Product App - Step-by-Step Documentation

1. Introduction

The **Product App** is a simple web application built using **Angular**. It displays a list of products, allows users to view product details, and provides navigation between pages.

2. Features

- **View a list of products** on the homepage.
- **Click on a product** to see its details.
- **Navigate between pages** using Angular routing.
- **Use a service** to fetch product data.

3. How Project Works?

- The app starts with the **Product List page** (/products).
- It loads products from a **JSON file**.
- Each product is displayed in a **card** with a 'View Details' button.
- Clicking "**VIew Details**" opens a **Product Details Page** (/products/:id).
- A **Back button** lets users return to the list.

4. Folder Structure

```
/product-app
```

```
├── src/
│   ├── app/                → Main application folder
│   │   ├── components/     → Small reusable UI parts
│   │   │   ├── product-card/
│   │   │   │   ├── product-card.component.ts
│   │   │   │   ├── product-card.component.html
│   │   │   │   ├── product-card.component.css
│   │   ├── pages/          → Main pages
│   │   │   ├── product-list/
│   │   │   │   ├── product-list.component.ts
│   │   │   │   ├── product-list.component.html
│   │   │   │   ├── product-list.component.css
│   │   │   ├── product-detail/
│   │   │   │   ├── product-detail.component.ts
│   │   │   │   ├── product-detail.component.html
│   │   │   │   ├── product-detail.component.css
│   │   ├── services/       → Handles product data
│   │   │   ├── product.service.ts
│   │   ├── app.routes.ts → Handles page navigation
│   │   ├── app.component.ts    → Main module that connects
everything
├── angular.json            → Angular configuration
├── package.json            → Lists required libraries
```

5. Implementation Details

5.1 Product Model

To ensure consistent product data, we define a Product model. This file defines a Product structure so TypeScript can check for errors.

File: `product.model.ts`

```
export interface Product {
  id: number;
  name: string;
  description: string;
  price: number;
}
```

5.2 Product Service

The ProductService fetches product data defined in the Products in Json format. The 2 methods

GetProducts() --> Returns all products.

GetProductById(id) --> Returns a single product based on ID.

File: `product.service.ts`

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

import { Observable, of } from 'rxjs';

import { Product } from '../models/product.model';

@Injectable({

  providedIn: 'root'

})
```

```
export class ProductService {

 constructor() { }

 private products: Product[] =
 [
  {
   "id": 1,
   "name": "Laptop",
   "description": "High-performance laptop with the latest processor.",
   "price": 999.99
  },
  {
   "id": 2,
   "name": "Smartphone",
   "description": "Latest smartphone with an advanced camera and 5G support.",
   "price": 699.99
  },
  {
   "id": 3,
   "name": "Wireless Headphones",
   "description": "Noise-canceling wireless headphones with long battery life.",
```

```
    "price": 199.99
  },
  {
    "id": 4,
    "name": "Smartwatch",
    "description": "Feature-rich smartwatch with health tracking and notifications.",
    "price": 149.99
  },
  {
    "id": 5,
    "name": "Gaming Mouse",
    "description": "Ergonomic gaming mouse with customizable RGB lighting.",
    "price": 49.99
  },
  {
    "id": 6,
    "name": "Mechanical Keyboard",
    "description": "RGB mechanical keyboard with tactile switches.",
    "price": 89.99
  },
  {
    "id": 7,
```

```
    "name": "4K Monitor",
    "description": "Ultra HD 4K monitor with a high refresh rate.",
    "price": 399.99
  },
  {
    "id": 8,
    "name": "Bluetooth Speaker",
    "description": "Portable Bluetooth speaker with deep bass and waterproof design.",
    "price": 129.99
  },
  {
    "id": 9,
    "name": "External Hard Drive",
    "description": "2TB external hard drive for secure and fast data storage.",
    "price": 119.99
  },
  {
    "id": 10,
    "name": "Webcam",
    "description": "1080p HD webcam with built-in microphone for video calls.",
    "price": 79.99
  }
```

```
  ];

  getProducts(): Observable<Product[]> {
    return of(this.products);
  }

  getProductById(id: number): Observable<Product | undefined> {
    return of(this.products.find(product => product.id === id));
  }
}
```

5.3 Product List Component

This displays all products from the service. Calls ProductService to fetch all products and store them in products.

File: `product-list.component.ts`

```
import { Component, OnInit } from '@angular/core';
import { ProductService } from
'../../services/product.service';
import { Product } from '../../models/product.model';

@Component({
  selector: 'app-product-list',
  templateUrl: './product-list.component.html',
  styleUrls: ['./product-list.component.css']
})
```

```
export class ProductListComponent implements OnInit {
  products: Product[] = [];

  constructor(private productService: ProductService) {}

  ngOnInit(): void {
    this.productService.getProducts().subscribe(data => {
      this.products = data;
    });
  }
}
```

File: product-list.component.html

```
<div class="container mt-4">

    <h2 class="text-center">Product List</h2>

    <div class="row">

      <div class="col-md-4" *ngFor="let product of
products">

        <br/>

        <app-product-card [product]="product"
(addToCart)="navigatetoproduct(product.id)" ></app-product-
card>



      </div>

    </div>

  </div>
```

Uses *ngFor to loop through products and display each one.

5.4 Product Card Component

File: product-card.component.ts

```
import { Component, Input, Output, EventEmitter } from
'@angular/core';
import { Product } from '../../models/product.model';

@Component({
  selector: 'app-product-card',
  templateUrl: './product-card.component.html',
  styleUrls: ['./product-card.component.css']
})
export class ProductCardComponent {
  @Input() product!: Product;
  @Output() viewDetailsEvent = new EventEmitter<number>();

  viewDetails(): void {
    this.viewDetailsEvent.emit(this.product.id);
  }
}
```

Displays Product details (name, Price). Emits an event when “View Details” is clicked.

5.5 Product Detail Component

File: product-detail.component.ts

```
import { Component, OnInit } from '@angular/core';
import { ActivatedRoute, Router } from '@angular/router';
import { ProductService } from
'../../services/product.service';
import { Product } from '../../models/product.model';

@Component({
  selector: 'app-product-detail',
  templateUrl: './product-detail.component.html',
  styleUrls: ['./product-detail.component.css']
})
export class ProductDetailComponent implements OnInit {
  product?: Product;

  constructor(
    private route: ActivatedRoute,
    private router: Router,
    private productService: ProductService
  ) {}

  ngOnInit(): void {
    const id =
Number(this.route.snapshot.paramMap.get('id'));
    this.productService.getProductById(id).subscribe(product
=> {
      this.product = product;
    });
  }

  goBack(): void {
    this.router.navigate(['/products']);
```

```
  }
}
```

6. Routing Setup

File: `app.routes.ts`

```
import { Routes } from '@angular/router';
import { ProductListComponent } from './pages/product-list/product-list.component';
import { ProductDetailComponent } from './pages/product-detail/product-detail.component';

export const routes: Routes = [
  { path: 'products', component: ProductListComponent },
  { path: 'products/:id', component: ProductDetailComponent },
  { path: '', redirectTo: 'products' }
];
```

The app allows users to move between pages using Routing (page navigation).

/products --> Shows the product list.

/products/:id --> Shows product details based on ID.

Default(‘ ‘) redirects to /products

7. Conclusion

This project demonstrates: ✅ Component-based development. ✅ Routing for navigation. ✅ Services for data fetching.