

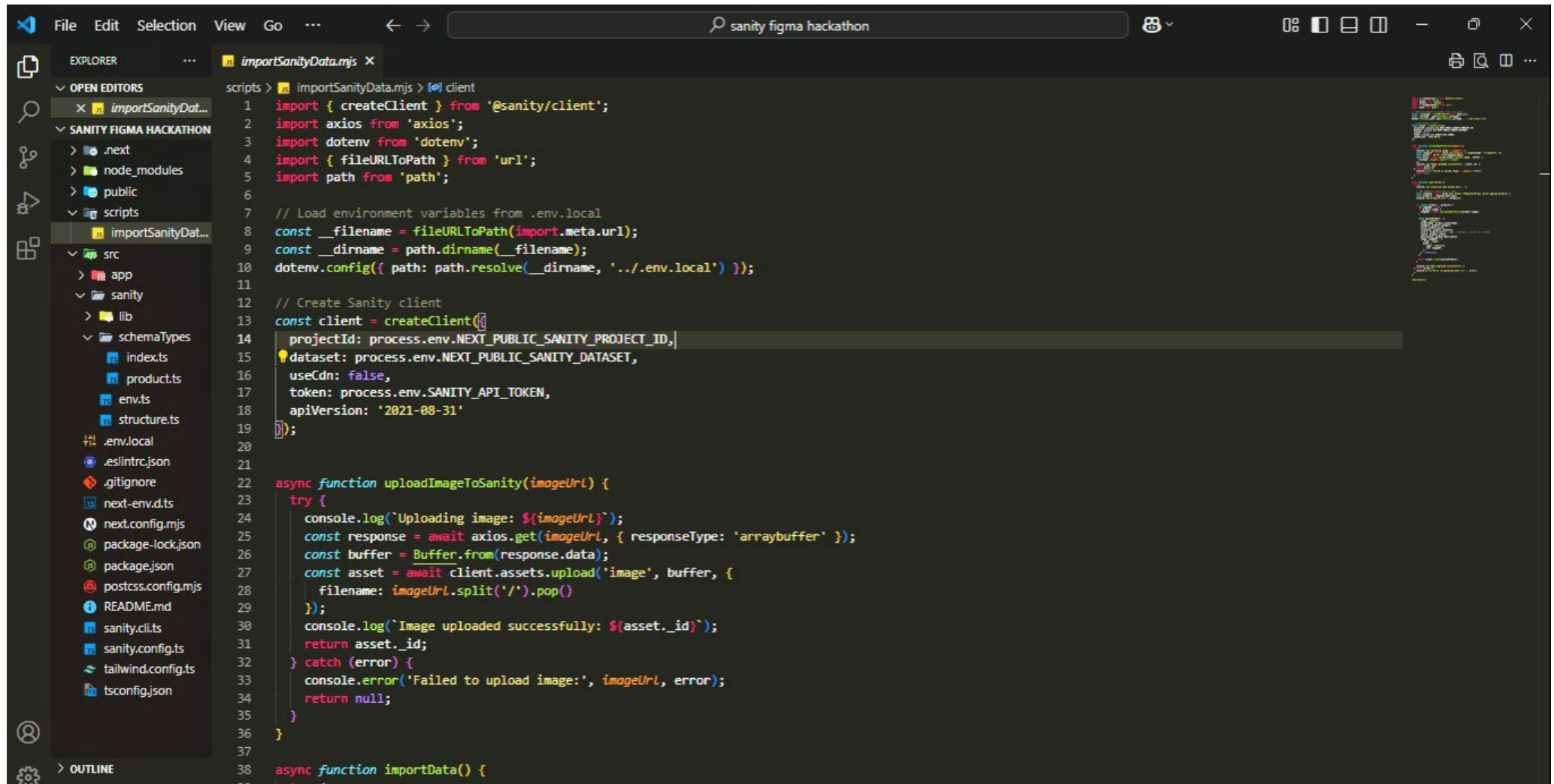
# hackathon3 day3 Apl integration



DAY 3 - API INTEGRATION  
AND DATA MIGRATION

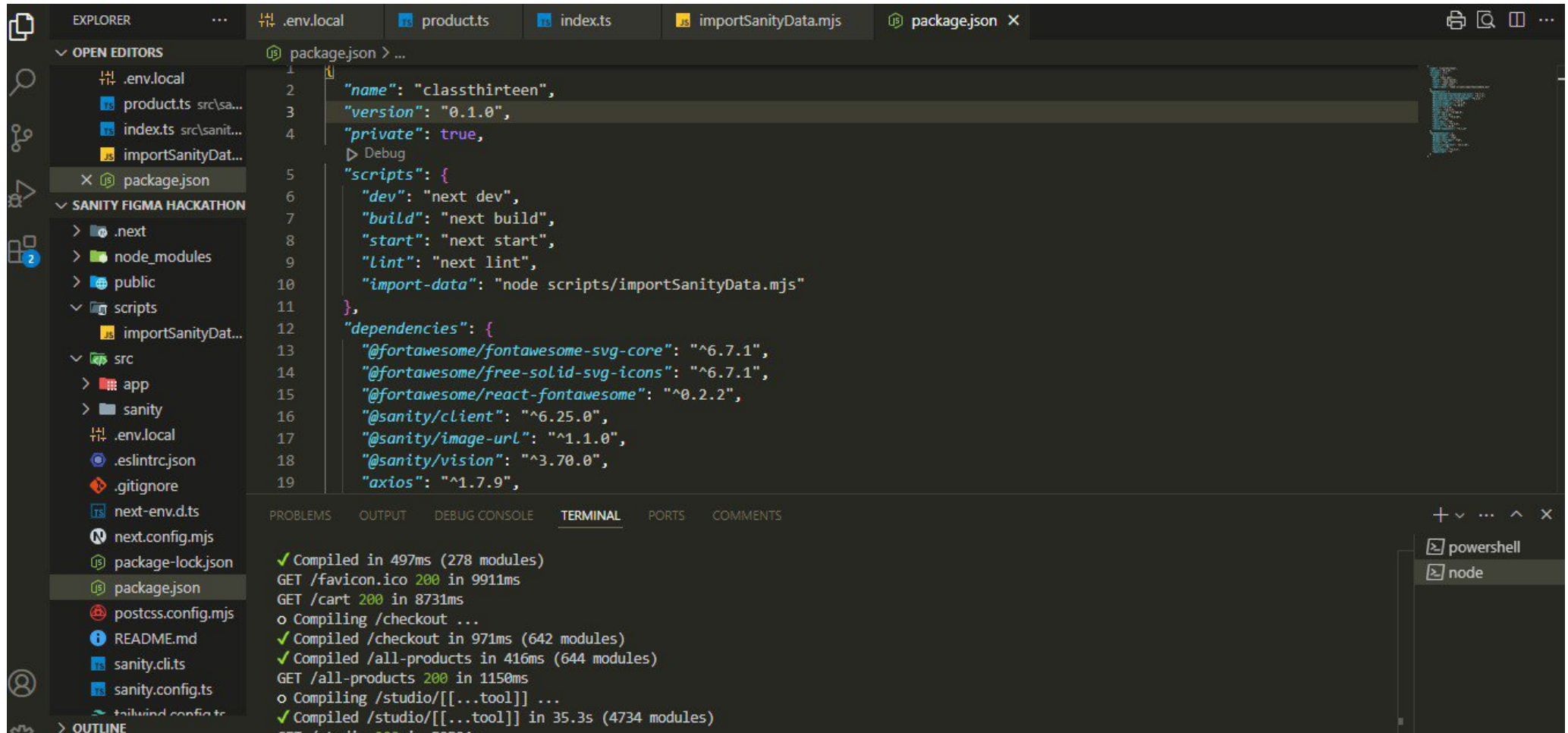


# 1. Migration Data To store API Data into Sanity



```
1 import { createClient } from '@sanity/client';
2 import axios from 'axios';
3 import dotenv from 'dotenv';
4 import { fileURLToPath } from 'url';
5 import path from 'path';
6
7 // Load environment variables from .env.local
8 const __filename = fileURLToPath(import.meta.url);
9 const __dirname = path.dirname(__filename);
10 dotenv.config({ path: path.resolve(__dirname, '../.env.local') });
11
12 // Create Sanity client
13 const client = createClient({
14   projectId: process.env.NEXT_PUBLIC_SANITY_PROJECT_ID,
15   dataset: process.env.NEXT_PUBLIC_SANITY_DATASET,
16   useCdn: false,
17   token: process.env.SANITY_API_TOKEN,
18   apiVersion: '2021-08-31'
19 });
20
21
22 async function uploadImageToSanity(imageUrl) {
23   try {
24     console.log(`Uploading image: ${imageUrl}`);
25     const response = await axios.get(imageUrl, { responseType: 'arraybuffer' });
26     const buffer = Buffer.from(response.data);
27     const asset = await client.assets.upload('image', buffer, {
28       filename: imageUrl.split('/').pop()
29     });
30     console.log(`Image uploaded successfully: ${asset._id}`);
31     return asset._id;
32   } catch (error) {
33     console.error('Failed to upload image:', imageUrl, error);
34     return null;
35   }
36 }
37
38 async function importData() {
39   try {
40     // ...
41   }
42 }
```

## 2.exporting data into Sanity



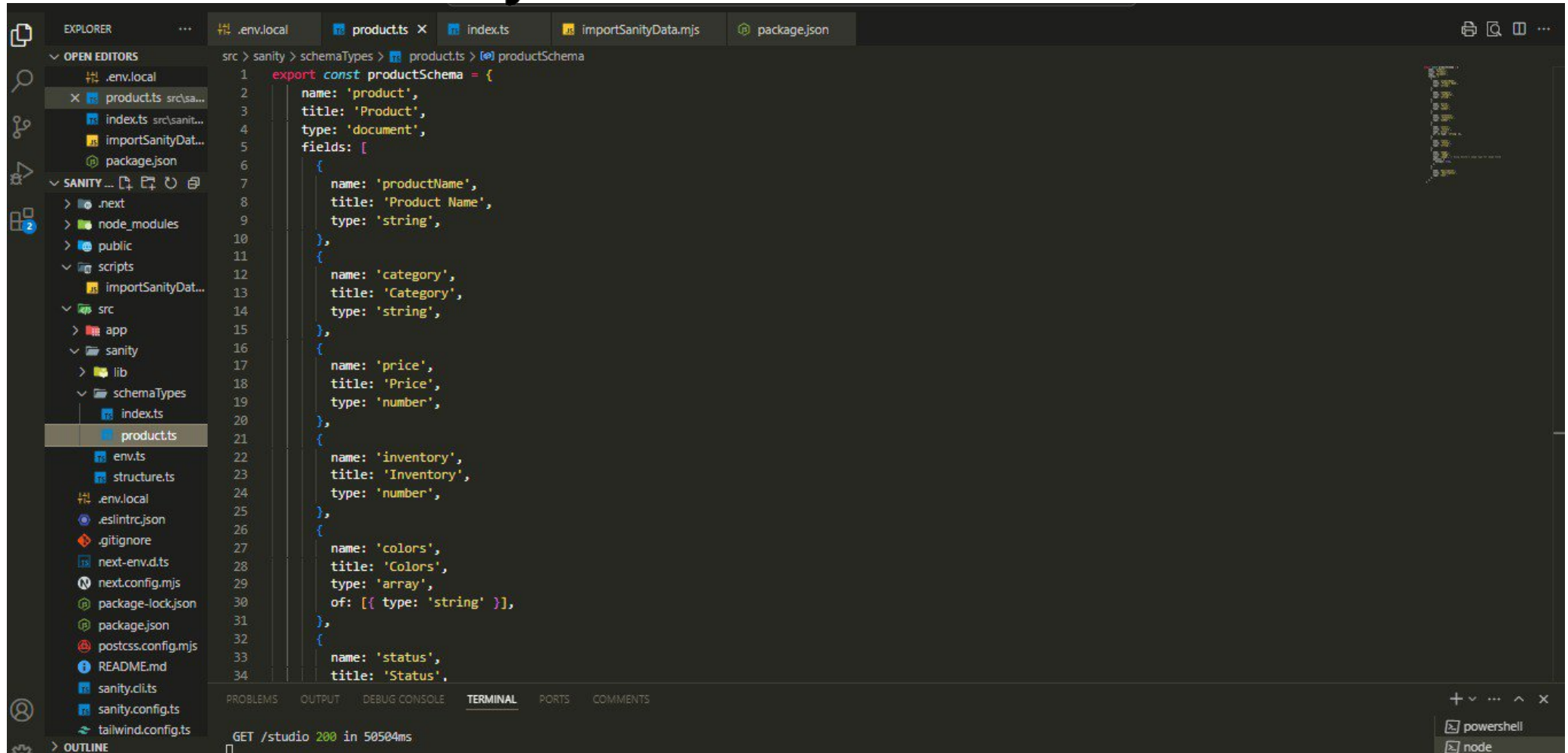
The screenshot shows a Visual Studio Code editor with the following components:

- EXPLORER:** Displays the project structure. The 'scripts' folder contains 'importSanityData.mjs'. The 'src' folder contains 'app', 'sanity', and 'tailwind.config.ts'.
- package.json:** The main configuration file for the project. It includes the following content:

```
1  {
2    "name": "classthirteen",
3    "version": "0.1.0",
4    "private": true,
5    "scripts": {
6      "dev": "next dev",
7      "build": "next build",
8      "start": "next start",
9      "lint": "next lint",
10     "import-data": "node scripts/importSanityData.mjs"
11   },
12   "dependencies": {
13     "@fortawesome/fontawesome-svg-core": "^6.7.1",
14     "@fortawesome/free-solid-svg-icons": "^6.7.1",
15     "@fortawesome/react-fontawesome": "^0.2.2",
16     "@sanity/client": "^6.25.0",
17     "@sanity/image-url": "^1.1.0",
18     "@sanity/vision": "^3.70.0",
19     "axios": "^1.7.9",
```
- TERMINAL:** Shows the output of a build command. The output indicates successful compilation and data export:

```
✓ Compiled in 497ms (278 modules)
GET /favicon.ico 200 in 991ms
GET /cart 200 in 8731ms
o Compiling /checkout ...
✓ Compiled /checkout in 971ms (642 modules)
✓ Compiled /all-products in 416ms (644 modules)
GET /all-products 200 in 1150ms
o Compiling /studio/[...tool] ...
✓ Compiled /studio/[...tool] in 35.3s (4734 modules)
```

# 3.Schema To Fill Accurate Result in Sanity



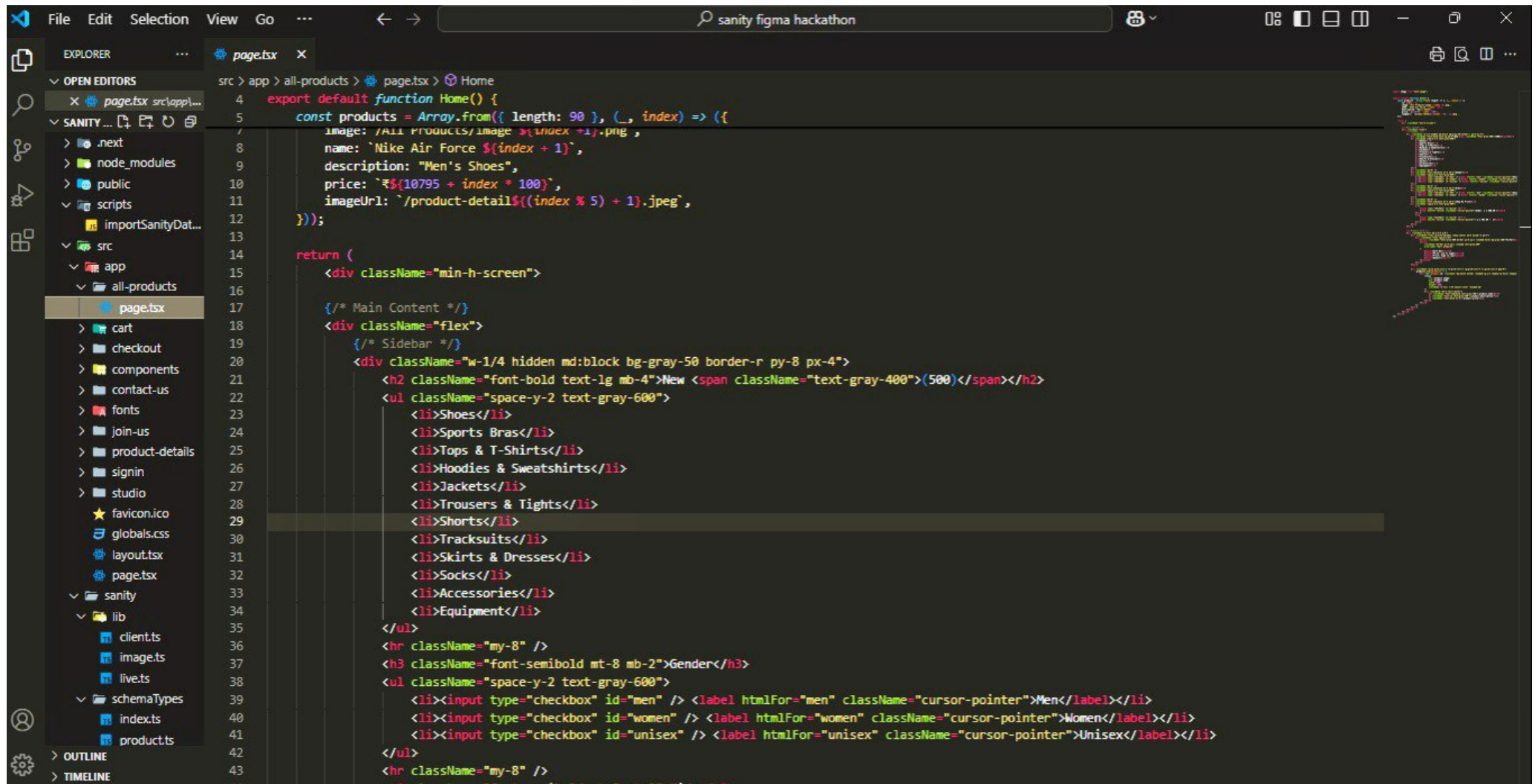
The screenshot shows a Visual Studio Code editor with a project structure on the left and a code editor in the center. The project structure includes folders like .next, node\_modules, public, scripts, src, app, sanity, lib, and schemaTypes. The schemaTypes folder contains index.ts and product.ts. The product.ts file is open, showing a Sanity schema definition for 'product'.

```
src > sanity > schemaTypes > product.ts > productSchema
1 export const productSchema = {
2   name: 'product',
3   title: 'Product',
4   type: 'document',
5   fields: [
6     {
7       name: 'productName',
8       title: 'Product Name',
9       type: 'string',
10    },
11    {
12      name: 'category',
13      title: 'Category',
14      type: 'string',
15    },
16    {
17      name: 'price',
18      title: 'Price',
19      type: 'number',
20    },
21    {
22      name: 'inventory',
23      title: 'Inventory',
24      type: 'number',
25    },
26    {
27      name: 'colors',
28      title: 'Colors',
29      type: 'array',
30      of: [{ type: 'string' }],
31    },
32    {
33      name: 'status',
34      title: 'Status',
```

The terminal at the bottom shows the command `GET /studio 200 in 50504ms`.



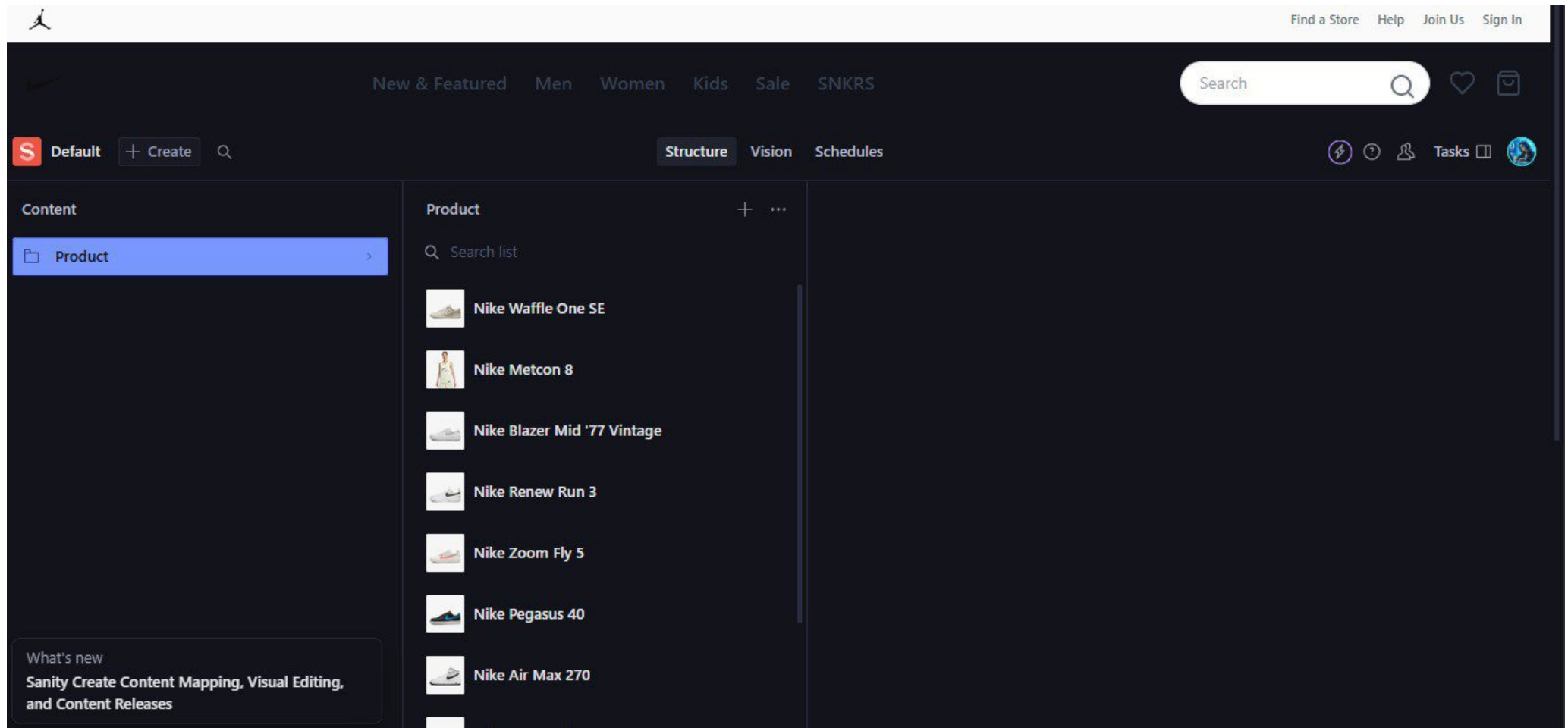
# 4. Making LOGIC for rendering API data



```
File Edit Selection View Go ... sanity figma hackathon
EXPLORER
  OPEN EDITORS
    page.tsx src/app/...
  SANITY ...
    .next
    node_modules
    public
    scripts
    importSanityDat...
  src
    app
      all-products
        page.tsx
      cart
      checkout
      components
      contact-us
      fonts
      join-us
      product-details
      signin
      studio
      favicon.ico
      globals.css
      layout.tsx
      page.tsx
    sanity
      lib
        clients.ts
        image.ts
        live.ts
        schemaTypes
        index.ts
        product.ts
    OUTLINE
    TIMELINE

src > app > all-products > page.tsx > Home
4 export default function Home() {
5   const products = Array.from({ length: 90 }, (_, index) => ({
6     image: `/All Products/image ${index + 1}.png`,
7     name: `Nike Air Force ${index + 1}`,
8     description: "Men's Shoes",
9     price: `₹${10795 + index * 100}`,
10    imageUrl: `/product-detail${(index % 5) + 1}.jpeg`,
11  }));
12
13
14  return (
15    <div className="min-h-screen">
16
17      {/* Main Content */}
18      <div className="flex">
19        {/* Sidebar */}
20        <div className="w-1/4 hidden md:block bg-gray-50 border-r py-8 px-4">
21          <h2 className="font-bold text-lg mb-4">New <span className="text-gray-400">(500)</span></h2>
22          <ul className="space-y-2 text-gray-600">
23            <li>Shoes</li>
24            <li>Sports Bras</li>
25            <li>Tops & T-Shirts</li>
26            <li>Hoodies & Sweatshirts</li>
27            <li>Jackets</li>
28            <li>Trousers & Tights</li>
29            <li>Shorts</li>
30            <li>Tracksuits</li>
31            <li>Skirts & Dresses</li>
32            <li>Socks</li>
33            <li>Accessories</li>
34            <li>Equipment</li>
35          </ul>
36          <hr className="my-8" />
37          <h3 className="font-semibold mt-8 mb-2">Gender</h3>
38          <ul className="space-y-2 text-gray-600">
39            <li><input type="checkbox" id="men" /> <label htmlFor="men" className="cursor-pointer">Men</label></li>
40            <li><input type="checkbox" id="women" /> <label htmlFor="women" className="cursor-pointer">Women</label></li>
41            <li><input type="checkbox" id="unisex" /> <label htmlFor="unisex" className="cursor-pointer">Unisex</label></li>
42          </ul>
43          <hr className="my-8" />
44        </div>
45      </div>
46    </div>
47  );
48}
```

# 5.integration Output





Find a Store Help Join Us Sign In



New & Featured Men Women Kids Sale SNKRS

Search



Default

+ Create



Structure

Vision

Schedules



Tasks



Content

Product

Product



Search list



Nike Waffle One SE



Nike Metcon 8



Nike Blazer Mid '77 Vintage



Nike Renew Run 3



Nike Zoom Fly 5



Nike Pegasus 40



Nike Air Max 270

Nike Waffle One SE



Product

# Nike Waffle One SE

Product Name

Nike Waffle One SE

Category

Women's Shoes

Price

What's new

Sanity Create Content Mapping, Visual Editing, and Content Releases