

# Document Tittle:Day 3- API Integration Report[Elite Furniture]

Api

Sanity setup

Data fetching

Sceema

Data fetching

This screenshot shows the VS Code editor with the `.env.local` file open. The Explorer sidebar on the left shows the project structure with folders like `src`, `sanity`, `lib`, `schemaTypes`, `TS`, `types`, and `products`. The `.env.local` file contains the following environment variables:

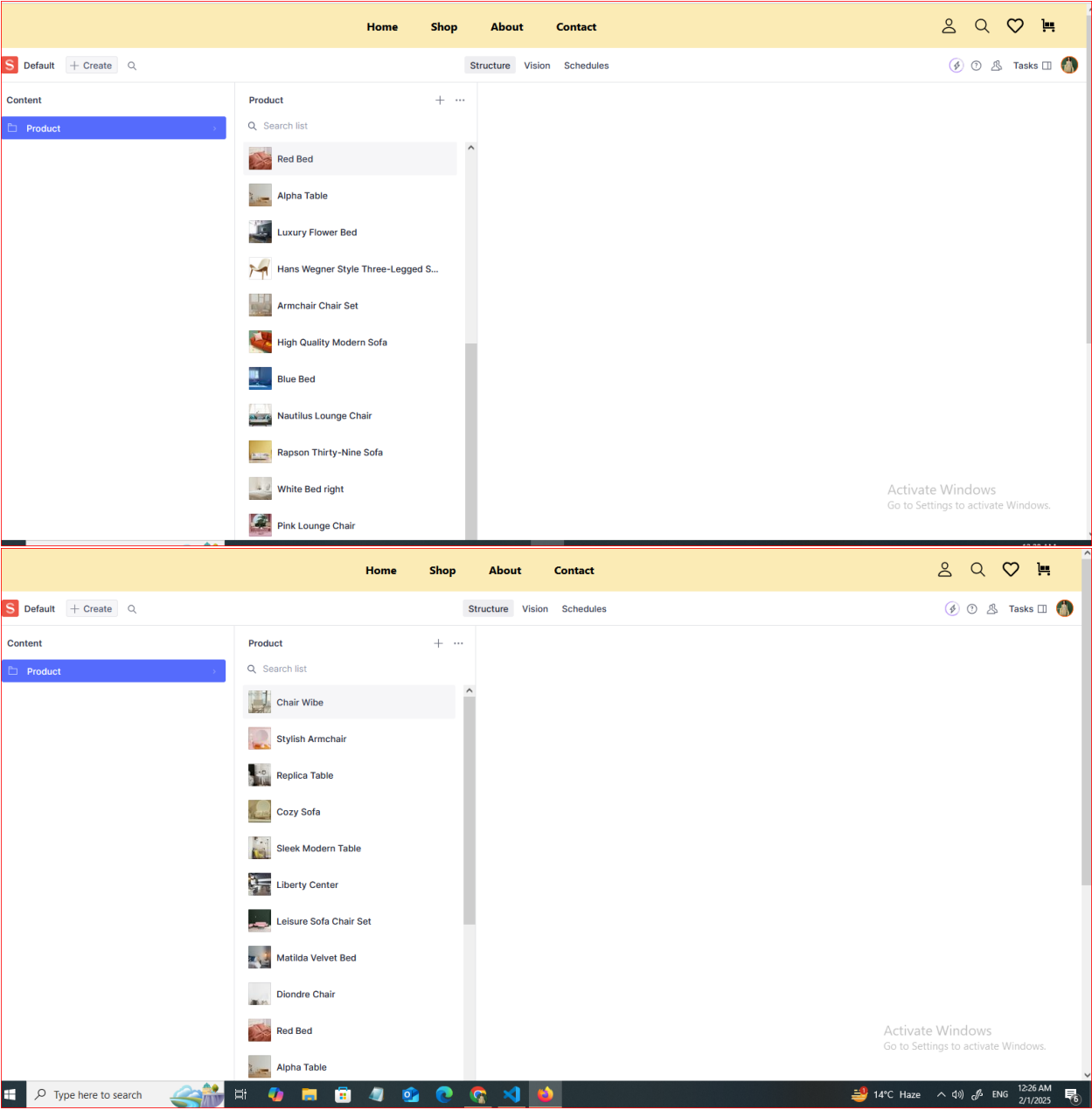
```
1 NEXT_PUBLIC_SANITY_PROJECT_ID="4oply5oy"  
2 NEXT_PUBLIC_SANITY_DATASET="production"  
3 SANITY_API_TOKEN=skuy3dt0zopHYcRiywbs0g1Hby2sSM60IRi1G1zK4tGLIjaIJBWwAsi85Ru4SsCpYzCwD2bTwEmLzqZOFg4x6mTmKAQYTshg46vtSe  
4  
5  
6
```

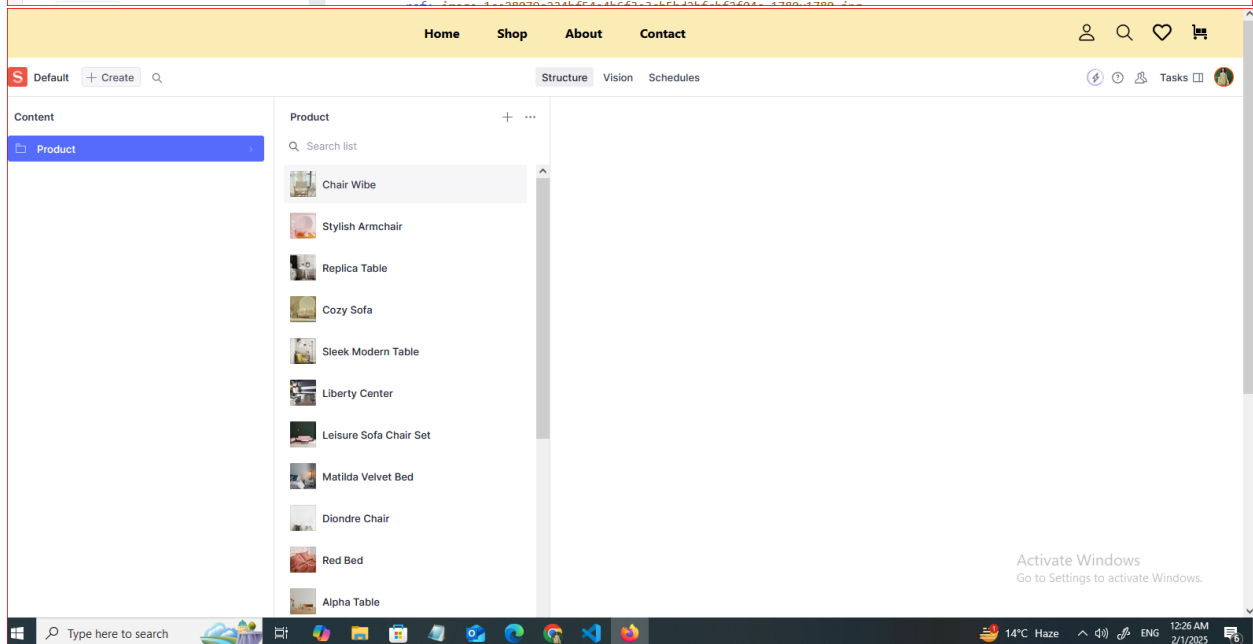
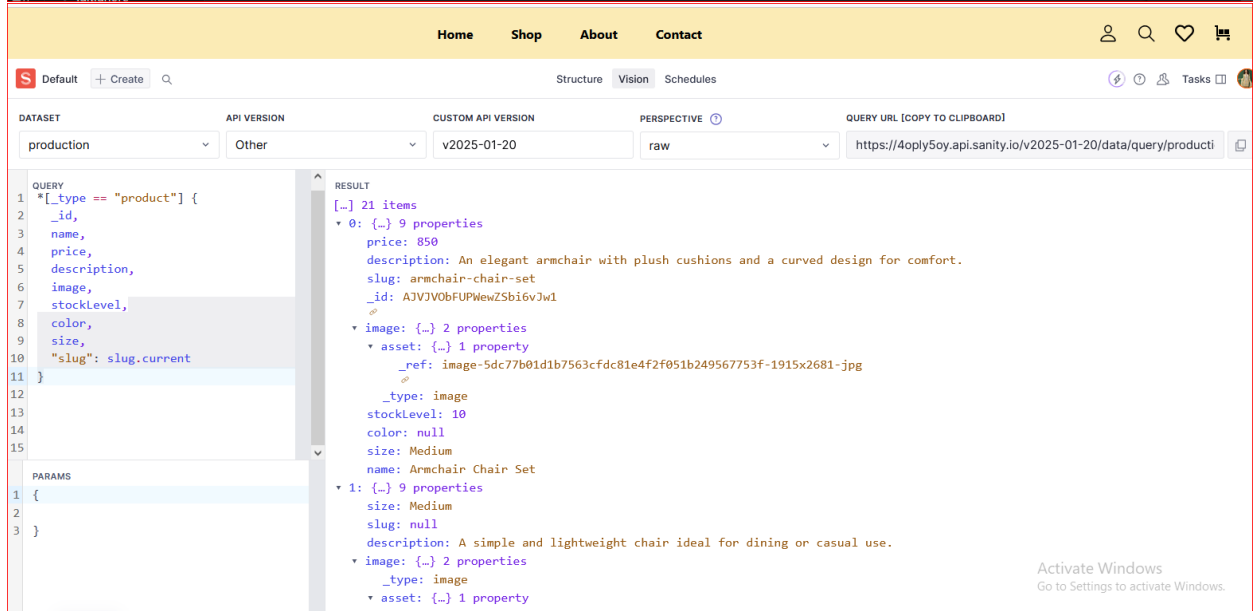
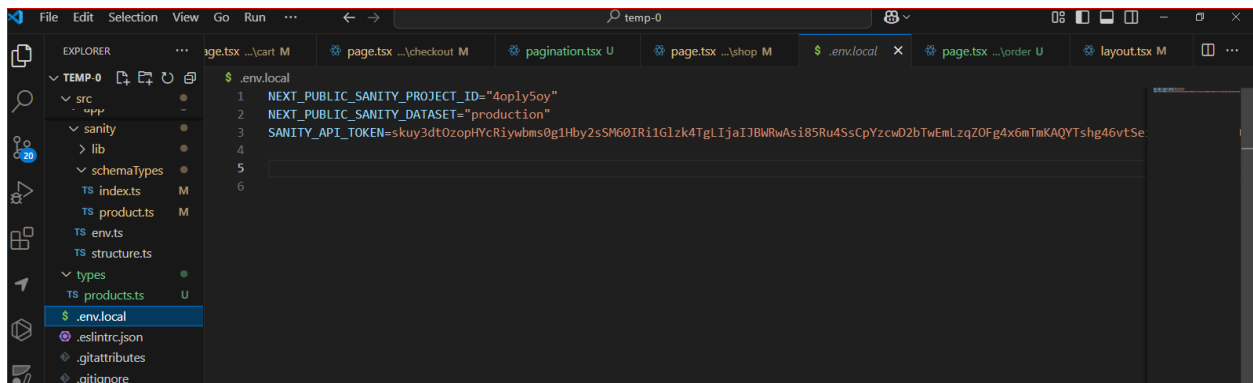
This screenshot shows the VS Code editor with the `product.ts` file open. The Explorer sidebar shows the file is located at `src > sanity > schemaTypes > TS > products > productSchema > fields`. The code defines a `productSchema` with the following fields:

```
18 const productSchema: ProductSchema = {  
22   fields: [  
23     {  
24       title: "Stock Level",  
25     },  
26     {  
27       name: "color",  
28       type: "array",  
29       title: "Color",  
30       of: [{ type: "string" }],  
31     },  
32     {  
33       name: "slug",  
34       type: "slug",  
35       title: "Slug",  
36       options: {  
37         source: "name",  
38         maxLength: 96,  
39         slugify: (input: string) =>  
40           input  
41             .toLowerCase()  
42             .replace(/\\s+/g, "-")  
43             .replace(/[\\w\\-]+/g, "")  
44             .slice(0, 96),  
45       },  
46     },  
47     {  
48       name: "size",  
49       type: "string",  
50     },  
51   ],  
52 }
```

This screenshot shows the VS Code editor with the `product.ts` file open. The Explorer sidebar shows the file is located at `src > sanity > schemaTypes > TS > products > productSchema > fields`. The code defines a `productSchema` with the following fields:

```
18 const productSchema: ProductSchema = {  
19   name: "product",  
20   type: "document",  
21   title: "Product",  
22   fields: [  
23     {  
24       name: "name",  
25       type: "string",  
26       title: "Name",  
27     },  
28     {  
29       name: "image",  
30       type: "image",  
31       title: "Image",  
32       options: {  
33         hotspot: true,  
34       },  
35     },  
36     {  
37       name: "price",  
38       type: "number",  
39       title: "Price",  
40     },  
41     {  
42       name: "description",  
43       type: "text",  
44       title: "Description",  
45     },  
46   ],  
47 }
```





C

```
File Edit Selection View Go Run ... temp-0
EXPLORER ... art M page.tsx ...\checkout M pagination.tsx U page.tsx ...\shop M JS importSanityData.mjs X page.tsx ...\order U layout.tsx
TEMP-0
  .next
  node_modules
  public
  scripts
    JS importSanityData.mjs
      src
        app
          about
            page.tsx
          account
            page.tsx
          cart
            page.tsx
          cartsidebar
            checkout
            page.tsx
          components
            Asghersofat.tsx
            blog.tsx

scripts > JS importSanityData.mjs > ...
1 import { createClient } from '@sanity/client';
2 import fetch from 'node-fetch';
3
4 // Initialize Sanity client
5 const client = createClient({
6   projectId: "4oply5oy",
7   dataset: "production",
8   useCdn: false, // Set to true if you want faster reads
9   apiVersion: '2025-01-13',
10  token: "skuy3dt0zopHYcRiywbms0g1Hby2sSM60IRi1Glzk4TgLIjaIJBRwAsi85Ru4SsCpYzcd2bTwEmLzqZOFg4x6mTmKAQYtshg46vtSeiKdSbM1I
11 });
12
13 // Function to upload an image to Sanity
14 Tabnine | Edit | Test | Explain | Document
15 async function uploadImageToSanity(imageUrl) {
16   try {
17     console.log(`Uploading image: ${imageUrl}`);
18
19     const response = await fetch(imageUrl);
20     if (!response.ok) {
21       throw new Error(`Failed to fetch image: ${imageUrl}`);
22     }
23   } catch (error) {
24     console.error(error);
25   }
26 }
```