

NAME : MADIHA RASHID

CLASS TIMING : FRIDAY 9AM TO 12AM

TEACHER NAME : HAMZA SYED

ROLL NO: 00195553

**TASK : DAY 03 API INTEGRATION
AND DATA MIRATION**

MY-APP

src

app

product

sign-in

studio

favicon.ico

globals.css

layout.tsx

page.tsx

sanity

lib

schemaTypes

TS index.ts

TS products.ts

TS env.ts

TS structure.ts

.env.local

.eslintrc.json

.gitignore

next-env.d.ts

next.config.mjs

package-lock.json

package.json

postcss.config.mjs

README.md

sanity.cli.ts

sanity.config.ts

tailwind.config.ts

tsconfig.json

src > sanity > schemaTypes > TS products.ts > productSchema

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

```
export const productSchema = {
  name: 'product',
  title: 'Product',
  type: 'document',
  fields: [
    {
      name: 'productName',
      title: 'Product Name',
      type: 'string',
    },
    {
      name: 'category',
      title: 'Category',
      type: 'string',
    },
    {
      name: 'price',
      title: 'Price',
      type: 'number',
    },
    {
      name: 'inventory',
      title: 'Inventory',
      type: 'number',
    },
    {
      name: 'colors',
      title: 'Colors',
      type: 'array',
      of: [{ type: 'string' }],
    },
  ],
}
```

SCHEMA

```
MY-APP
├── .next
├── node_modules
├── public
├── scripts
├── JS importSanityData.mjs
├── src
├── app
├── cart
├── checkoutpage
├── component
├── banner.tsx
├── bestproduct.tsx
├── bestproduct1.tsx
├── essentials.tsx
├── flight.tsx
├── footer.tsx
├── gear.tsx
├── header.tsx
├── header2.tsx
├── header3.tsx
├── herosection.tsx
├── practis.tsx
├── silp.tsx
├── contactus
├── joinus
├── new-feature
├── product
├── page.tsx
├── sign-in
├── studio
└──
```

```
scripts > JS importSanityData.mjs > ...
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  }
```

Sanity integration

```

async function importData() {
  try {
    console.log('migrating data please wait...');

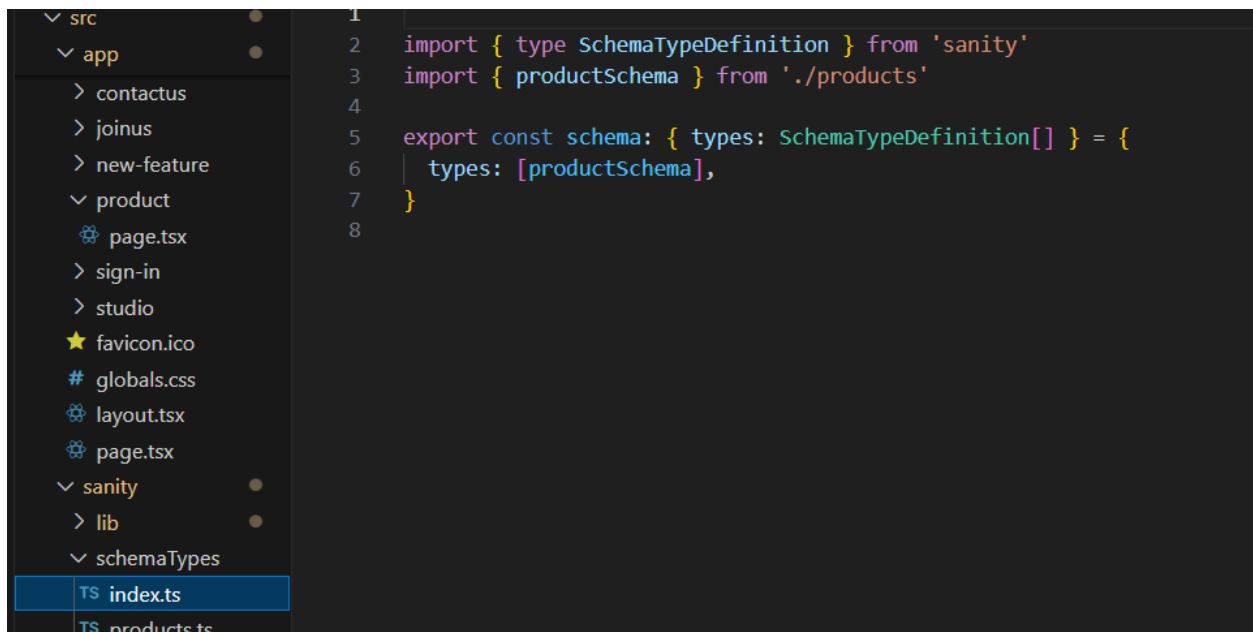
    // API endpoint containing car data
    const response = await axios.get('https://template-03-api.vercel.app/api/products');
    const products = response.data.data;
    console.log("products ==>> ", products);

    for (const product of products) {
      let imageRef = null;
      if (product.image) {
        imageRef = await uploadImageToSanity(product.image);
      }

      const sanityProduct = {
        _type: 'product',
        productName: product.productName,
        category: product.category,
        price: product.price,
        inventory: product.inventory,
        colors: product.colors || [], // Optional, as per your schema
        status: product.status,
        description: product.description,
        image: imageRef ? {
          _type: 'image',
          asset: {
            _type: 'reference',
            _ref: imageRef,
          },
        } : undefined,
      };

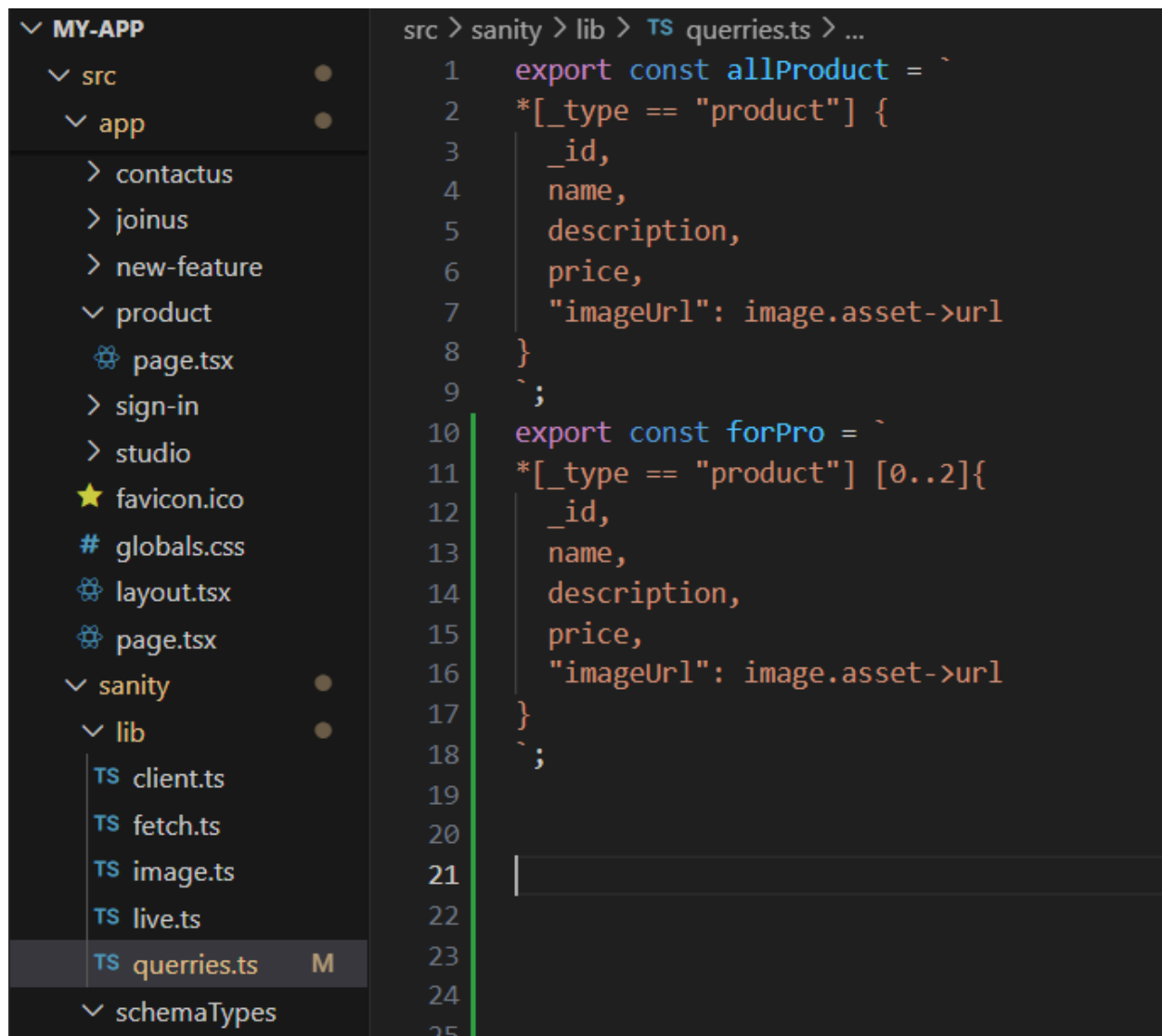
      await client.create(sanityProduct);
    }
  }
}

```



```
1
2 import { type SchemaTypeDefinition } from 'sanity'
3 import { productSchema } from './products'
4
5 export const schema: { types: SchemaTypeDefinition[] } = {
6   types: [productSchema],
7 }
8
```

INDEX.TS



```
src > sanity > lib > TS queries.ts > ...
1  export const allProduct = `
2    *[_type == "product"] {
3      _id,
4      name,
5      description,
6      price,
7      "imageUrl": image.asset->url
8    }
9  `;
10 export const forPro = `
11   *[_type == "product"] [0..2]{
12     _id,
13     name,
14     description,
15     price,
16     "imageUrl": image.asset->url
17   }
18 `;
19
20
21
22
23
24
25
```

QUERRIES.TS FILE

Data fetch

Content

Product

Product

🔍 Search list



Nike Metcon 8



Nike Waffle One SE



Nike Blazer Mid '77 Vintage



Nike Renew Run 3



Nike Zoom Fly 5



Nike Pegasus 40



Nike Air Max 270



Nike Court Vision Low



Nike Dri-FIT UV Hyverse



Nike Dunk Low Retro SE



Nike Standard Issue Basketball Jersey

What's new

Sanity Create Content Mapping, Visual Editing,
and Content Releases

localhost:3000/studio/structure/product:0e0ztuIY6NWdvbiBFRhFi5

Products



The Nike Air Force 1 Mid '07 delivers timeless style with premium leather and mid-cut design. Perfect for everyday wear, it provides exceptional comfort and durability. The iconic Air-Sole cushioning adds responsive support for long-lasting performance.

₹ 10795



The Nike Court Vision Low Next Nature combines retro basketball style with modern sustainability. Made with at least 20% recycled content, it features a sleek design, durable materials, and ultimate comfort for everyday wear.

₹ 4995



The Nike Dri-FIT UV Hyverse graphic top offers comfort and UV protection in style. Made from soft, breathable fabric, this fitness top keeps you cool and dry during workouts or casual outings.

₹ 2495

DISPLAY ON BROWSER

API integration migration

API integration migration refers to the process of transferring or adapting an existing application programming interface (API) to work within a new system, platform, or environment. This involves updating endpoints, data formats, or workflows to ensure compatibility while maintaining functionality and minimizing disruptions.