

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

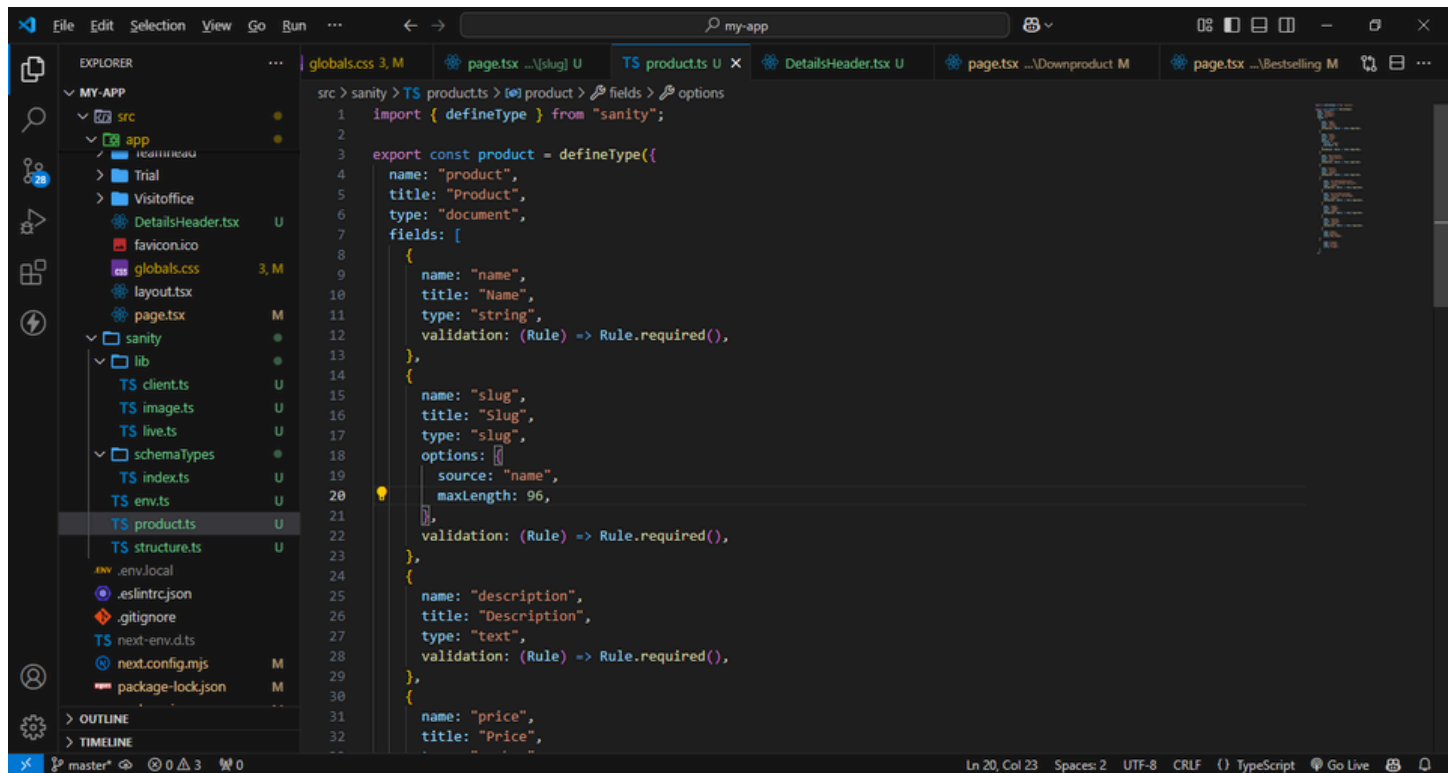
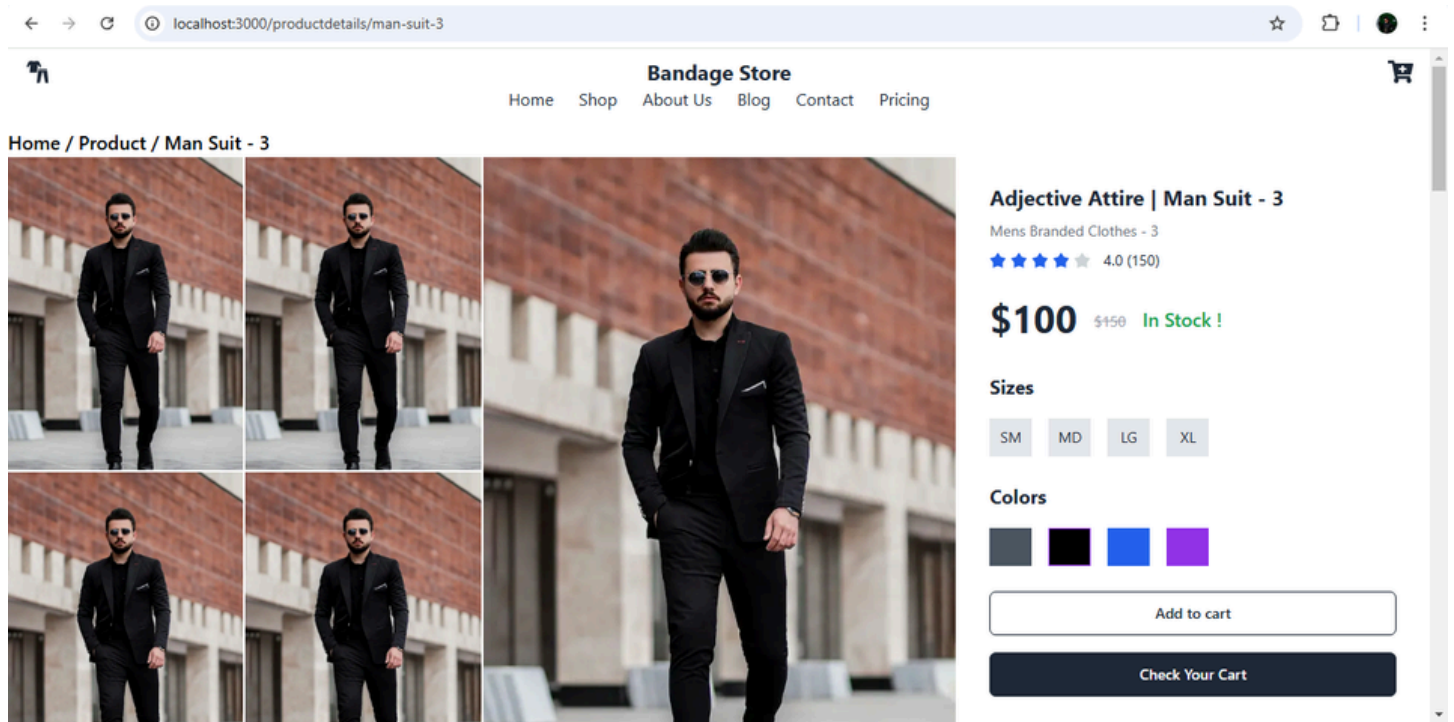
localhost:3000

Featured Products

Bestseller Products

Problems trying to resolve the conflict between





This screenshot shows the VS Code editor with the file `JS importSanityData.mjs` open. The Explorer sidebar on the left shows the project structure for `MY-APP`, including folders like `src`, `app`, and `api`. The main editor displays the following TypeScript code:

```
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       }
```

This screenshot shows the VS Code editor with the file `page.tsx` open. The Explorer sidebar on the left shows the project structure for `MY-APP`, including folders like `src`, `app`, and `api`. The main editor displays the following TypeScript code:

```
12 export default function Home() {
13
14   // Fetch product data from Sanity
15   const fetchData = async () => {
16     try {
17       const query = `*[_type == "product"] | order(_createdAt asc){
18         name,
19         price,
20         priceWithoutDiscount,
21         discountPercentage,
22         image,
23         slogan,
24         "slug":slug.current
25       }`;
26       const data = await client.fetch(query);
27
28       if (data && Array.isArray(data)) {
29         setProducts(data);
30       } else {
31         console.error("Unexpected data format:", data);
32       }
33     } catch (error) {
34       console.error("Failed to fetch products:", error);
35     }
36   };
37
38   useEffect(() => {
39     fetchData();
40   }, []);
41
42   return (
43     <div>
```