

Day-03 API Integration Report - 'SoleCare'

API Integration Process:

Step 1: API Endpoint Setup

- API Endpoint: "<https://template-03-api.vercel.app/api/products>"
- In the provided API the product data was given included fields : product name, price, category, description, inventory, image, color, status.
- "Axios" (library) was used to fetch data from the API.

Step 2: Fetch Data from API

-The API call retrieves an array of product objects from the endpoint, ensuring proper data serialization (converting data into a format that can be stored).



```
1 const response = await axios.get('https://template-03-api.vercel.app/api/products');  
2 const products = response.data.data;
```

Adjustments in schemas:

Color, Status and Inventory fields updated, as I didn't add them yesterday but today I updated the fields according to provided API.

```
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',
35      type: 'string',
36    },
37    {
38      name: 'image',
39      title: 'Product Image',
40      type: 'image',
41      options: {
42        hotspot: true,
43      },
44    },
45    {
46      name: 'description',
47      title: 'Description',
48      type: 'text',
49    },
50    {
51      name: 'id',
52      title: 'Product ID',
53      type: 'string',
54    },
55    {
```

```
56     name: 'size',
57     title: 'Size',
58     type: 'string',
59   },
60 ],
61 };
62
```

Migration Steps:

- Installed required dependencies: `@sanity/client`, `axios` and `dotenv`.
- Update `.env.local` file to store environmental variables.
- Retrieved product data from API endpoint using **Axios**.
- Downloaded images from API's image field using **Axios**.

For uploading images.

```
1  const response = await axios.get(imageUrl, { responseType: 'arraybuffer' });
2  const buffer = Buffer.from(response.data);
3  const asset = await client.assets.upload('image', buffer, {
4    filename: imageUrl.split('/').pop()
5  });
```

This code fetches an image from a given URL, converts it into a buffer, and uploads it to a storage client.

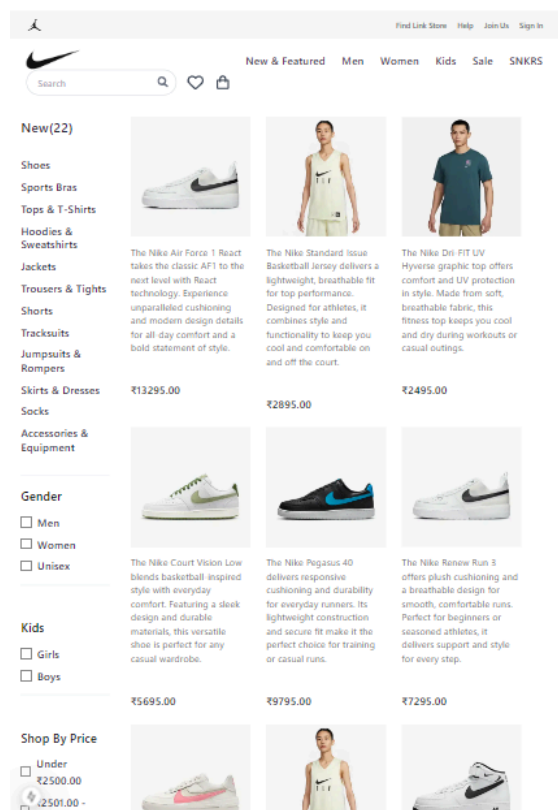
Tools used:

- **Sanity Client** for interacting with Sanity CMS.
- **Axios** for making HTTP requests to fetch API data and images.
- **Dotenv** to load environmental variables from `.env.local`.

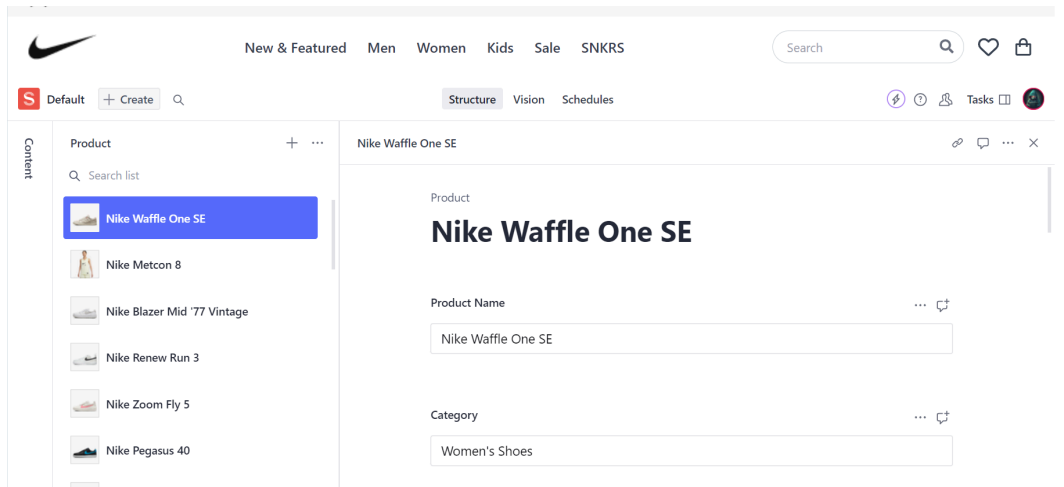
API calls:

```
PROBLEMS OUTPUT DEBUG CONSOLE TEL
{
  productName: "Nike Blazer Mid '7",
  category: "Men's Shoes",
  price: 8495,
  inventory: 22,
  colors: [ 'White', 'Black' ],
  status: 'Just In',
  image: 'https://template-03-api',
  description: "The Nike Blazer Mid '7 is a classic sneaker that combines versatility and a timeless look.",
},
{
  productName: 'Nike Metcon 8',
  category: "Men's Training Shoes",
  price: 10295,
  inventory: 25,
  colors: [ 'Black' ],
  status: 'Best Seller',
  image: 'https://template-03-api',
  description: "The Nike Metcon 8 is a training shoe designed for weightlifting, CrossFit, and other high-intensity workouts.",
},
{
  productName: 'Nike Waffle One S',
  category: "Women's Shoes",
  price: 7895,
  inventory: 30,
  colors: [ 'Pink' ],
  status: 'Just In',
  image: 'https://template-03-api',
  description: "The Nike Waffle One S is a women's sneaker known for its comfort and performance in one standout shoe.",
}
}
main 0 0 0 0
```

Data Displayed in Front-end.



Sanity CMS Fields:



Code Snippets:

API Integration:

```
1  const response = await axios.get('https://template-03-api.vercel.app/api/products');
2  const products = response.data.data;
```

Migration Script:

```

1  async function uploadImageToSanity(imageUrl) {
2      try {
3          console.log('Uploading image: ${imageUrl}');
4          const response = await axios.get(imageUrl, { responseType: 'arraybuffer' });
5          const buffer = Buffer.from(response.data);
6          const asset = await client.assets.upload('image', buffer, {
7              filename: imageUrl.split('/').pop()
8          });
9          console.log('Image uploaded successfully: ${asset._id}');
10         return asset._id;
11     } catch (error) {
12         console.error('Failed to upload image:', imageUrl, error);
13         return null;
14     }
15 }
16
17 async function importData() {
18     try {
19         console.log('migrating data please wait...');
20
21         // API endpoint containing product data
22         const response = await axios.get('https://template-03-api.vercel.app/api/products');
23         const products = response.data.data;
24         console.log("products ==> ", products);
25
26         for (const product of products) {
27             let imageRef = null;
28             if (product.image) {
29                 imageRef = await uploadImageToSanity(product.image);
30             }
31
32             const sanityProduct = {
33                 _type: 'product',
34                 productName: product.productName,
35                 category: product.category,
36                 price: product.price,
37                 inventory: product.inventory,
38                 colors: product.colors || [], // Optional, as per your schema
39                 status: product.status,
40                 description: product.description,
41                 image: imageRef ? {
42                     _type: 'image',
43                     asset: {
44                         _type: 'reference',
45                         _ref: imageRef,
46                     },
47                 } : undefined,
48             };
49
50             await client.create(sanityProduct);
51         }
52
53         console.log('Data migrated successfully!');
54     } catch (error) {
55         console.error('Error in migrating data ==> ', error);
56     }
57 }
58
59 importData();
60

```

That's how I completed Today's task of Day-03 of Hackathon-03.

Created by ARISHBA KHAN - SENIOR STUDENT.

Sunday 2 to 5PM.