



FRESH CART

DEVELOPED USING ANGULAR, BOOTSTRAP, AND REST API INTEGRATION

PRESENTED BY:

MAHMOUD ISMAIL

PROJECT OVERVIEW

Title: Fresh Cart

Content:

- The project is an e-commerce web application that allows users to browse, search, and purchase products online.
- It was developed using Angular as the front-end framework.
- Integrated Bootstrap for responsive design and UI components.
- Connected to an external REST API to fetch product data dynamically.

FEATURES AND FUNCTIONALITY

- **Product Listing:** Dynamic product cards fetched from the API, categorized by product type (Electronics, Clothes, Shoes, etc..).
- **Search & Filter:** Users can search for products or filter by category and price.
- **Product Details:** Each product has a dedicated page with detailed information, including price, description, and availability.
- **Shopping Cart & Checkout:** Users can add products to the cart, update quantities, and proceed to checkout.

TECHNOLOGIES USED

- **Angular (Framework):** For building reusable components, routing, and state management.
- **Typescript (Programming Language):** For clean, scalable, and type-safe code.
- **Bootstrap (Styling):** Ensured mobile-first design and responsive layout.
- **Route API (Integration):** Used for fetching product data and handling backend communication.
- **RXJS (Observables):** Managed asynchronous data fetching and real-time updates.
- **HTML/CSS:** Structured the web pages and styled the interface.

API INTEGRATION

API INTEGRATION FOR DYNAMIC DATA

- Fresh Cart fetches live product data from Route API
- Utilized Angular's **HttpClientModule** for API communication.
- Implemented **GET** requests for retrieving product listings and **POST** for submitting orders.
- **Error handling:** Managed API errors with Interceptors.

KEY COMPONENTS

- **Product Component:** Displays product information and integrates with the cart system.
- **Cart Component:** Manages cart items, allows for updating quantities, and shows a total price.
- **Checkout Component:** Handles order submission and validation before proceeding with the payment.
- **Search Component:** Enables real-time searching and filtering of products using observables

CHALLENGES FACED

- **Data Fetching Performance:** Ensuring efficient product loading with large datasets.
- **Solution:** Used lazy loading and pagination for better performance.
- **State Management:** Managing the shopping cart state across different components.
- **Solution:** Implemented Angular Services and Behavior Subjects to maintain state.

Responsive Design: Making sure the app looked consistent on various screen sizes.

Solution: Applied Bootstrap's grid system and media queries for full responsiveness.

FUTURE ENHANCEMENTS

- **Payment Gateway:** resolve the Bug in payment methods like **Stripe** or **PayPal**.
- **Product Reviews:** Enable customers to leave reviews and rate products.
- **Admin Dashboard:** Develop an admin panel to add/edit products and track sales.
- **Whish List:** Making a wish list component