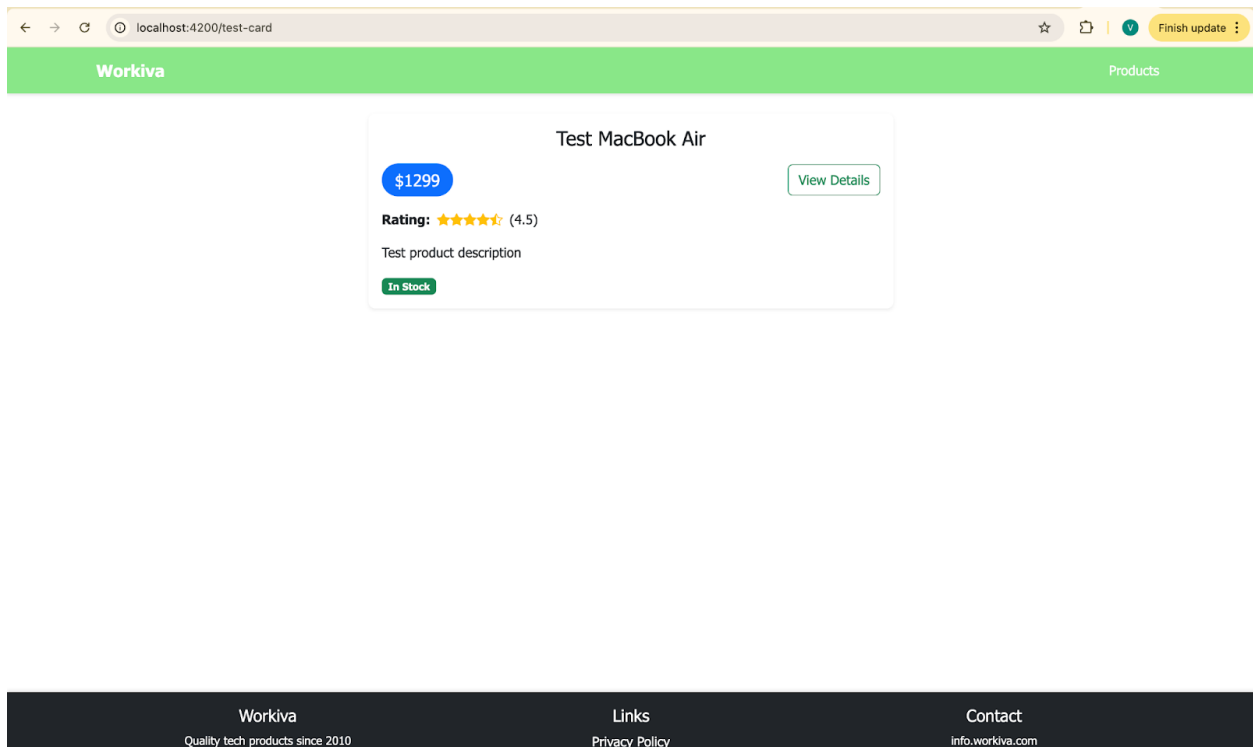# Application Manual

The Developed UI consists of three routes

1. /test-card
2. /products
3. /products/:id

Detailed Explanation of Routes as follows:

Route 1: /test-card

This route is implemented to test Step 2 mentioned in the task. @Input and @Output are used to accept the product object and emit an event that shows the details.
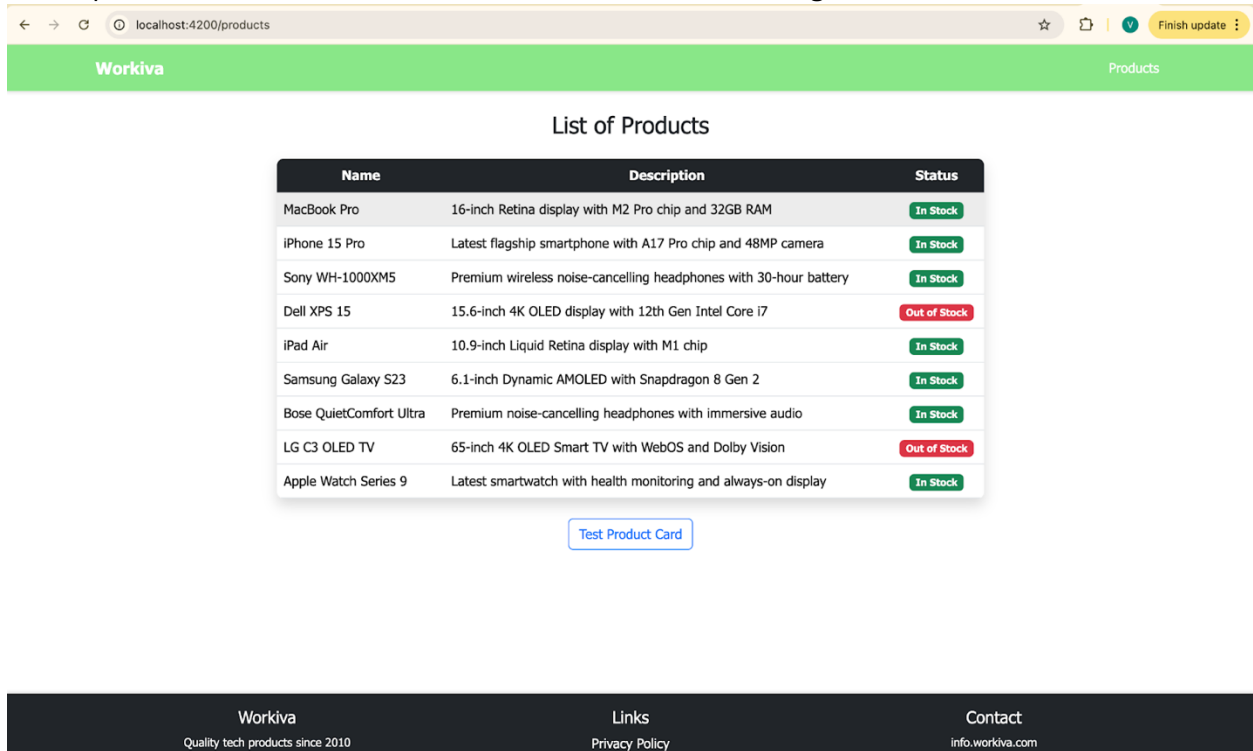

On clicking the view details button, an alert event is triggered that shows the product id.

Route 2: /products

This route is implemented to show all the products present in the products.json file. This is present inside the assets folder of the application.

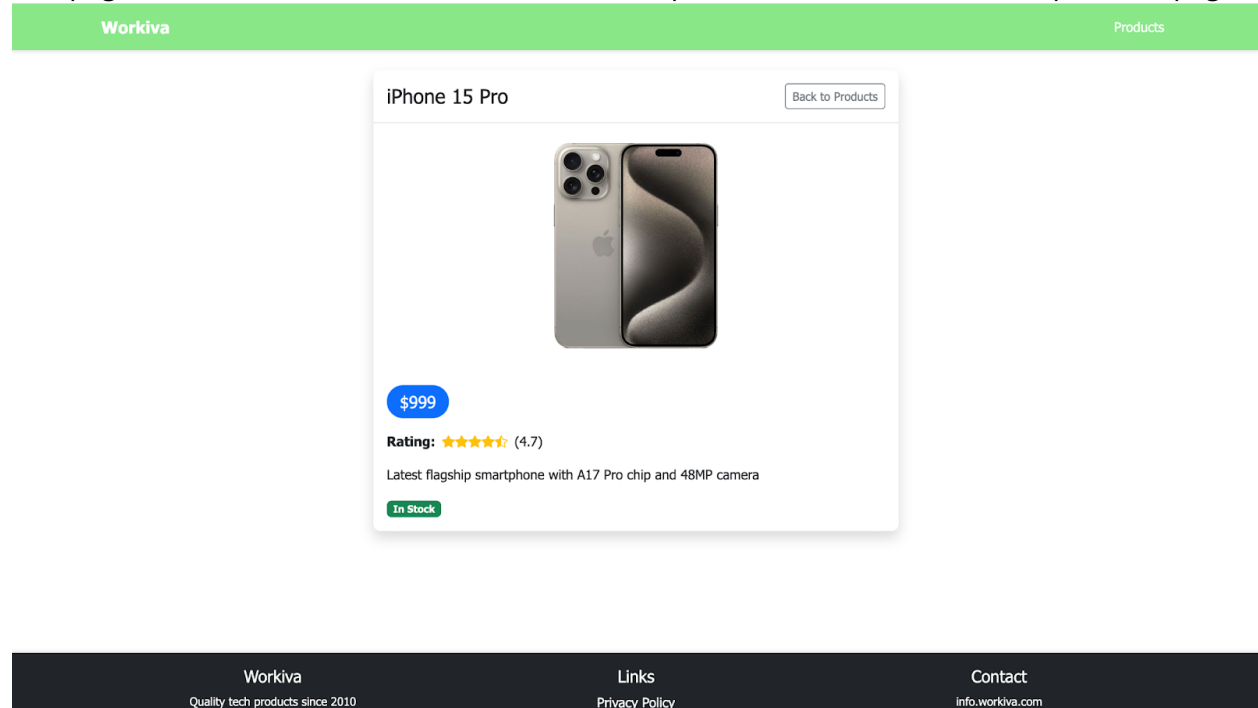All the products are listed in a table as rows as shown in the image below.

The Test Product Card redirects to testing the product card component.
Each row in the table is clickable and on clicking a product it redirects to the /products/:id page where the id is automatically fetched based on the product selected.
The Products hyperlink present in the top bar can always be used to navigate back to the product page.

Route 3: /products/:id

This page consists of detailed information about the product that is selected in the products page.

Technical Specifications

Since the application is developed on the angular base, there are components for every route involved and a model for defining the basic structure.

Model – Product.model

This forms the basic template for the product. It contains the following fields:

Id: Unique identifier of the product

Name: Name of the Product

Price: Price of the Product

Description: Basic Details of the product

Image: The image of the product,

rating: Average Rating received for the product

 inStock: Whether the product is in stock or not

This model is reused in all the components to display or manipulate the data surrounding the product.

Components

The component is basic building block of the application. We have three main components namely

➔ Product-card: To test the working of the product
➔ Product-list: To list all the products present in the inventory
➔ Product-detail: To show all the details of selected product

Assets

This contains all the miscellaneous items required for functioning of the UI such as:

➔ Images for displaying the products
➔ The product.json which is the inventory that stores the current products