

DAY 3

Api Integration Report

Step 1:

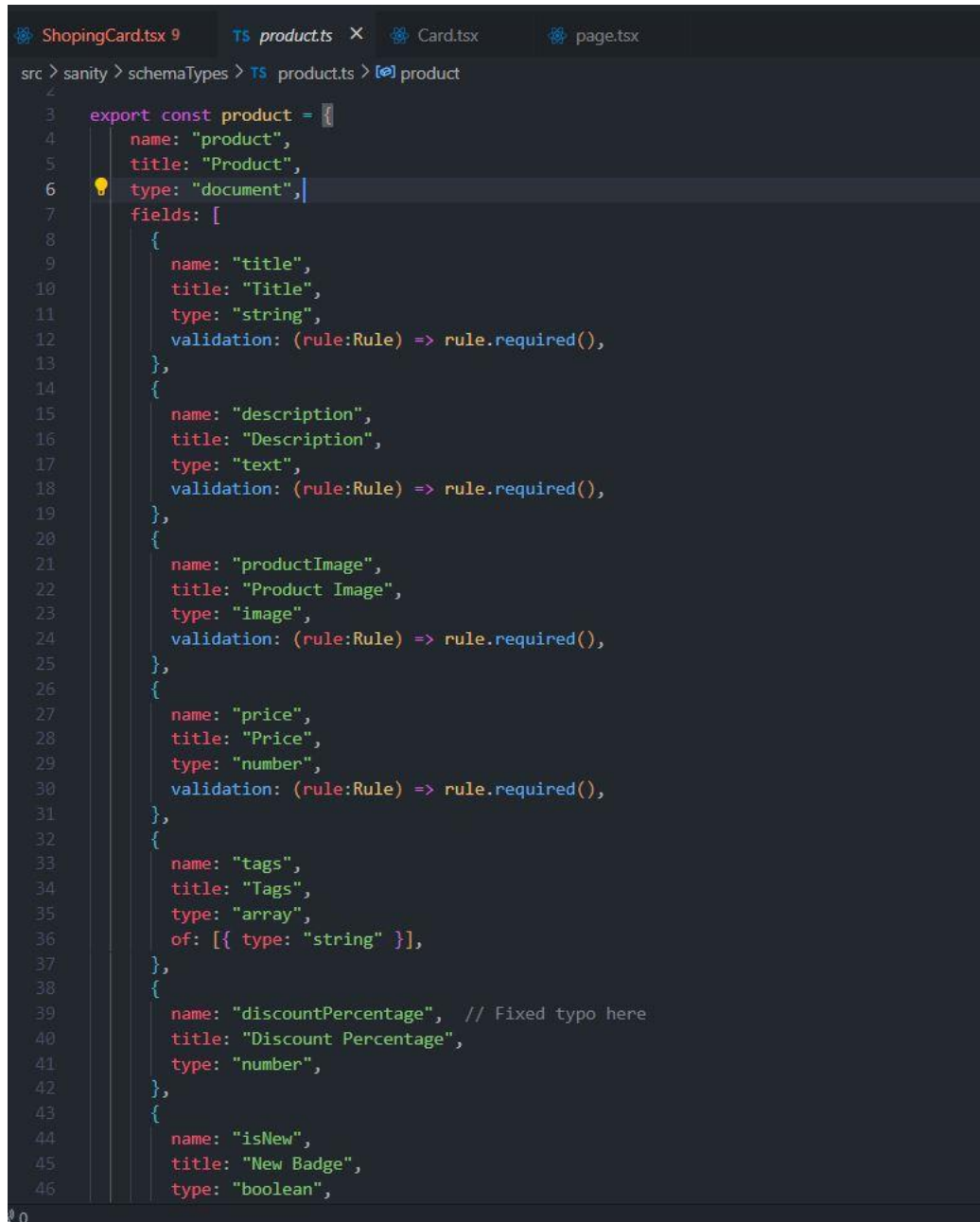
Data Structure : The Api provides product data , including fields like products name, price , color , description.

Step 2:

Fetch Data from Api:
Code Snippet

```
const response = await axios.get('https://template6-six.vercel.app/api/products')  
const products = response.data
```

2. Shcema



The screenshot shows a code editor with a dark theme. The top bar displays the file path: `src > sanity > schemaTypes > TS product.ts > product`. The editor contains a TypeScript file named `product.ts` with the following content:

```
1  export const product = {
2
3    name: "product",
4    title: "Product",
5    type: "document",
6    fields: [
7      {
8        name: "title",
9        title: "Title",
10       type: "string",
11       validation: (rule: Rule) => rule.required(),
12     },
13     {
14       name: "description",
15       title: "Description",
16       type: "text",
17       validation: (rule: Rule) => rule.required(),
18     },
19     {
20       name: "productImage",
21       title: "Product Image",
22       type: "image",
23       validation: (rule: Rule) => rule.required(),
24     },
25     {
26       name: "price",
27       title: "Price",
28       type: "number",
29       validation: (rule: Rule) => rule.required(),
30     },
31     {
32       name: "tags",
33       title: "Tags",
34       type: "array",
35       of: [{ type: "string" }],
36     },
37     {
38       name: "discountPercentage", // Fixed typo here
39       title: "Discount Percentage",
40       type: "number",
41     },
42     {
43       name: "isNew",
44       title: "New Badge",
45       type: "boolean",
46     }
47   ]
48 }
```

3. Data Migration

1. Install Required Dependencies `@sanity/client` : axios , dotenv created a .env.local file to secure store.

2. Retrieve Product data from the api endpoint using axios.