⚡ MenuItem
1 export interface MenuItem {
2   id: number;
3   name: string;
4   description: string;
5   price: number;
6   category: string;
7   imageUrl: string;
8   ingredients: string[];
9   isVegetarian: boolean;
10  isSpicy: boolean;
11  isAvailable: boolean;
12 }
```

Ln 1, Col 1 Spaces: 2 UTF-8 CRLF

File Edit Selection View Go Run Terminal Help ↵ → 🔍 restaurant

EXPLORER OPEN EDITORS RESTAURANT .angular .vscode node_modules public src app components cart home menu-detail menu-list navbar order-form models cart-item.model.ts TS menu-item.model.ts services cart.service.ts

OUTLINE TIMELINE

TS cart.service.ts

```
src > app > services > TS cart.service.ts ...
1 import { Injectable } from '@angular/core';
2 import { MenuItem } from './models/menu-item.model';
3 import { CartItem } from '../models/cart-item.model';
4
5 @Injectable({
6   providedIn: 'root'
7 })
8 export class CartService {
9   cartItems: CartItem[] = [];
10
11   addToCart(item: MenuItem): void {
12     const existing = this.cartItems.find(ci => ci.menuItem.id === item.id);
13
14     if (existing) {
15       existing.quantity++;
16     } else {
17       this.cartItems.push({ menuItem: item, quantity: 1 });
18     }
19   }
20
21   removeItem(itemId: number): void {
22     this.cartItems = this.cartItems.filter(ci => ci.menuItem.id !== itemId);
23   }
24
25   getTotal(): number {
26     return this.cartItems.reduce((sum, item) =>
27       sum + (item.menuItem.price * item.quantity), 0);
28   }
29
30   getItemCount(): number {
31     return this.cartItems.reduce((count, item) => count + item.quantity, 0);
32   }
33
34   clearCart(): void {
35     this.cartItems = [];
36   }
37 }
```

Ln 1, Col 1 Spaces: 2 UTF-8 CRLF



```
src > app > services > TS menu.service.ts > ...
1 import { Injectable } from '@angular/core';
2 import { MenuItem } from '../models/menu-item.model';
3
4 @Injectable({
5   providedIn: 'root'
6 })
7 export class MenuService {
8   getMenuItems(): MenuItem[] {
9     return [
10       {
11         id: 1,
12         name: 'Margherita Pizza',
13         description: 'Classic pizza with tomato sauce and mozzarella',
14         price: 12.99,
15         category: 'Pizza',
16         imageUrl: 'https://images.unsplash.com/photo-1604068549290-dea0e4a305ca?w=400',
17         ingredients: ['Tomato sauce', 'Mozzarella', 'Basil'],
18         isVegetarian: true,
19         isSpicy: false,
20         isAvailable: true
21     },
22     {
23       id: 2,
24       name: 'Spicy Chicken Burger',
25       description: 'Juicy chicken burger with spicy mayo',
26       price: 10.99,
27       category: 'Burger',
28       imageUrl: 'https://images.unsplash.com/photo-1568901346375-23c9450c58cd?w=400',
29       ingredients: ['Chicken Patty', 'Spicy Mayo', 'Lettuce'],
30       isVegetarian: false,
31       isSpicy: true,
32       isAvailable: true
33     },
34     {
35       id: 3,
36       name: 'Caesar Salad',
37       description: 'Fresh salad with Caesar dressing',
38     }
39   ]
40 }
```

```
File Edit Selection View Go Run Terminal Help < > Q restaurant
EXPLORER OPEN EDITORS
  X TS toast.service.ts src\app\services
  RESTAURANT .angular .vscode node_modules public
  src app
    components cart home menu-detail menu-list navbar order-form
    models cart-item.model.ts menu-item.model.ts
    services cart.service.ts menu.service.ts
    TS toast.service.ts
  TS app.component.ts
  TS app.config.ts
  # app.css
  <> app.html
  TS app.module.ts
  TS app.routes.ts
  TS app.spec.ts
  TS app.ts
  > OUTLINE
  > TIMELINE
  ↵ main ⇨ 0 △ 0
```

```
src > app > services > TS toast.service.ts ...
1 import { Injectable } from '@angular/core';
2
3 @Injectable({
4   providedIn: 'root'
5 })
6 export class ToastService {
7   show(message: string) {
8     const toast = document.createElement('div');
9     toast.innerText = message;
10
11   toast.style.position = 'fixed';
12   toast.style.bottom = '20px';
13   toast.style.left = '50%';
14   toast.style.transform = 'translateX(-50%)';
15   toast.style.background = '□#333';
16   toast.style.color = '■fff';
17   toast.style.padding = '10px 20px';
18   toast.style.borderRadius = '6px';
19   toast.style.zIndex = '1000';
20
21   document.body.appendChild(toast);
22
23   setTimeout(() => toast.remove(), 2000);
24 }
25
26 }
```



File Edit Selection View Go Run Terminal Help ↺ → Q restaurant

EXPLORER OPEN EDITORS RESTAURANT src/app app component.ts

```
src > app > TS app.component.ts src\app
1 import { Component } from '@angular/core';
2 import { RouterModule } from '@angular/router';
3 import { NavbarComponent } from './components/navbar/navbar.component';
4
5 @Component({
6   selector: 'app-root',
7   standalone: true,
8   imports: [RouterModule, NavbarComponent],
9   template: `
10   <div class="app">
11     <app-navbar></app-navbar>
12     <div class="content">
13       <router-outlet></router-outlet>
14     </div>
15   </div>
16   `,
17   styles: [
18     .app {
19       min-height: 100vh;
20     }
21     .content {
22       padding: 20px;
23     }
24   ]
25 })
26 export class AppComponent {
27   title = 'restaurant-app';
28 }
```

Ln 1, Col 1 Spaces: 2 UTF-8 CRLF

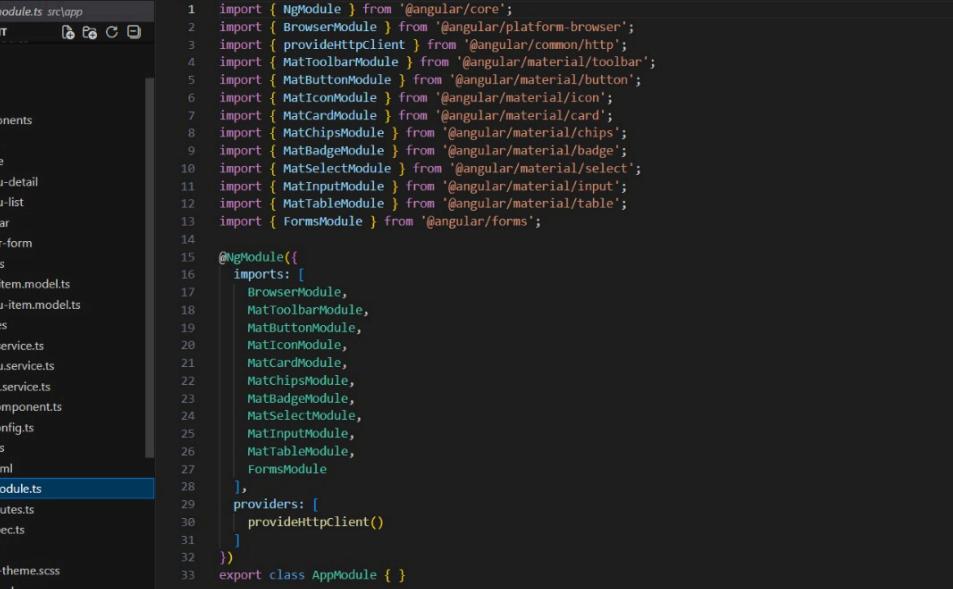
File Edit Selection View Go Run Terminal Help ↺ → Q restaurant

EXPLORER OPEN EDITORS RESTAURANT src/app app config.ts

```
src > app > TS app.config.ts src\app
1 import { ApplicationConfig } from '@angular/core';
2 import { provideRouter } from '@angular/router';
3 import { routes } from './app.routes';
4
5 export const appConfig: ApplicationConfig = {
6   providers: [
7     provideRouter(routes)
8   ]
9 };
10
```

Ln 1, Col 1 Spaces: 2 UTF-8 CRLF





The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** On the left, it lists the project structure under "OPEN EDITORS". The file "app.module.ts" is currently selected and highlighted in blue.
- Code Editor:** The main area displays the content of "app.module.ts". The code imports various Angular modules and defines the AppModule with its imports, providers, and export class.
- Terminal:** At the bottom, the terminal shows the command "main" and some status indicators like "0 △ 0".
- Status Bar:** At the bottom right, it shows "Ln 1, Col 1" and other system information.

```
src > app > app.module.ts > ...
1 import { NgModule } from '@angular/core';
2 import { BrowserModule } from '@angular/platform-browser';
3 import { provideHttpClient } from '@angular/common/http';
4 import { MatToolbarModule } from '@angular/material/toolbar';
5 import { MatButtonModule } from '@angular/material/button';
6 import { MatIconModule } from '@angular/material/icon';
7 import { MatCardModule } from '@angular/material/card';
8 import { MatChipsModule } from '@angular/material/chips';
9 import { MatBadgeModule } from '@angular/material/badge';
10 import { MatSelectModule } from '@angular/material/select';
11 import { MatInputModule } from '@angular/material/input';
12 import { MatTableModule } from '@angular/material/table';
13 import { FormsModule } from '@angular/forms';
14
15 @NgModule({
16   imports: [
17     BrowserModule,
18     MatToolbarModule,
19     MatButtonModule,
20     MatIconModule,
21     MatCardModule,
22     MatChipsModule,
23     MatBadgeModule,
24     MatSelectModule,
25     MatInputModule,
26     MatTableModule,
27     FormsModule
28   ],
29   providers: [
30     provideHttpClient()
31   ]
32 })
33 export class AppModule { }
```

The screenshot shows a code editor interface with the following details:

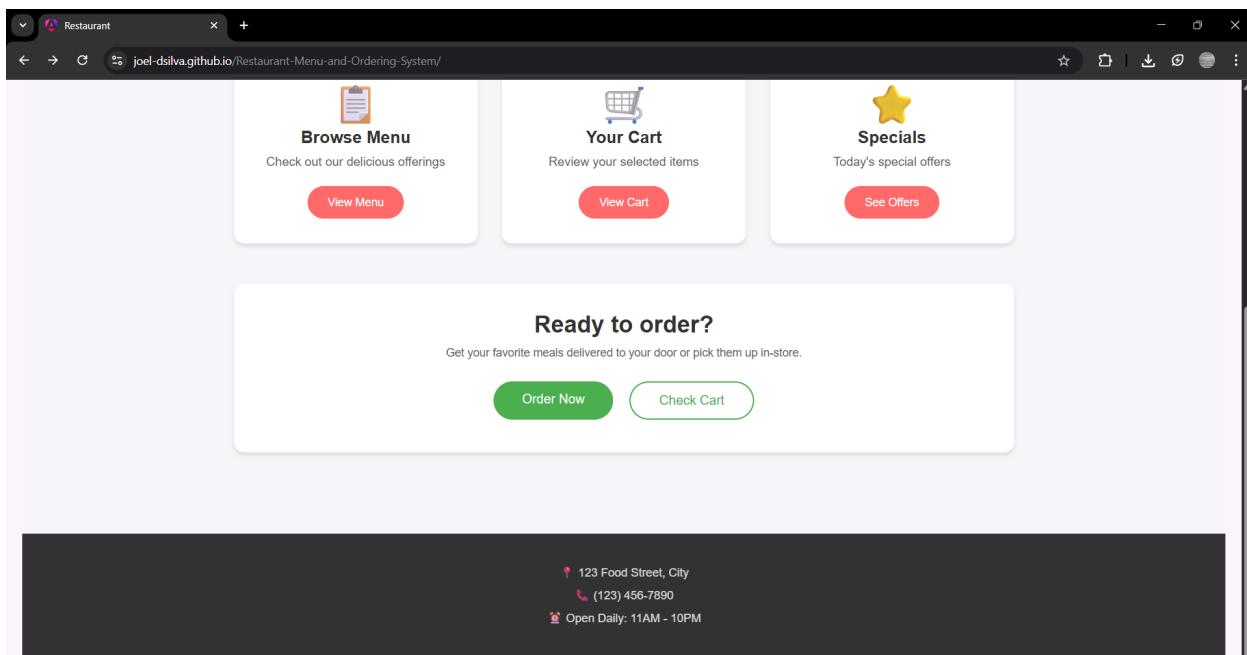
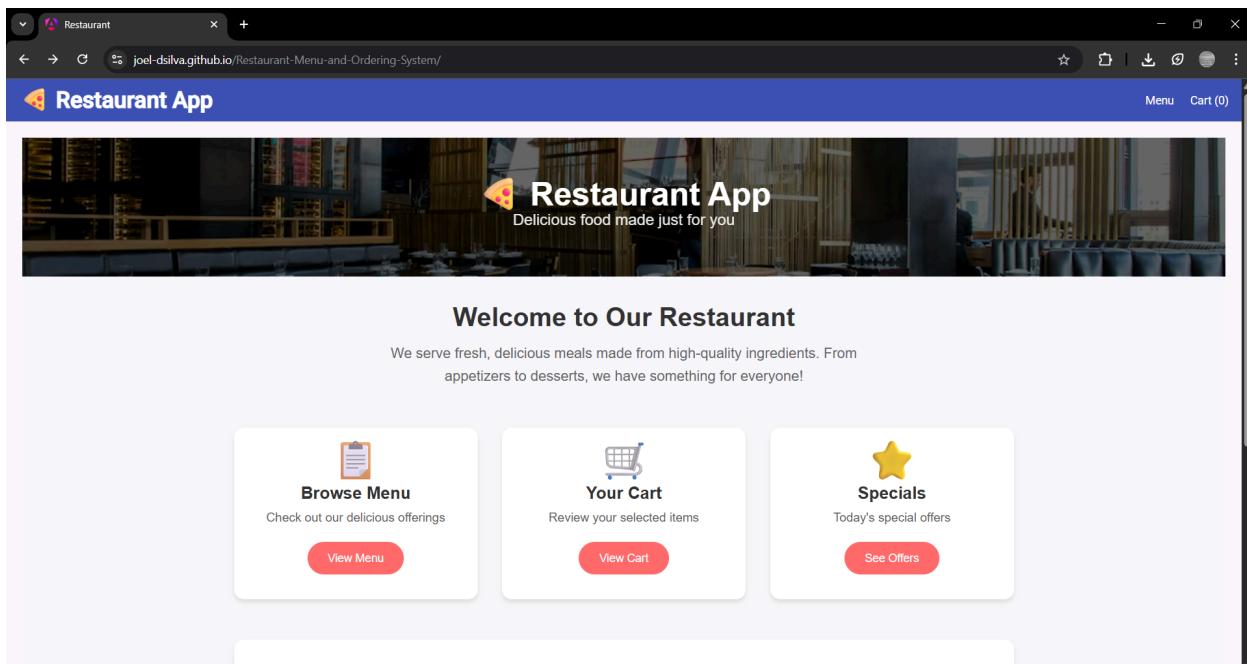
- File Explorer (Left):** Shows the project structure under "OPEN EDITORS". The "src/app" folder is expanded, revealing components like "app", "cart", "home", "menu-detail", "menu-list", "navbar", "order-form", and "models". Other files include "cart-item.model.ts", "menu-item.model.ts", "services" (with "cart.service.ts", "menu.service.ts", and "toast.service.ts"), and "app" (with "component.ts", "config.ts", "css", "html", "module.ts", and "routes.ts").
- Editor Area (Center):** The file "app.routes.ts" is open, showing the following TypeScript code for defining routes:

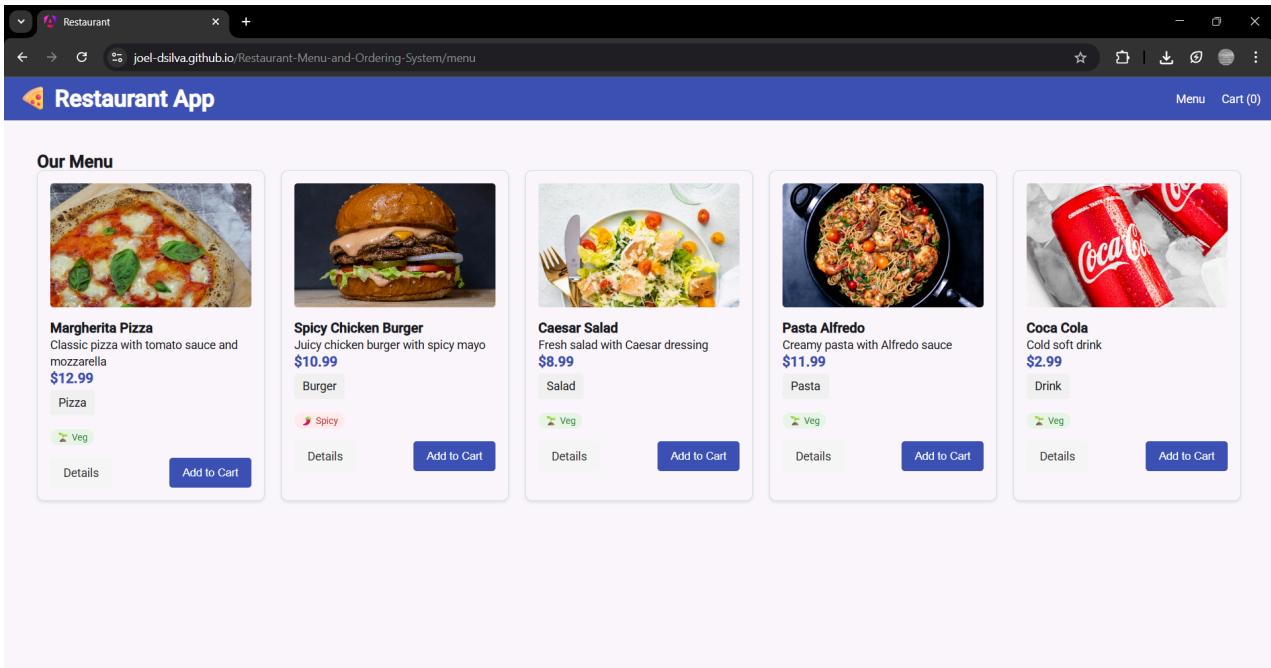
```
src > app > TS app.routes.ts ...  
1 import { Routes } from '@angular/router';  
2 import { HomeComponent } from './components/menu-list/menu-list.component';  
3 import { MenuListComponent } from './components/menu-detail/menu-detail.component';  
4 import { CartComponent } from './components/cart/cart.component';  
5 import { HomeComponent } from './components/home/home.component';  
6 export const routes: Routes = [  
7   { path: '', component: HomeComponent },  
8   { path: 'menu', component: MenuListComponent },  
9   { path: 'menu/:id', component: MenuDetailComponent },  
10  { path: 'cart', component: CartComponent },  
11  {  
12    path: 'checkout',  
13    loadComponent: () => import('./components/order-form/order-form.component')  
14      .then(m => m.OrderformComponent)  
15  },  
16  { path: '**', redirectTo: '/menu' }  
17];
```

- Top Bar:** Includes File, Edit, Selection, View, Go, Run, Terminal, Help, and a search bar.
- Bottom Status Bar:** Shows "Ln 1, Col 1" and "Spaces: 2" and "CRLF".

13. Screenshots of Final Output

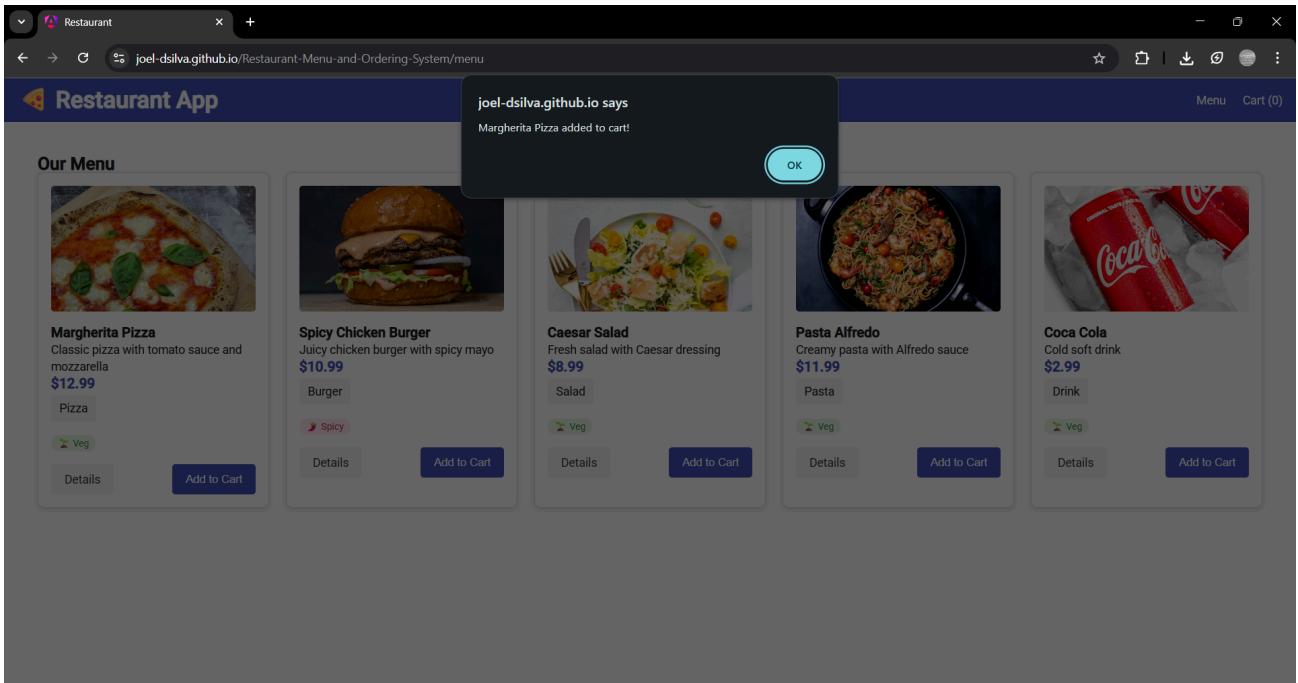
Home:



Menu:


Our Menu

- Margherita Pizza**
Classic pizza with tomato sauce and mozzarella
\$12.99
Pizza
Veg
[Details](#) [Add to Cart](#)
- Spicy Chicken Burger**
Juicy chicken burger with spicy mayo
\$10.99
Burger
Spicy
[Details](#) [Add to Cart](#)
- Caesar Salad**
Fresh salad with Caesar dressing
\$8.99
Salad
Veg
[Details](#) [Add to Cart](#)
- Pasta Alfredo**
Creamy pasta with Alfredo sauce
\$11.99
Pasta
Veg
[Details](#) [Add to Cart](#)
- Coca Cola**
Cold soft drink
\$2.99
Drink
Veg
[Details](#) [Add to Cart](#)

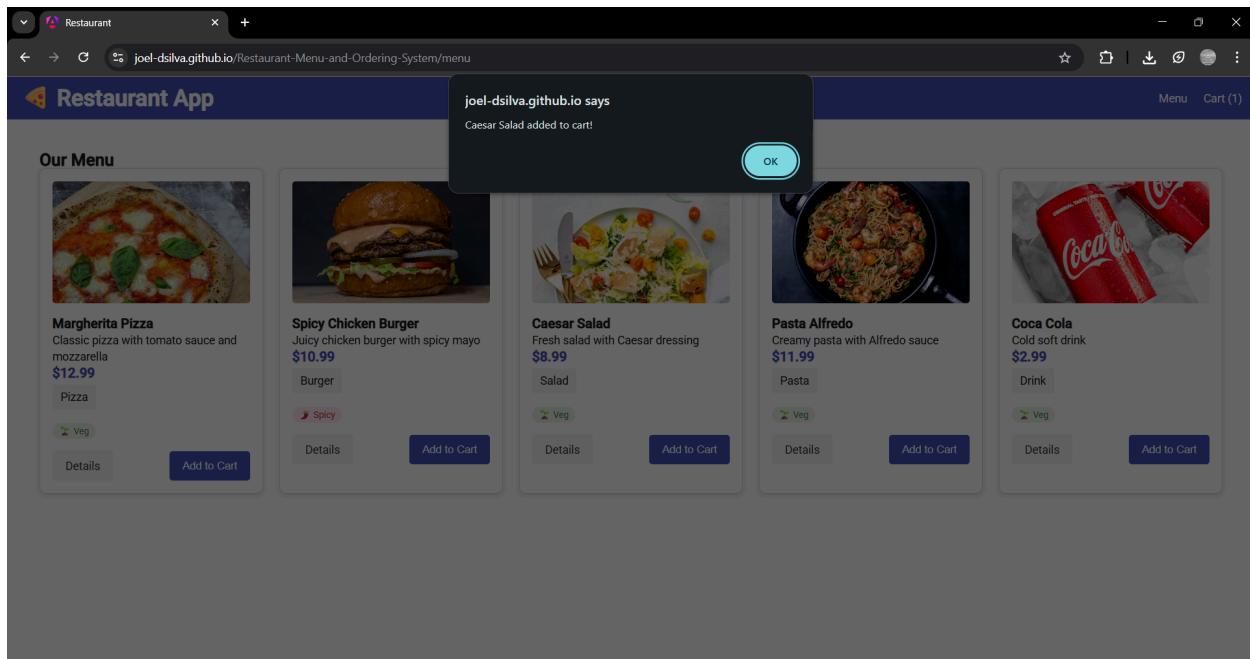


joel-dsilva.github.io says
Margherita Pizza added to cart!

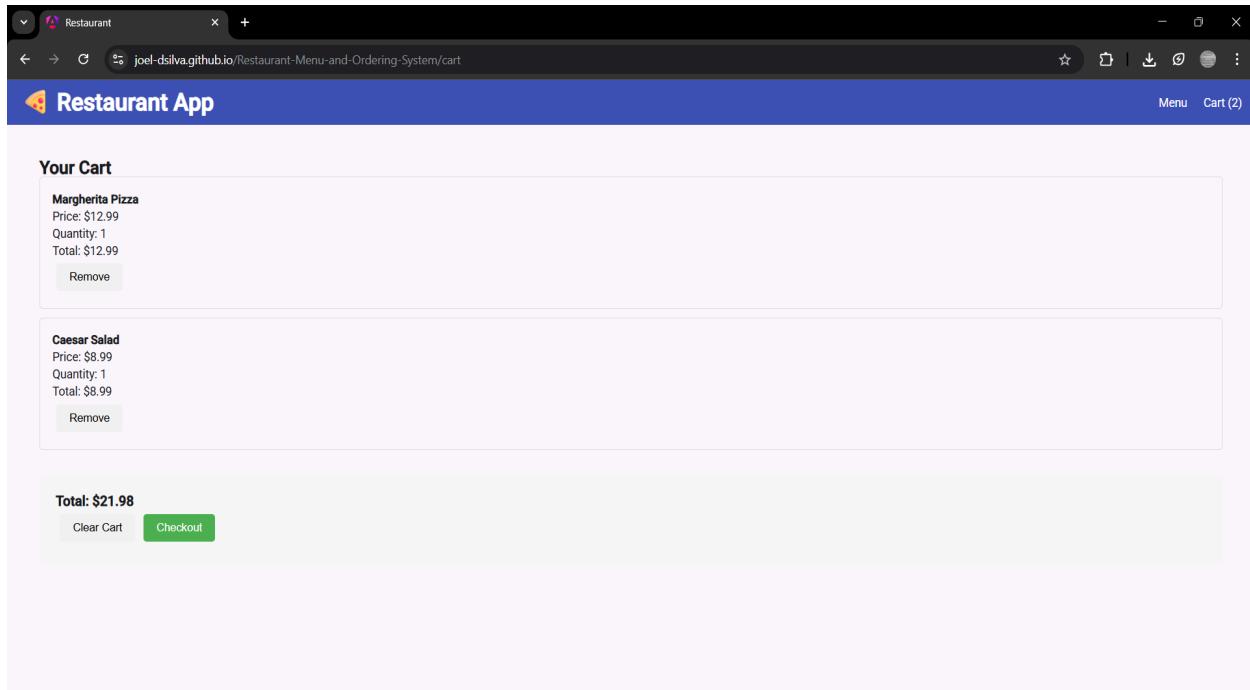
OK

Our Menu

- Margherita Pizza**
Classic pizza with tomato sauce and mozzarella
\$12.99
Pizza
Veg
[Details](#) [Add to Cart](#)
- Spicy Chicken Burger**
Juicy chicken burger with spicy mayo
\$10.99
Burger
Spicy
[Details](#) [Add to Cart](#)
- Caesar Salad**
Fresh salad with Caesar dressing
\$8.99
Salad
Veg
[Details](#) [Add to Cart](#)
- Pasta Alfredo**
Creamy pasta with Alfredo sauce
\$11.99
Pasta
Veg
[Details](#) [Add to Cart](#)
- Coca Cola**
Cold soft drink
\$2.99
Drink
Veg
[Details](#) [Add to Cart](#)



Cart:



A screenshot of a web browser displaying the cart page of the restaurant app. The cart contains two items: Margherita Pizza and Caesar Salad. The total price is \$21.98. Buttons for "Clear Cart" and "Checkout" are at the bottom. The browser address bar shows the URL: joel-dsilva.github.io/Restaurant-Menu-and-Ordering-System/cart.

Item	Price	Quantity	Total
Margherita Pizza	\$12.99	1	\$12.99
Caesar Salad	\$8.99	1	\$8.99
Total:	\$21.98		



Checkout:

A screenshot of a web browser displaying the 'Restaurant App' checkout page. The URL in the address bar is 'joel-dsilva.github.io/Restaurant-Menu-and-Ordering-System/checkout'. The page has a blue header with the app's logo and a navigation bar with 'Menu' and 'Cart (2)'. The main content area is titled 'Checkout' and contains an 'Order Summary' section with items: 'Margherita Pizza x 1 = \$12.99' and 'Caesar Salad x 1 = \$8.99', totaling '\$21.98'. Below this is a 'Customer Information' section with fields for Name, Email, and Phone, each with an input field. At the bottom are 'Back to Cart' and 'Place Order' buttons.

Checkout

Order Summary

Margherita Pizza x 1 = \$12.99

Caesar Salad x 1 = \$8.99

Total: \$21.98

Customer Information

Name:

Email:

Phone:

[Back to Cart](#) [Place Order](#)

A screenshot of the same web browser showing the checkout page after filling out the customer information. The 'Name' field now contains 'Joel DSilva', the 'Email' field contains 'joeldsilva2006@gmail.com', and the 'Phone' field contains '9175761792'. The rest of the page remains the same, including the order summary and buttons.

Checkout

Order Summary

Margherita Pizza x 1 = \$12.99

Caesar Salad x 1 = \$8.99

Total: \$21.98

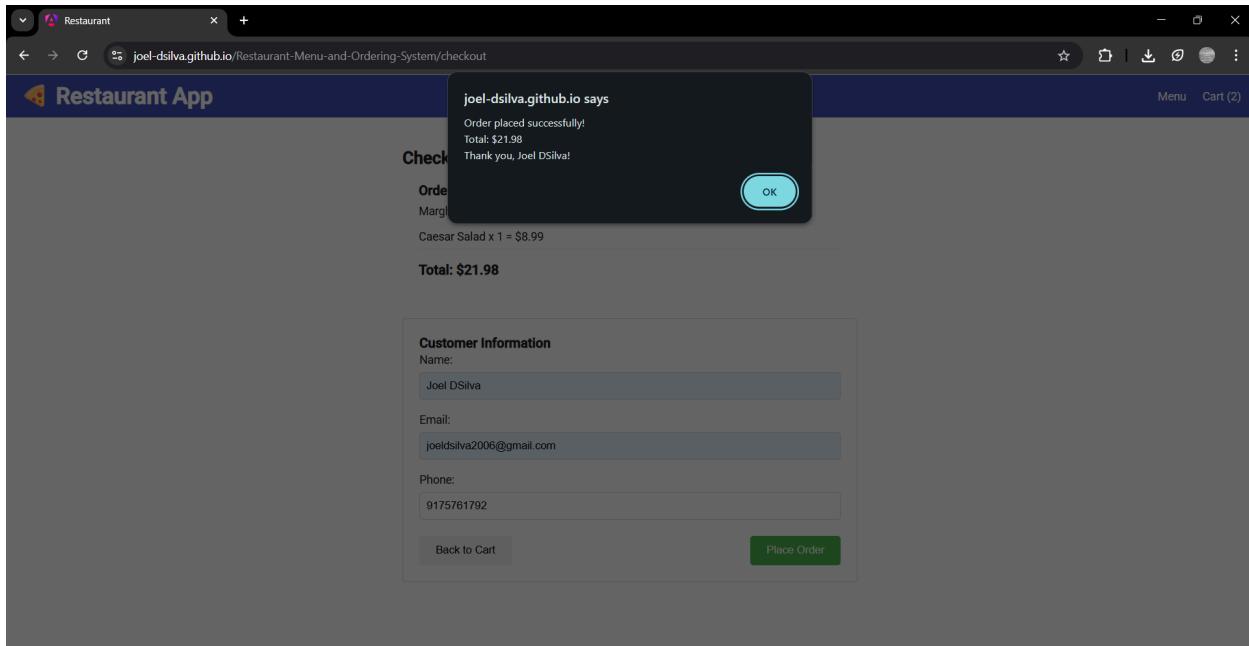
Customer Information

Name:

Email:

Phone:

[Back to Cart](#) [Place Order](#)



11. Conclusion

The Restaurant Menu and Ordering System demonstrates the practical application of Angular in building a real-world web application. This project enhanced our understanding of component-based development, routing, state management, and responsive UI design. It serves as a strong foundation for developing scalable web applications using modern JavaScript frameworks.

12. References

- L&T LMS : <https://learn.lntedutech.com/Landing/MyCourse>
- Angular Documentation – <https://angular.io>