

localhost:3000/studio/vision

S

Default

+ Create

Q

StructureVisionSchedules

DATASET

production

API VERSION

Other

CUSTOM API VERSION

v2025-01-17

PERSPECTIVE

raw

QUERY URL [COPY TO CLIPBOARD]

https://silrkg44.api.sanity.io/v2025-01-17/data/query/prc

QUERY

1 *[_type == "product"]

PARAMS

1 {

2

3 }

RESULT

[...] 20 items

▼ 0: {...} 15 properties

price: 50

_id: 0e0ztuJY6NWdyb1BER9jJf

slug: {...} 2 properties

current: man-suit-4

_type: slug

_rev: BQj9k9W1Mf2GlcUg31ZTt

_type: product

priceWithoutDiscount: 100

name: Man Suit - 4

image: {...} 2 properties

_type: image

asset: {...} 2 properties

_type: reference

_ref: image-630f95192510cafdb03875309e3e6c991b9659f1-500x750-jpg

Fetch

Listen

Execution: 8ms End-to-end: 219ms

Save result as

JSON





CSV

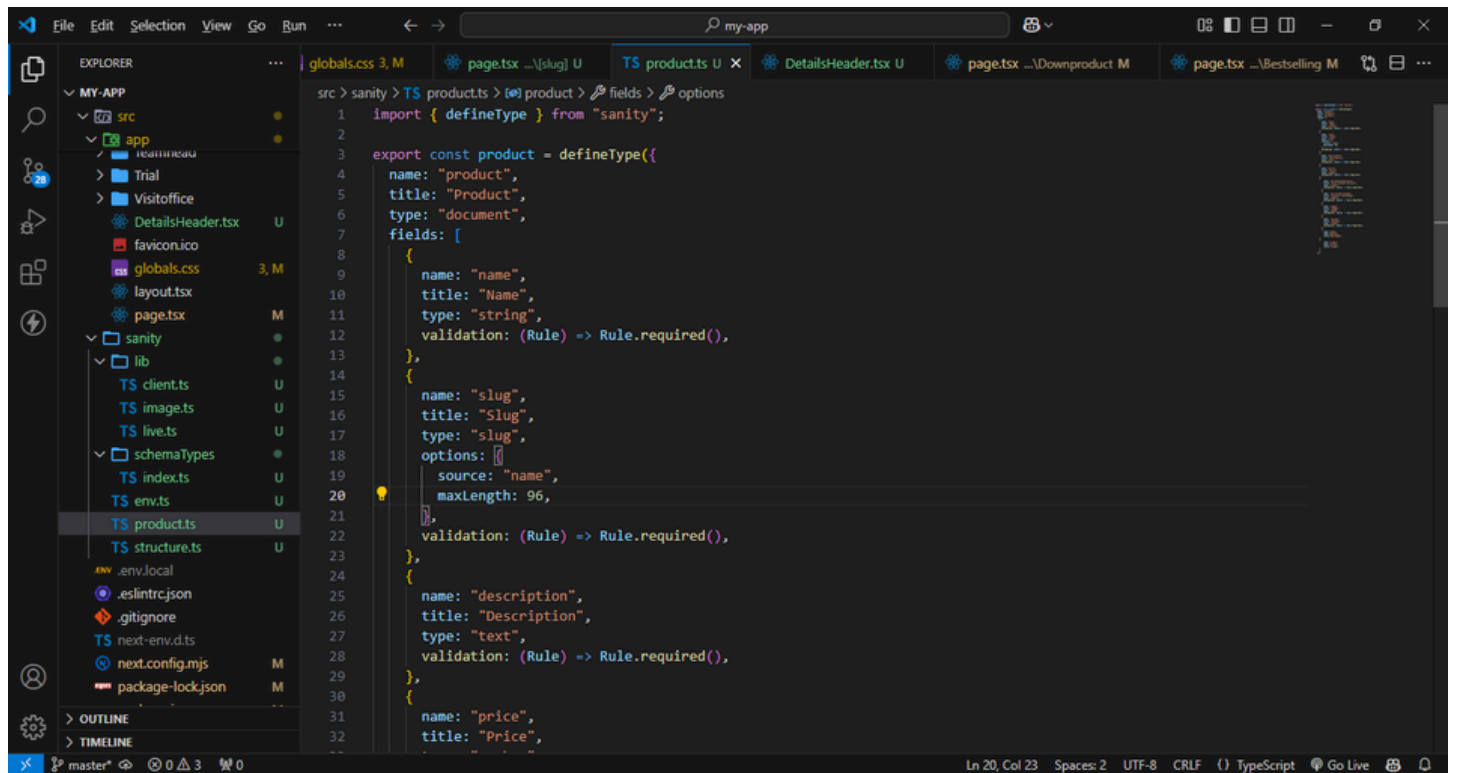
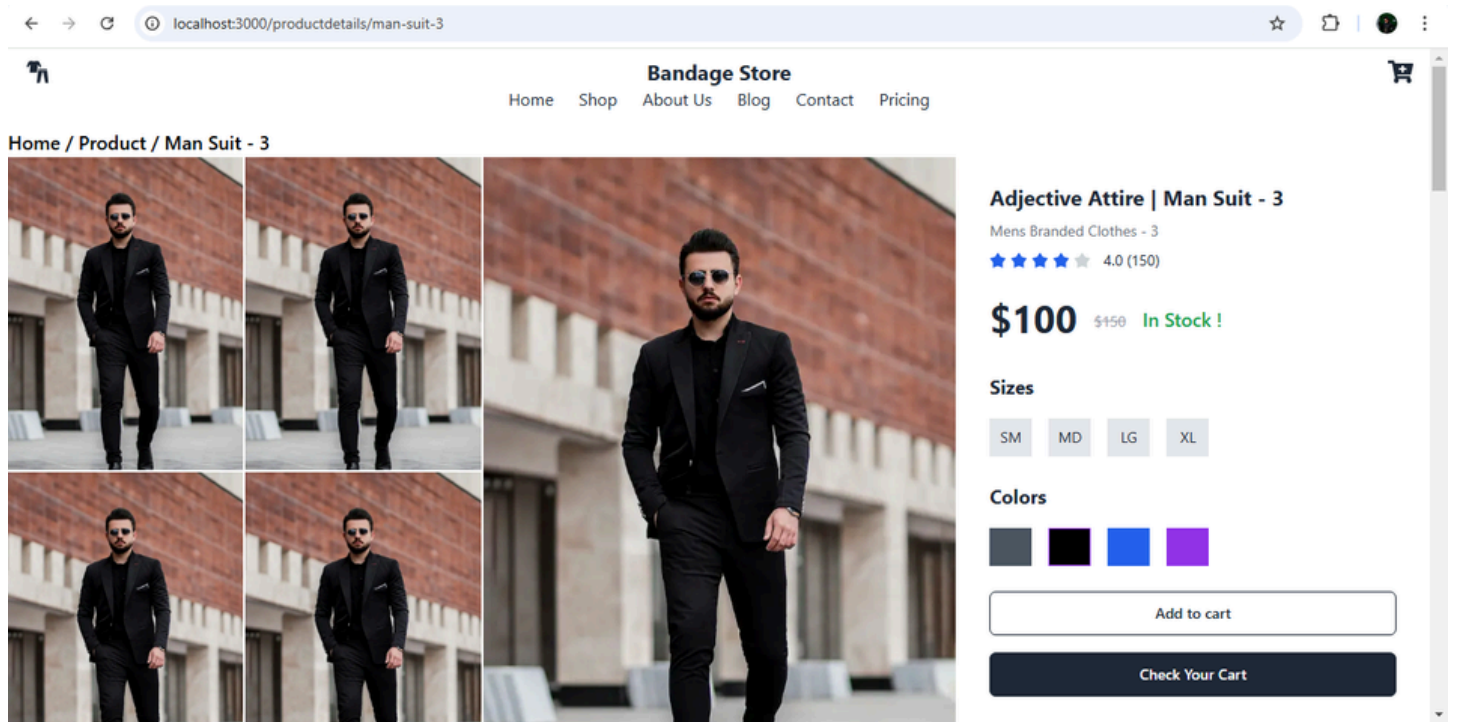
localhost:3000

Featured Products

Bestseller Products

Problems trying to resolve the conflict between





my-app

EXPLORER

- MY-APP
 - node_modules
 - public
 - scripts
 - importSanityData.mjs
 - src
 - app
 - About
 - Aboutbig
 - Aboutbigban
 - Aboutpage
 - Aboutparagraph
 - api
 - Bestselling
 - page.tsx
 - Bigcompanieshead
 - brandbanner
 - cart
 - Cartdescrip
 - components
 - Contact
 - Contacthero
 - downbanner
 - Downproduct
 - page.tsx
 - FaqSec
 - OUTLINE
 - TIMELINE

scripts > JS importSanityData.mjs > uploadImageToSanity

```
19 async function uploadImageToSanity(imageUrl) {
20   const buffer = Buffer.from(response.data);
21   const asset = await client.assets.upload('image', buffer, {
22     filename: imageUrl.split('/').pop()
23   });
24   console.log('Image uploaded successfully: ${asset.id}')
25   return asset._id
26 } catch (error) {}
27 console.error('Failed to upload image:', imageUrl, error)
28 return null
29 }
30
31 async function importData() {
32   try {
33     console.log('Fetching products from API...')
34     const response = await axios.get('https://template6-six.vercel.app/api/products')
35     const products = response.data
36     console.log('Fetched ${products.length} products')
37     for (const product of products) {
38       console.log('Processing product: ${product.title}')
39       let imageRef = null
40       if (product.image) {
41         imageRef = await uploadImageToSanity(product.image)
42       }
43       const sanityProduct = {
44         _type: 'product',
45         name: product.name,
46         description: product.description,
47         price: product.price,
48         discountPercentage: product.discountPercentage,
49         priceWithoutDiscount: product.priceWithoutDiscount,
50         rating: product.rating,
51         isNew: product.isNew
52       }
```

Ln 32, Col 4 Spaces: 2 UTF-8 CRLF () JavaScript Go Live

my-app

EXPLORER

- MY-APP
 - src
 - app
 - ShopSection
 - SimplePrice
 - SimplePricecontent
 - Simplepricetrust
 - studio
 - Team
 - Teamhead
 - Trial
 - Visitoffice
 - DetailsHeader.tsx
 - favicon.ico
 - globals.css
 - layout.tsx
 - page.tsx
 - sanity
 - lib
 - TS client.ts
 - TS image.ts
 - TS live.ts
 - schemaTypes
 - TS index.ts
 - TS env.ts
 - TS products.ts
 - OUTLINE
 - TIMELINE

src > app > page.tsx > Home

```
8 import Header from "../components/Header"
9 import { client } from "@sanity/lib/client"
10 import { useEffect, useState } from "react"
11 export default function Home() {
12   const [products, setProducts] = useState<any[]>([]);
13
14   const fetchData = async () => {
15     const query = `[_type == "product"] | order(_createdAt asc){
16       name,
17       description,
18       price,
19       priceWithoutDiscount,
20       discountPercentage,
21       image,
22       rating,
23       slogan,
24       isNew,
25       "slug":slug.current
26     }`
27     const data = await client.fetch(query);
28     setProducts(data);
29   }
30
31   useEffect(() => {
32     fetchData();
33   }, []);
34
35   return(
36     <div>
37       <Header/>
38     </div>
39   )
40 }
```

Ln 51, Col 2 Spaces: 2 UTF-8 LF () TypeScript JSX Go Live