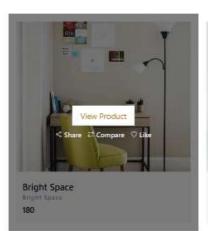


```
src > sanity > schemaTypes > TS product.ts > [❷] product > 🌽 fields
      import { defineType } from "sanity"
      export const product = defineType({
           name: "product",
           title: "Product",
           type: "document",
       😯 fields: [
  7
               {
                   name: "title",
                   title: "Title",
                   validation: (rule) => rule.required(),
 11
 12
                   type: "string"
 13
               },
 14
                   name: "description",
 15
                   type: "text",
                   validation: (rule) => rule.required(),
 17
                   title: "Description",
 18
 19
               },
                   name: "productImage",
 21
 22
                   type: "image",
                   validation: (rule) => rule.required(),
 23
                   title: "Product Image"
 25
               },
 27
                   name: "price",
                   type: "number",
                   validation: (rule) => rule.required(),
 29
                   title: "Price".
```

Our Product









Syltherine Syltherine 200



Bed 8ed 250



Retro Vibe Retro Viba 340



The Lucky Lamp
The Lucky Lamp
200



Pure Aura Fure Aura 280



Vase Set Vase Set 150

```
import Link from "next/link";
import Image from "next/image";
import { IoShareSocialOutline } from "react-icons/io5";
import { VscArrowSwap } from "react-icons/vsc";
import { IoHeartOutline } from "react-icons/io5";
import { useEffect, useState } from "react";
import { client } from "@/sanity/lib/client";
const SecThree = () => {
 const [products, setProducts] = useState([]);
  const [loading, setLoading] = useState(true); // Loading state
  // Fetching products from Sanity
  useEffect(() => {
    const fetchProducts = async () => {
      try {
        setLoading(true); // Start loading
        const data = await client.fetch() *[_type == "product"][0...12]{
         _id,
          title,
          description,
          price,
          "imageUrl": productImage.asset->url,
          oldPrice,
          discountTag
        } );
        setProducts(data);
      } catch (error) {
        console.error("Error fetching products:", error);
```