

S MUHAMMAD USMAN temp 2 furniture 22 days left in trial

temp 2 furniture

PLAN STATUS PROJECT ID
Growth Trial Active 89i5a83u

Getting started Overview Members Studios Datasets Access Activity Usage Plan API Settings

Next steps

Initialize your project with the CLI

Run this command in your Terminal to continue setting up your project.

```
npm create sanity@latest -- --project 89i5a83u --dataset produ
```

Copy

Having issues with the CLI?

Usage

What's new

Sanity Create Content Mapping, Visual Editing, and Content Releases

131 / 250k
API Requests

View more →

Project members

View all →



Invite your first team member

+ Invite project members

Activity

View more →

05:54 PM

S MUHAMMAD USMAN temp 2 furniture 22 days left in trial

Getting started Overview Members Studios Datasets Access Activity Usage Plan API Settings

Webhooks

CORS origins

+ Add CORS origin

CORS origins Hosts that can connect to the project API.

ORIGIN	CREDENTIALS	CREATED	Actions
http://localhost:3001	Allowed	1 week	trash
http://localhost:3000	Allowed	1 week	trash
http://localhost:3333	Allowed	1 week	trash

Tokens

+ Add API token

What's new
Sanity Create Content Mapping, Visual Editing, and Content Releases

Tokens are used to authenticate apps and scripts to access project data.

NAME	PERMISSIONS	CREATED	Actions
tokentemp2	Editor	1 week	trash

S MUHAMMAD USMAN temp 2 furniture 22 days left in trial

Getting started Overview Members Studios Datasets Access Activity Usage Plan API Settings

http://localhost:3333 Allowed 1 week trash

Webhooks

CORS origins

Tokens

+ Add API token

What's new
RATIONAL

Tokens are used to authenticate apps and scripts to access project data.

NAME	PERMISSIONS	CREATED	Actions
tokentemp2	Editor	1 week	trash

The screenshot shows a VS Code interface with the following details:

- File Explorer (Left):** Shows the project structure under "OPEN EDITORS".
 - fweb**: .next, .vercel, Hackathon-migration, node_modules, public.
 - src**: app, components, lib.
 - sanity**: lib, schemaTypes
 - category.ts**
 - index.ts** (highlighted)
 - products.ts**
 - env.ts**
 - structure.ts**
 - .env.local**
 - .eslintrc.json**
 - .gitignore**
 - components.json**
- Editor (Center):** The file "index.ts" is open, showing a TypeScript code snippet for defining a Sanity schema.

```
1 import { type SchemaTypeDefinition } from "sanity";
2 import { Category } from "./category";
3 import { product } from "./products";
4
5
6 export const schema: { types: SchemaTypeDefinition[] } = {
7   types: [product, Category],
8 };
9
```
- Terminal (Bottom):** Displays the build logs for a Next.js 15.0.4 application using Turbopack.
 - ▲ Next.js 15.0.4 (Turbopack)
 - Local: http://localhost:3000
 - Environments: .env.local

✓ Starting...

✓ Ready in 4s

○ Compiling / ...

✓ Compiled / in 11s

GET / 200 in 13133ms

○ Compiling /productb ...

✓ Compiled /productb in 1619ms

GET /productb 200 in 1820ms

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

- Title Bar:** The title bar displays "figmaweb".
- File Explorer:** On the left, there is a sidebar with various icons representing different file types and project components.
- Code Editor:** The main area shows a TypeScript file named `products.ts`. The code defines a `product` type as a `document` with fields: `category` (a `reference` to another category), `name` (a required `string`), `slug` (a `slug`), and `title` (a required `string`). The code uses `defineField` and `defineType` from the `sanity` package.
- Status Bar:** The bottom status bar shows "main" and "Initializing 'fweb/tsconfig.json'". It also includes standard status indicators like battery level, signal strength, and temperature (19°C).

```
ts products.ts X
fweb > src > sanity > schemaTypes > ts products.ts > ...
1 import { defineField, defineType } from "sanity";
2
3 export const product = defineType({
4   name: "product",
5   title: "Product",
6   type: "document",
7   fields: [
8     defineField({
9       name: "category",
10      title: "Category",
11      type: "reference",
12      to: [
13        {
14          type: "category",
15        },
16      ],
17    }),
18    defineField({
19      name: "name",
20      title: "Title",
21      validation: (rule) => rule.required(),
22      type: "string",
23    }),
24    defineField({
25      name: "slug",
26      title: "Slug",
27      validation: (rule) => rule.required(),
28      type: "slug",
29    }),
30    defineField({
```

```
ts products.ts X
fweb > src > sanity > schemaTypes > ts products.ts > ...
3   export const product = defineType({
7     fields: [
11       defineField({
12         name: "image",
13         type: "image",
14         validation: (rule) => rule.required(),
15         title: "Product Image",
16       }),
17       defineField({
18         name: "price",
19         type: "number",
20         validation: (rule) => rule.required(),
21         title: "Price",
22       }),
23       defineField({
24         name: "quantity",
25         title: "Quantity",
26         type: "number",
27         validation: (rule) => rule.min(0),
28       }),
29       defineField({
30         name: "tags",
31         type: "array",
32         title: "Tags",
33         of: [
34           {
35             type: "string",
36           },
37         ],
38       },
39     ],
40   });
41 
```

Ln 1, Col 1 Spaces:2 UTF-8 CRLF () TypeScript ⓘ Go Live ⓘ Prettier

```
ts category.ts X
fweb > src > sanity > schemaTypes > ts category.ts > ...
1   import { defineType, defineField } from "sanity";
2
3   export const Category = defineType({
4     name: "category",
5     title: "Category",
6     type: "document",
7     fields: [
8       defineField({
9         name: "name",
10        title: "Name",
11        type: "string",
12        validation: (rule) => rule.required(),
13      }),
14       defineField({
15         name: "slug",
16         title: "Slug",
17         type: "slug",
18         validation: (rule) => rule.required(),
19         options: {
20           source: "name",
21         },
22       }),
23     ],
24   });
25 
```

Ln 1, Col 1 Spaces:2 UTF-8 CRLF () TypeScript ⓘ Go Live ⓘ Prettier

S Default + Create Search Structure

Content

- Product
- Category

Product

- Marble Ease
- Vase Set
- The Dandy chair
- Bright Space
- Cloud Haven Chair
- Bold Nest
- Sunny Chic
- Reflective Haven

localhost:3000/studio/structure/product

S Default + Create Search Structure Vision

Content

- Product
- Category

Category

- Tableware
- Chairs
- Plant Pots
- Ceramics
- Tables
- Cutlery
- Crockery

localhost:3000/studio/structure/category

localhost:3000/studio/vision

S Default + Create Search Structure Vision Schedules

DATASET API VERSION CUSTOM API VERSION PERSPECTIVE QUERY URL [COPY TO CLIPBOARD]

production Other v2025-01-17 raw https://89i5a83u.api.sanity.io/v2025-01-17/data

QUERY

```
1 *[_type == "product"]{  
2   _id,  
3   name,  
4   slug,  
5   price,  
6   quantity,  
7   tags,
```

PARAMS

```
{  
}
```

RESULT

```
[...] 25 items  
  0: {...} 12 properties  
    features: null  
    dimensions: null  
    categoryName: null  
    categorySlug: null  
    image: null  
    slug: null  
    description: null  
    price: null  
    quantity: null  
    tags: null  
    _id: drafts.b79618ab-b227-43ba-9766-598b6883be3b
```

Execution: 14ms End-to-end: 639ms Save result as JSON

The screenshot shows the VS Code interface with the following details:

- File Explorer:** Shows the project structure under "OPEN EDITORS". The current file, "DataFetch.tsx", is selected.
- Editor:** Displays the code for "DataFetch.tsx". The code uses the DatafetchClient module to fetch products from a database.
- Bottom Bar:** Includes status icons like "main", "0", "0", "0", and "0", and a set of icons for file operations.
- Bottom Status:** Shows "Ln 1, Col 1" and other settings like "Spaces: 2", "UTF-8", "CRLF", "TypeScript JSX", "Go Live", and "Prettier".

```
const DataFetch = async () => {
  const query = await client.fetch(
    `*[_type == "product"]{
      _id,
      name,
      slug,
      price,
      quantity,
      tags,
      description,
      features,
      dimensions,
      "categoryName": category->name,
      "categorySlug": category->slug.current,
      image {
        asset -> {
          url
        }
      }
    }`;
  return <DatafetchClient products={query} />;
};
```

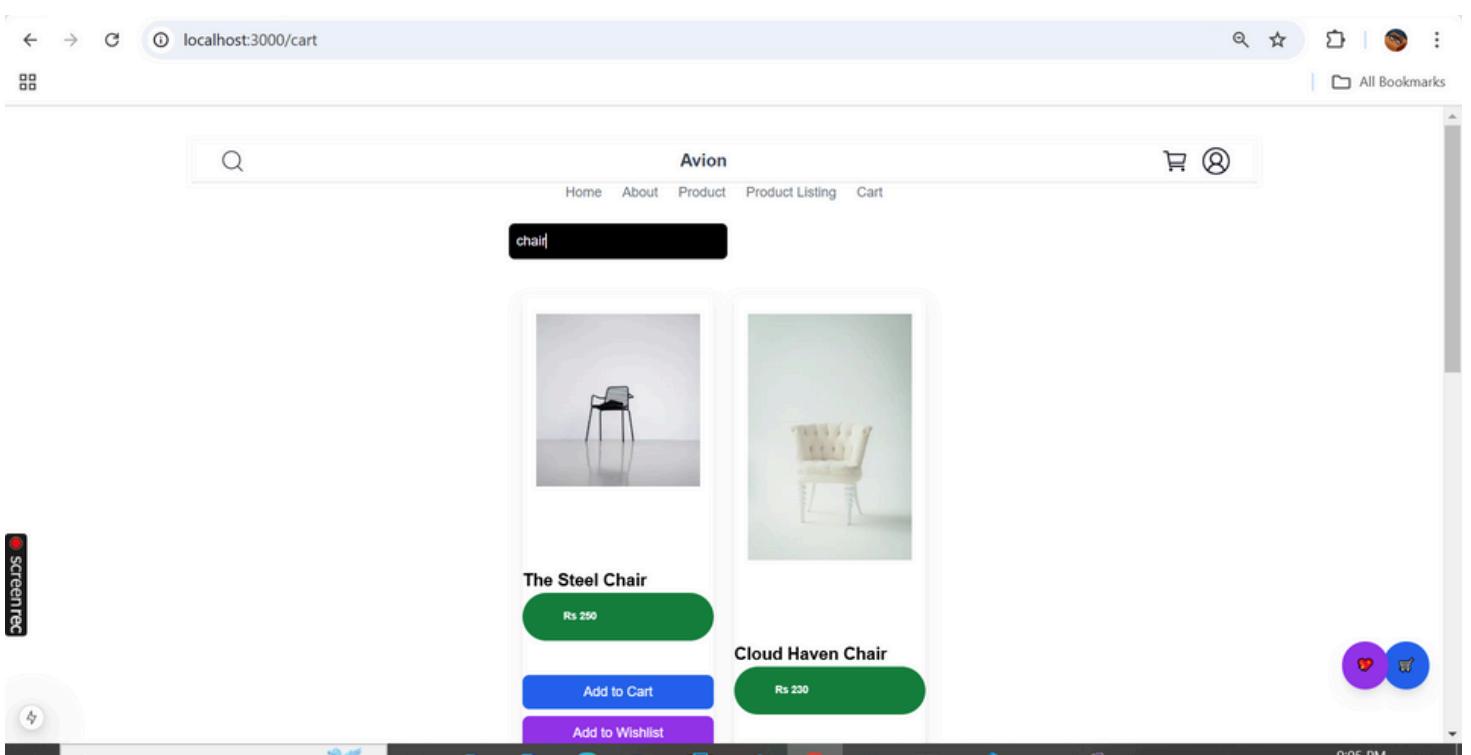
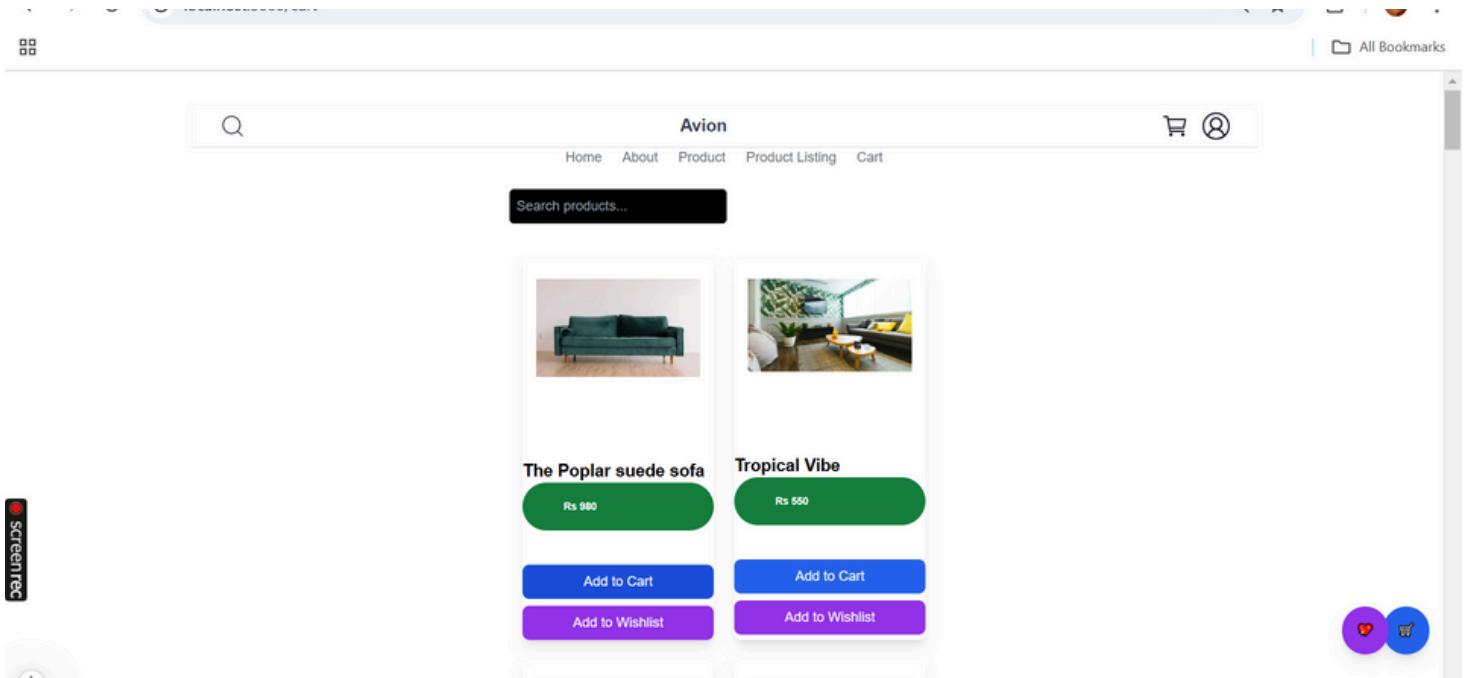
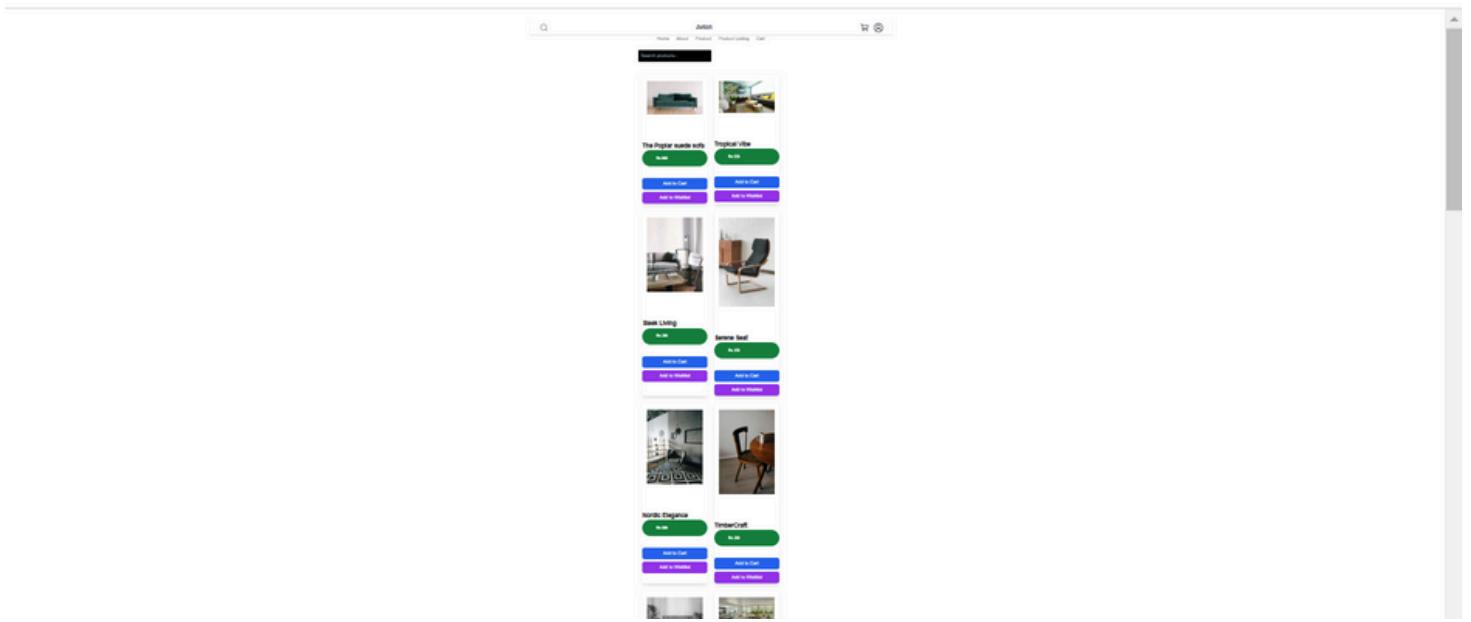
The screenshot shows the VS Code interface with the following details:

- File Explorer:** Shows the project structure under "OPEN EDITORS". The current file, "DataFetchClient.tsx", is selected.
- Editor:** Displays the code for "DataFetchClient.tsx". It contains utility functions for managing a shopping cart and wishlist.
- Bottom Bar:** Includes status icons like "main", "0", "0", "0", and "0", and a set of icons for file operations.
- Bottom Status:** Shows "Ln 13, Col 13" and other settings like "Spaces: 2", "UTF-8", "CRLF", "TypeScript JSX", "Go Live", and "Prettier".

```
const DataFetchClient = ({ products }: { products: Product[] }) => {
  const removeFromCart = (cartId: string) => {
    setCart((prevCart) => {
      const updatedCart = prevCart.filter((item) => item.cartId !== cartId);
      localStorage.setItem("Cart", JSON.stringify(updatedCart));
      return updatedCart;
    });
  };

  const updateQuantity = (cartId: string, increment: boolean) => {
    setCart((prevCart) => {
      const updatedCart = prevCart.map((item) => {
        if (item.cartId === cartId) {
          const newQuantity = increment ? item.quantity + 1 : item.quantity - 1;
          return { ...item, quantity: Math.max(newQuantity, 1) };
        }
        return item;
      });
      localStorage.setItem("Cart", JSON.stringify(updatedCart));
      return updatedCart;
    });
  };

  const addToWishlist = (product: Product) => {
    setWishlist((prevWishlist) => {
      if (!prevWishlist.find((item) => item._id === product._id)) {
        const updatedWishlist = [...prevWishlist, product];
        localStorage.setItem("Wishlist", JSON.stringify(updatedWishlist));
        return updatedWishlist;
      }
    });
  };
};
```



localhost:3000/cart

Avion

Home About Product Product Listing Cart

Search products...

Your Wishlist

The Poplar suede sofa	Rs 980	Move to Cart	Remove
Tropical Vibe	Rs 550	Move to Cart	Remove

[Clear Wishlist](#)

Add to Cart Add to Wishlist

Add to Cart Add to Wishlist

Cart icon with 8 items

Screenrec recording indicator

Avion

Home About Product Product Listing Cart

Search products...

Your Cart

The Poplar suede sofa	- 2 +	Rs 1960	Remove
Tropical Vibe	- 1 +	Rs 550	Remove

Total: Rs 2510

[Confirm Order](#)

Add to Cart Add to Wishlist

Add to Cart Add to Wishlist

Cart icon with 8 items

Screenrec recording indicator

Checkout

Name:
usman

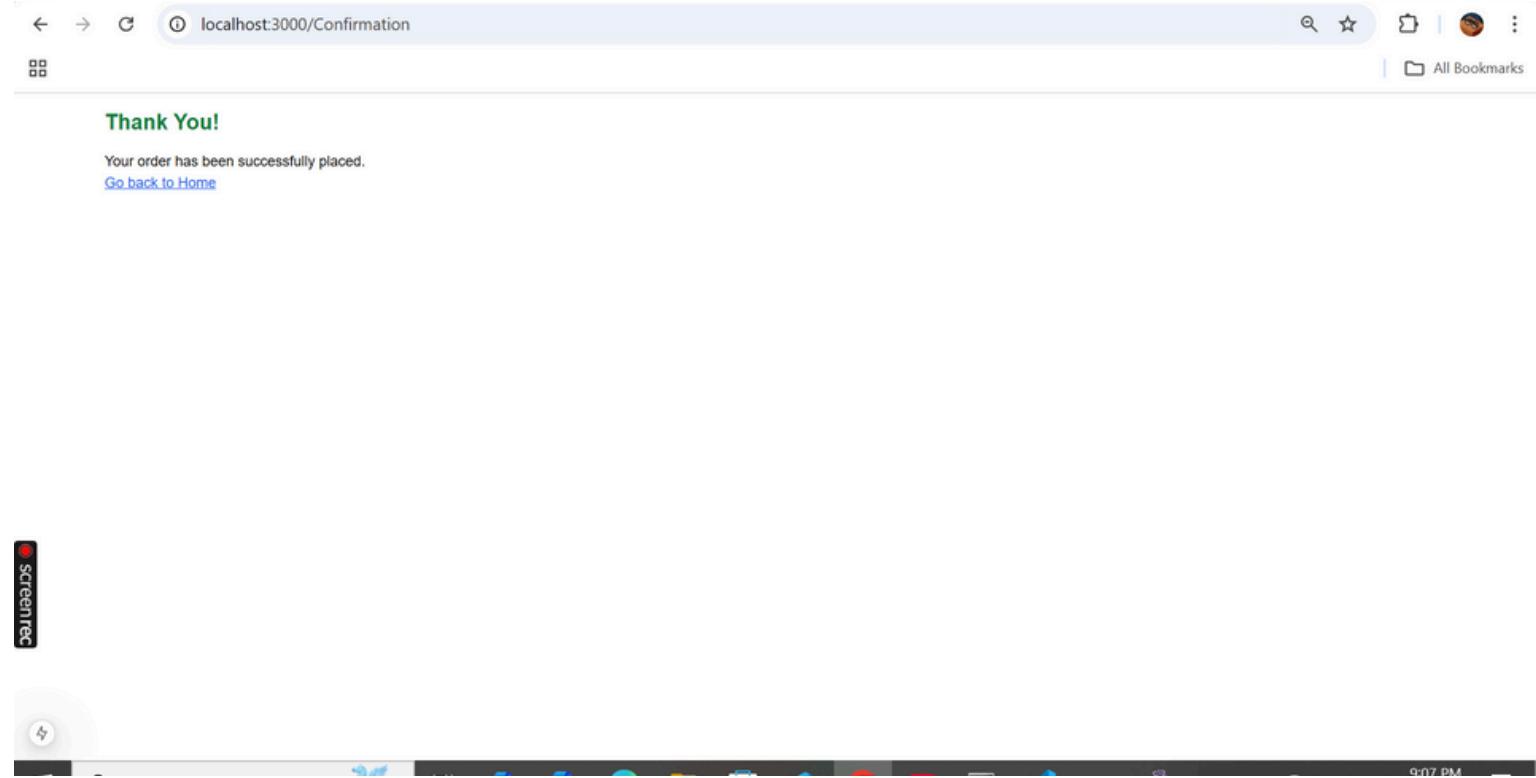
Address:
street 16 apex building house 03 ,Bihar Colony, Karachi City

Contact:
03082949058

Payment Method:
Credit Card

[Confirm Order](#)

Screenrec recording indicator



File Edit Selection View ... ← → figmaweb

EXPLORER OPEN EDITORS ProductListingclient.tsx

fweb > src > app > components > ProductListingclient.tsx > ...

```
1 "use client";
2 import React, { useState, useEffect } from "react";
3 import Data from "./images";
4 import Image from "next/image";
5
6 type Product = {
7   slug: string | number;
8   quantity: number;
9   features: string;
10  title: string;
11  _id: string;
12  imageUrl: string;
13  name: string;
14  price: number;
15  description: string;
16 };
17
18 const ProductListing: React.FC = () => {
19   const [products, setProducts] = useState<Product[]>([[]]);
20   const [selectedProduct, setSelectedProduct] = useState<Product | null>(null);
21   const [showDetail, setShowDetail] = useState(false);
22
23   useEffect(() => {
24     const fetchData = async () => {
25       const productData: Product[] = await Data();
26       setProducts(productData);
27     };
28     fetchData();
29   }, []);
30 }
```

here to search

Ln 1, Col 1 Spaces: 2 UTF-8 CRLF {} TypeScript JSX ⚡ Go Live ✅ Prettier

19°C

localhost:3000/productb

Avion

Home About Product Product Listing Cart

Product Listing

The Poplar suede sofa

Check Price
Check Stock
Check Features
Check Description

Tropical Vibe

Check Price
Check Stock
Check Features
Check Description

Sleek Living

screenrec

localhost:3000/productb

The Poplar suede sofa

Price: Rs. 980
Stock: 50
Features: Premium material Handmade upholster Quality timeless classic
Description: A timeless design, with premium materials features as one of our most popular and iconic pieces. The dandy chair is perfect for any stylish living space with beech legs and lambskin leather upholstery.

Close

screenrec

localhost:3000/productb

The Poplar suede sofa

Price: Rs. 980
Stock: 50
Features: Premium material Handmade upholster Quality timeless classic
Description: A timeless design, with premium materials features as one of our most popular and iconic pieces. The dandy chair is perfect for any stylish living space with beech legs and lambskin leather upholstery.

Close

screenrec

