

Day 3 Documentation

API_Integration_and_Heckathon_2025

1.Sanity Integration

"I have successfully added and structured the required data (e.g., product details, categories, and images) in Sanity CMS. The data is now ready to be fetched and integrated into the e-commerce website using Sanity's API for seamless content management and real-time updates.

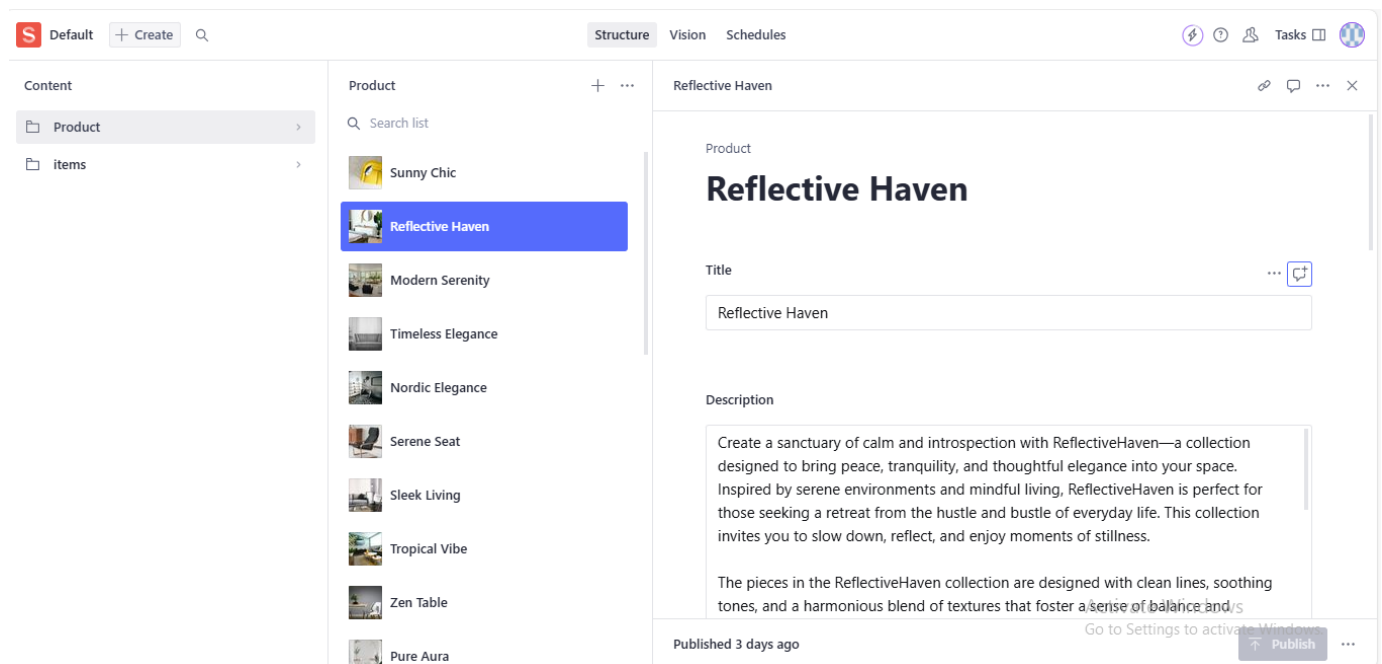
```
src > sanity > schemaTypes > TS products.ts > product > fields
1  import { defineType } from "sanity"
2
3  export const product = defineType({
4    name: "product",
5    title: "Product",
6    type: "document",
7    fields: [
8      {
9        name: "title",
10       title: "Title",
11       validation: (rule) => rule.required(),
12       type: "string"
13     },
14     {
15       name: "description",
16       type: "text",
17       validation: (rule) => rule.required(),
18       title: "Description",
19     },
20     {
21       name: "productImage",
22       type: "image",
23       validation: (rule) => rule.required(),
24       title: "Product Image"
25     },
26     {
27       name: "price",
28       type: "number",
29       validation: (rule) => rule.required(),
30       title: "Price",
31     },
32   ],
33 })
```

0

Ln 10, Col 28 Spaces: 4 UTF-8 CRLF () TypeScript Go Live 7:49 PM

Activate Windows
Go to Settings to activate Windows.

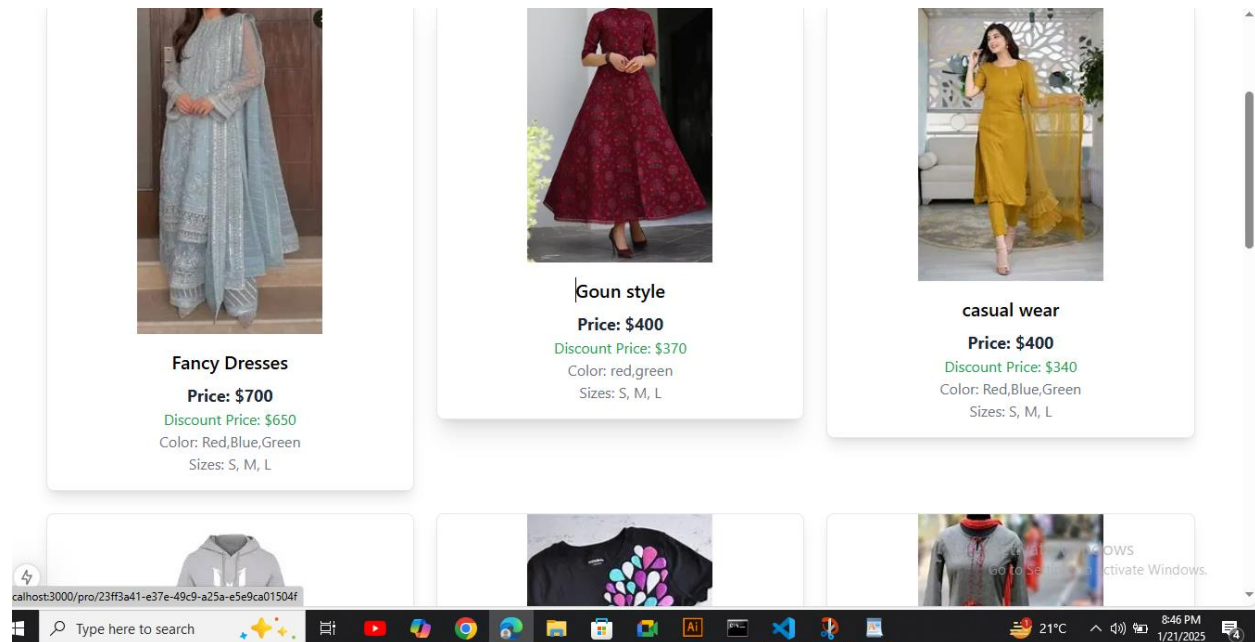
```
script > JS importSanityData.mjs > uploadImageToSanity
11 async function uploadImageToSanity(imageUrl) {
20   const buffer = await response.arrayBuffer();
21   const bufferImage = Buffer.from(buffer);
22
23   const asset = await client.assets.upload('image', bufferImage, {
24     filename: imageUrl.split('/').pop(),
25   });
26
27   console.log(`Image uploaded successfully: ${asset._id}`);
28   return asset._id;
29 } catch (error) {
30   console.error('Failed to upload image:', imageUrl, error);
31   return null;
32 }
33
34
35 async function uploadProduct(product) {
36   try {
37     const imageId = await uploadImageToSanity(product.imageUrl);
38
39     if (imageId) {
40       const document = {
41         _type: 'product',
42         title: product.title,
43         price: product.price,
44         productImage: {
45           _type: 'image',
46           asset: {
47             _ref: imageId,
48           },
49         },
50         tags: product.tags,
```



2.Data Fetching

"The data has been successfully fetched from Sanity CMS using its API (GROQ/GraphQL). The integration ensures real-time, accurate, and structured content delivery for the e-

commerce website, enabling dynamic product listings, details, and other relevant information to be displayed seamlessly."



Some Key points;

1. **Data Source:** Data is stored and managed in Sanity CMS.
2. **API Integration:** Fetched using Sanity's API (GROQ or GraphQL).
3. **Real-Time Updates:** Ensures the latest data is displayed on the website.
4. **Structured Content:** Data is organized (e.g., products, categories, images).
5. **Dynamic Display:** Powers product listings, details, and other features.
6. **Seamless Integration:** Connects smoothly with the website for accurate content delivery.