

Document Title: "Day 3 - API Integration Report - AVION"

Create Sanity Project

S

Malikzeeshantariq

hackathon-Ecommerce-2025

56 days left in trial

HA

Malikzeeshantariq

hackathon-Ecommerce-2025

PLAN

Boosted Growth Trial

STATUS

Active

PROJECT ID

fg8bqe9f

Getting started

Overview

Members

Studios

Datasets

Access

Activity

Usage

Plan

API

Settings

Next steps

Initialize your project with the CLI

Run this command in your Terminal to continue setting up your project.

npm create sanity@latest -- --project fg8bqe9f --dataset produ

Copy

Having issues with the CLI?

Project members

View all

Invite your first team member

Invite project members

Usage

View more

1.3k / 1m	596 / 250k
API CDN Requests	API Requests
41.8 MB / 100 GB	552.8 MB / 100 GB
Assets	Bandwidth
1 / 2	32 / 10k
	Documents

Activity

View more

API

Jan 22 at 3:37 PM

API token new read token was added to the project

API

Jan 19 at 5:37 PM

API token Read Token was added to the project

What's new

Sanity Create Content Mapping, Visual Editing, and Content Releases



Malikzeeshantariq

hackathon-Ecommerce-2025



56 days left in trial



Malikzeeshantariq

## hackathon-Ecommerce-2025

PLAN

Boosted Growth Trial

STATUS

Active

PROJECT ID

fg8bqe9f



Getting started



Overview



Members



Studios



Datasets



Access



Activity



Usage



Plan



API



Settings

### Webhooks

CORS origins

Tokens

### GROQ-powered webhooks

HTTP callbacks to a given URL triggered by changes in your content lake

[Learn more about webhooks](#)

[+ Create webhook](#)

0 of 2  
webhooks  
(2 included in plan)

[Get more webhooks](#)



There are no GROQ-powered webhooks in this project  
Maybe try creating a new webhook?

# Added Token

S

Malikzeeshantariq

hackathon-Ecommerce-2025

56 days left in trial

Getting started

Overview

Members

Studios

Datasets

Access

Activity

Usage

Plan

API

Settings

Webhooks

CORS origins

Tokens

CORS origins

Hosts that can connect to the project API.

ORIGIN	CREDENTIALS	CREATED	
http://localhost:3000	Allowed	3 days	

+ Add CORS origin

Tokens

Tokens are used to authenticate apps and scripts to access project data.

NAME	PERMISSIONS	CREATED	
new read token	Viewer	7 hours	
Read Token	Viewer	3 days	
backend token	Developer	3 days	

+ Add API token

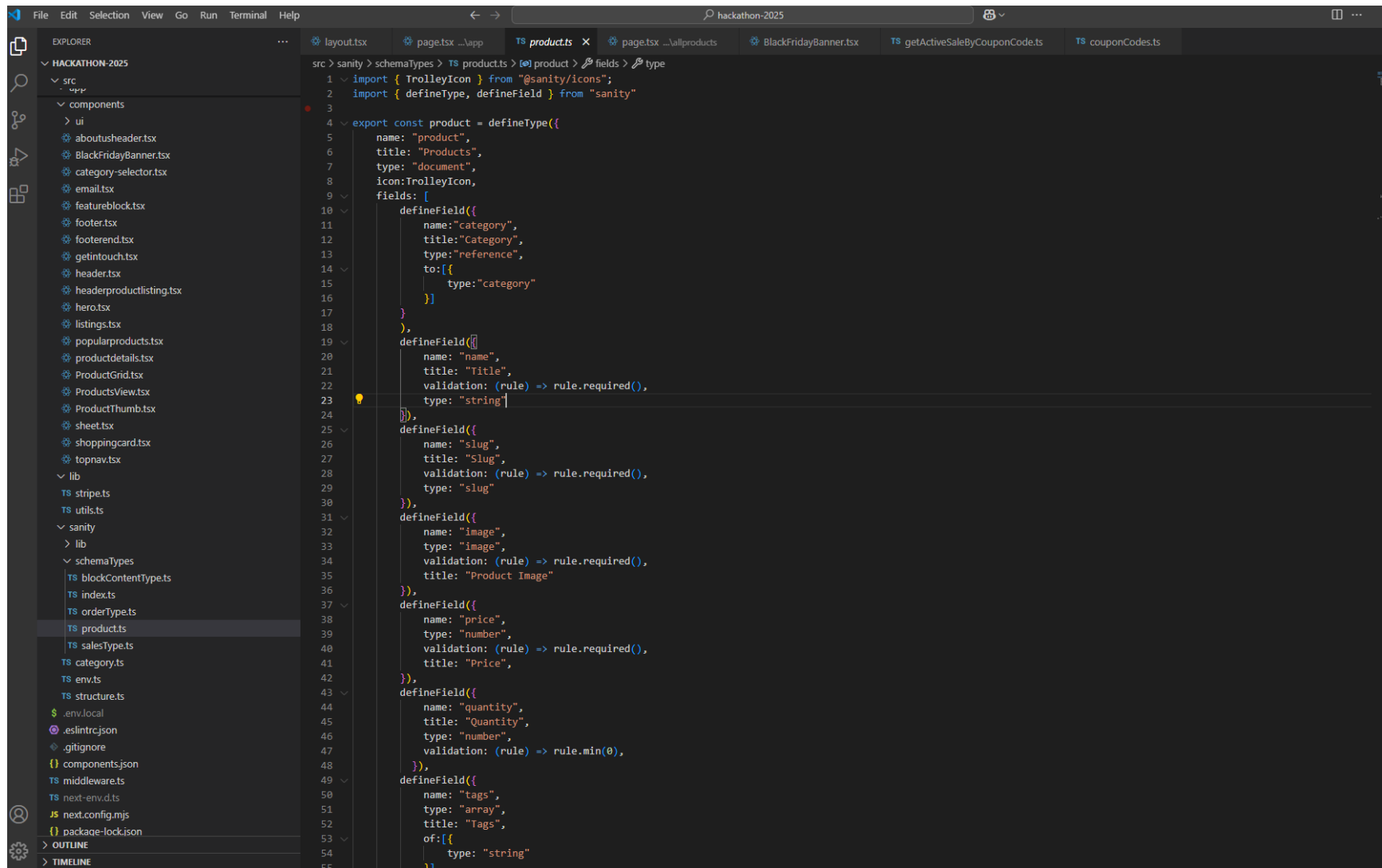
What's new

Sanity Create Content Mapping, Visual Editing, and Content Releases

TEMS OPERATIONAL

ights reserved.

# Product Schema Type



The screenshot shows a VS Code editor with a project named 'hackathon-2025'. The Explorer sidebar on the left shows a file tree with folders 'src' and 'sanity'. The 'sanity' folder contains 'lib' and 'schemaTypes'. The 'product.ts' file is selected under 'schemaTypes'. The main editor displays the code for the 'product' schema type, which is a Sanity schema type using the 'defineType' function. The code defines fields for 'category', 'name', 'slug', 'image', 'price', 'quantity', and 'tags'.

```
src > sanity > schemaTypes > TS product.ts > [0] product > fields > type
1 import { TrolleyIcon } from "@sanity/icons";
2 import { defineType, defineField } from "sanity"
3
4 export const product = defineType({
5   name: "product",
6   title: "Products",
7   type: "document",
8   icon: TrolleyIcon,
9   fields: [
10    defineField({
11      name: "category",
12      title: "Category",
13      type: "reference",
14      to: [{
15        type: "category"
16      }]
17    }),
18    defineField({
19      name: "name",
20      title: "Title",
21      validation: (rule) => rule.required(),
22      type: "string"
23    }),
24    defineField({
25      name: "slug",
26      title: "Slug",
27      validation: (rule) => rule.required(),
28      type: "slug"
29    }),
30    defineField({
31      name: "image",
32      type: "image",
33      validation: (rule) => rule.required(),
34      title: "Product Image"
35    }),
36    defineField({
37      name: "price",
38      type: "number",
39      validation: (rule) => rule.required(),
40      title: "Price",
41    }),
42    defineField({
43      name: "quantity",
44      title: "Quantity",
45      type: "number",
46      validation: (rule) => rule.min(0),
47    }),
48    defineField({
49      name: "tags",
50      type: "array",
51      title: "Tags",
52      of: [{
53        type: "string"
54      }]
55    })
56  ]
57 })
```

File Edit Selection View Go Run Terminal Help

hackathon-2025

EXPLORER

- HACKATHON-2025
  - src
    - components
      - ui
        - aboutusheader.tsx
        - BlackFridayBanner.tsx
        - category-selector.tsx
        - email.tsx
        - featureblock.tsx
        - footer.tsx
        - footerend.tsx
        - getintouch.tsx
        - header.tsx
        - headerproductlisting.tsx
        - hero.tsx
        - listings.tsx
        - popularproducts.tsx
        - productdetails.tsx
        - ProductGrid.tsx
        - ProductsView.tsx
        - ProductThumb.tsx
        - sheet.tsx
        - shoppingcard.tsx
        - topnav.tsx
      - lib
        - stripe.ts
        - utils.ts
      - sanity
        - lib
          - schemaTypes
            - TS blockContentType.ts
            - TS index.ts
            - TS orderType.ts
            - TS product.ts
            - TS salesType.ts
            - TS category.ts
            - TS env.ts
            - TS structure.ts
      - .env.local
      - .eslintrc.json
      - .gitignore
      - components.json
      - middleware.ts
      - TS next-env.d.ts
      - JS next.config.mjs
      - package-lock.json
    - OUTLINE
    - TIMELINE

layout.tsx page.tsx ...lapp TS salesType.ts X page.tsx ...lallproducts BlackFridayBanner.tsx TS getActiveSaleByCouponCode.ts TS couponCodes.ts

```
src > sanity > schemaTypes > TS salesType.ts > fields
1 import { TagIcon } from "@sanity/icons";
2 import { defineField, defineType } from "sanity";
3
4
5 export const salesType = defineType({
6   name: 'sale',
7   title: 'Sale',
8   type: 'document',
9   icon: TagIcon,
10  fields: [
11    defineField({
12      name: 'title',
13      title: 'Sale Title',
14      type: 'string',
15      validation: (Rule) => Rule.required(),
16    }),
17    defineField({
18      name: 'description',
19      title: 'Sale Description',
20      type: 'text',
21    }),
22    defineField({
23      name: 'discountAmount',
24      title: 'Discount Amount',
25      type: 'number',
26      description: 'Amount off in percentage or fixed value',
27    }),
28    defineField({
29      name: 'couponCode',
30      title: 'Coupon Code',
31      type: 'string',
32    }),
33    defineField({
34      name: 'validFrom',
35      title: 'Valid From',
36      type: 'datetime',
37    }),
38    defineField({
39      name: 'validUntil',
40      title: 'Valid Until',
41      type: 'datetime',
42    }),
43    defineField({
44      name: 'isActive',
45      title: 'Is Active',
46      type: 'boolean',
47      description: 'Toggle to activate/deactivate the sale',
48      initialValue: true,
49    }),
50  ],
51  preview: {
52    select: {
53      title: 'title',
54      description: 'description',
55    },
56  },
57 });
```

EXPLORER

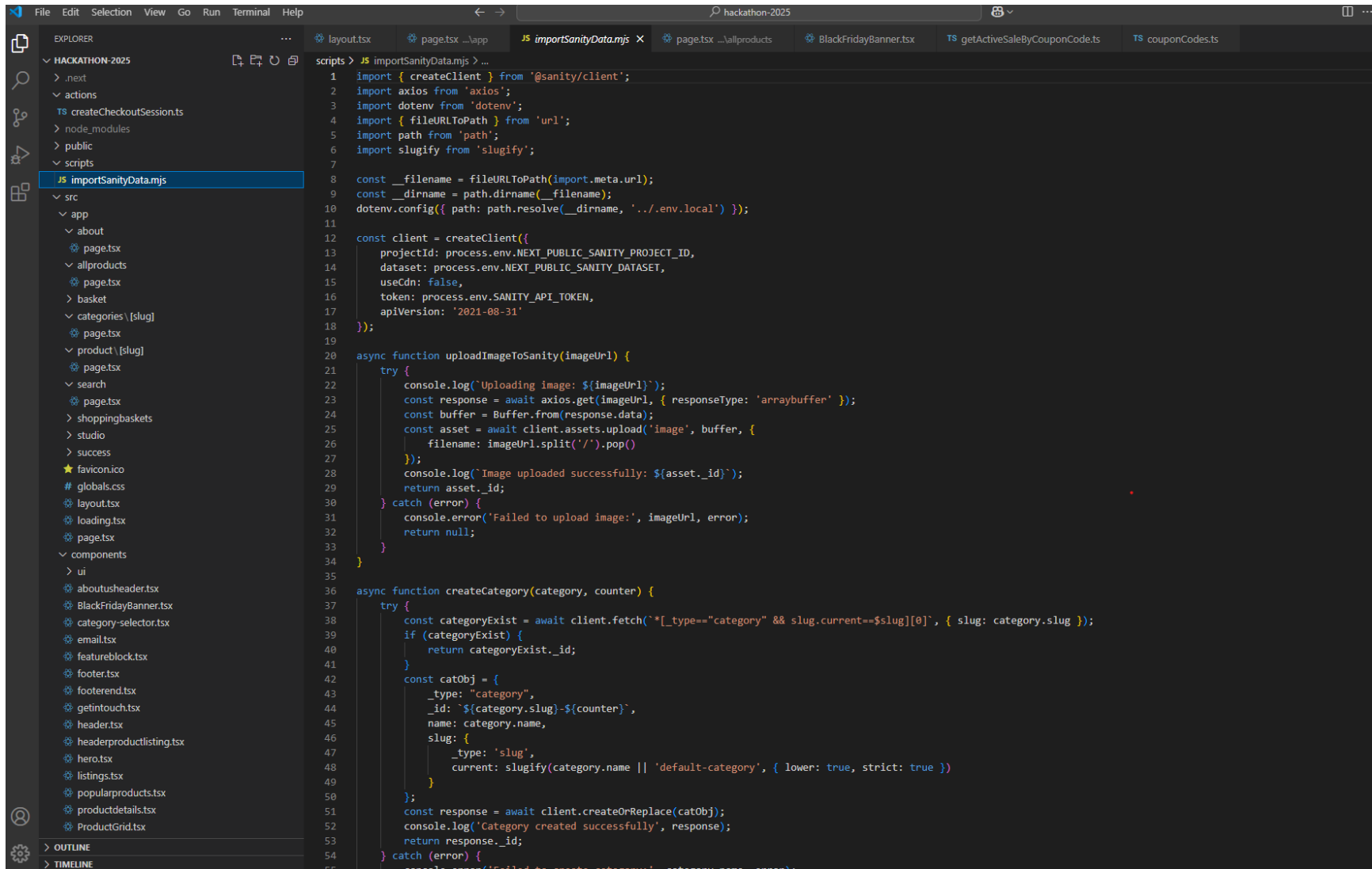
HACKATHON-2025

- src
  - components
    - ui
      - aboutusheader.tsx
      - BlackFridayBanner.tsx
      - category-selector.tsx
      - email.tsx
      - featureblock.tsx
      - footer.tsx
      - footerend.tsx
      - getintouch.tsx
      - header.tsx
      - headerproductlisting.tsx
      - hero.tsx
      - listings.tsx
      - popularproducts.tsx
      - productdetails.tsx
      - ProductGrid.tsx
      - ProductsView.tsx
      - ProductThumb.tsx
      - sheet.tsx
      - shoppingcard.tsx
      - topnav.tsx
  - lib
    - stripe.ts
    - utils.ts
  - sanity
    - lib
      - schemaTypes
        - blockContentType.ts
        - index.ts
        - orderType.ts
        - products.ts
        - salesType.ts
        - category.ts
        - env.ts
        - structure.ts
    - .env.local
    - .eslintrc.json
    - .gitignore
    - components.json
    - middleware.ts
    - next-env.d.ts
    - next.config.mjs
    - package-lock.json
  - OUTLINE
  - TIMELINE

src > sanity > schemaTypes > TS orderType.ts > orderType

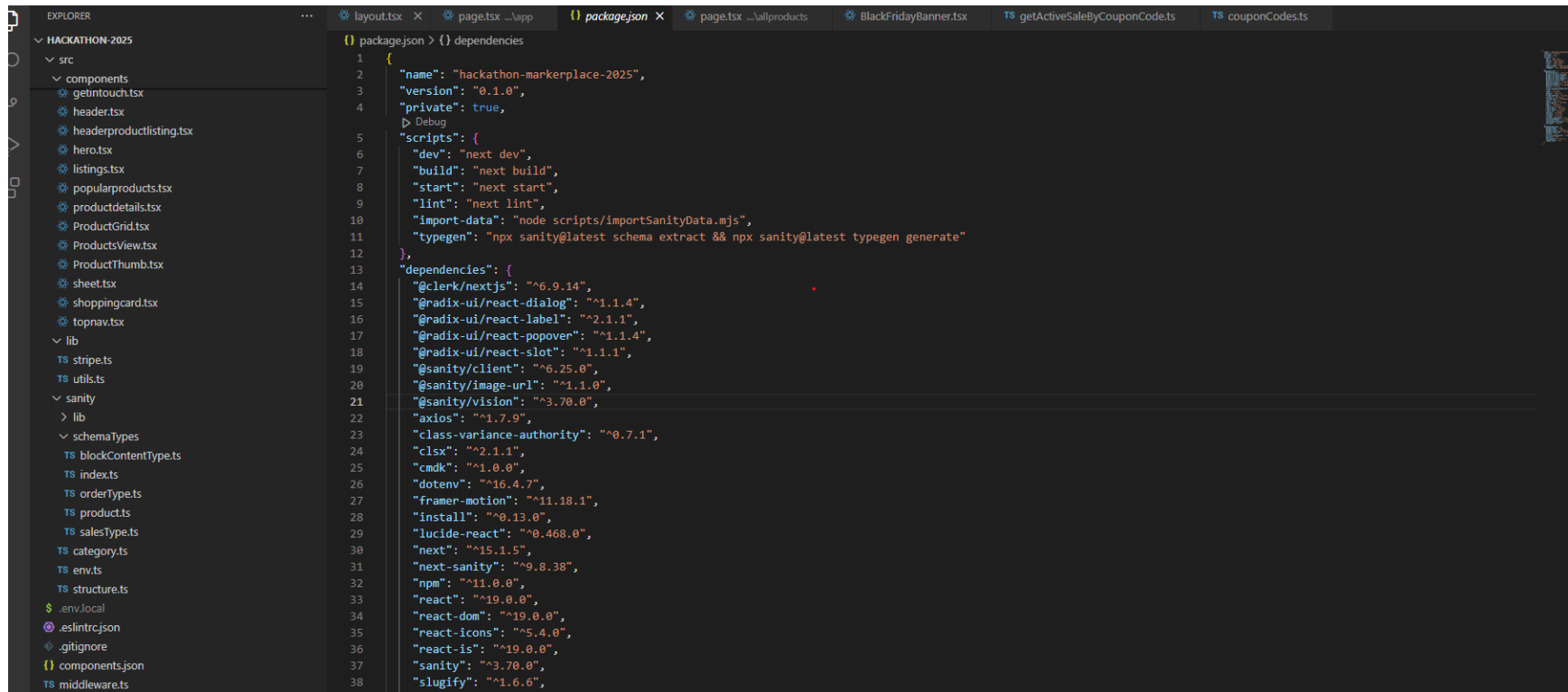
```
1 import { defineArrayMember, defineField, defineType } from "sanity";
2 import { BasketIcon } from "@sanity/icons";
3
4
5 export const orderType = defineType({
6   name: 'order',
7   title: 'Order',
8   type: 'document',
9   icon: BasketIcon,
10  fields: [
11    defineField({
12      name: 'orderNumber',
13      title: 'Order Number',
14      type: 'string',
15      validation: (Rule) => Rule.required(),
16    }),
17    defineField({
18      name: 'stripeCheckoutSessionId',
19      title: 'Stripe Checkout Session ID',
20      type: 'string',
21    }),
22    defineField({
23      name: 'stripeCustomerId',
24      title: 'Stripe Customer ID',
25      type: 'string',
26      validation: (Rule) => Rule.required(),
27    }),
28    defineField({
29      name: 'clerkUserId',
30      title: 'Store User ID',
31      type: 'string',
32      validation: (Rule) => Rule.required(),
33    }),
34    defineField({
35      name: 'customerName',
36      title: 'Customer Name',
37      type: 'string',
38      validation: (Rule) => Rule.required(),
39    }),
40    defineField({
41      name: 'email',
42      title: 'Customer Email',
43      type: 'string',
44      validation: (Rule) => Rule.required(),
45    }),
46    defineField({
47      name: 'stripePaymentIntentId',
48      title: 'Stripe Payment Intent ID',
49      type: 'string',
50      validation: (Rule) => Rule.required(),
51    }),
52    defineField({
53      name: 'products',
54      title: 'Products',
```

# Import Data into Sanity Template-02 using ImportScript



The screenshot shows a VS Code editor window with the file explorer on the left and the code editor in the center. The file explorer shows a project named 'HACKATHON-2025' with a 'scripts' folder containing 'importSanityData.mjs'. The code editor displays the content of 'importSanityData.mjs', which is a JavaScript file for importing data into Sanity. The code includes imports for 'createClient', 'axios', 'dotenv', 'fileURLToPath', 'path', and 'slugify'. It defines a client configuration and two async functions: 'uploadImageToSanity' and 'createCategory'.

```
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 import slugify from 'slugify';
7
8 const __filename = fileURLToPath(import.meta.url);
9 const __dirname = path.dirname(__filename);
10 dotenv.config({ path: path.resolve(__dirname, '../.env.local') });
11
12 const client = createClient({
13   projectId: process.env.NEXT_PUBLIC_SANITY_PROJECT_ID,
14   dataset: process.env.NEXT_PUBLIC_SANITY_DATASET,
15   useCdn: false,
16   token: process.env.SANITY_API_TOKEN,
17   apiVersion: '2021-08-31'
18 });
19
20 async function uploadImageToSanity(imageUrl) {
21   try {
22     console.log(`Uploading image: ${imageUrl}`);
23     const response = await axios.get(imageUrl, { responseType: 'arraybuffer' });
24     const buffer = Buffer.from(response.data);
25     const asset = await client.assets.upload('image', buffer, {
26       filename: imageUrl.split('/').pop()
27     });
28     console.log(`Image uploaded successfully: ${asset._id}`);
29     return asset._id;
30   } catch (error) {
31     console.error('Failed to upload image:', imageUrl, error);
32     return null;
33   }
34 }
35
36 async function createCategory(category, counter) {
37   try {
38     const categoryExist = await client.fetch(`*[ _type=="category" && slug.current=="${category.slug}" ]`, { slug: category.slug });
39     if (categoryExist) {
40       return categoryExist._id;
41     }
42     const catObj = {
43       _type: "category",
44       _id: `${category.slug}-${counter}`,
45       name: category.name,
46       slug: {
47         _type: 'slug',
48         current: slugify(category.name || 'default-category', { lower: true, strict: true })
49       }
50     };
51     const response = await client.createOrReplace(catObj);
52     console.log('Category created successfully', response);
53     return response._id;
54   } catch (error) {
55     console.error('Failed to create category:', category.name, error);
56   }
57 }
```





## Fetching data into Next. Js Application

Filter by the Category ▾



### Bed

A timeless design, with premium materials features as one of our most popular and iconic pieces. The dandy chair is...

**\$250.00**



### Bold Nest

A timeless design, with premium materials features as one of our most popular and iconic pieces. The dandy chair is...

**\$260.00**



### Bright Space

A timeless design, with premium materials features as one of our most popular and iconic pieces. The dandy chair is...

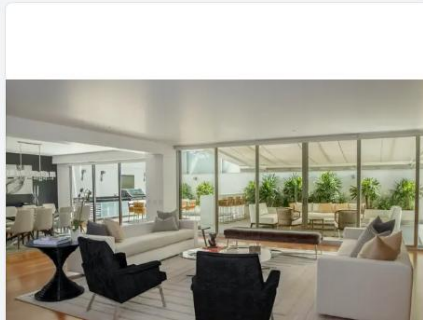
**\$180.00**



### Cloud Haven Chair

A timeless design, with premium materials features as one of our most popular and iconic pieces. The dandy chair is...

**\$230.00**



# Display Data in Sanity

S

Default

+ Create

Q

StructureVisionSchedules

Avion Ecommerce website

Category

Products

Order

Sale

Category

Search list

Crockory

Cutlery

Tables

Ceramics

Plant Pots
















Chairs

Tableware

Avion Ecommerce website

- 🏠 Category >
- 🛒 Products >**
- 📦 Order >
- 🏷️ Sale >

Products + ...

- 🔍 Search list
-  **Bed**  
\$250
  -  **Marble Ease**  
\$419
  -  **Vase Set**  
\$150
  -  **The Dandy chair**  
\$150
  -  **Bright Space**  
\$180
  -  **Cloud Haven Chair**  
\$230
  -  **Bold Nest**  
\$260
  -  **Sunny Chic**  
\$400
  -  **Reflective Haven**  
\$300
  -  **Modern Serenity**  
\$480
  -  **Timeless Elegance**  
\$320
  -  **TimberCraft**  
\$320
  -  **Nordic Elegance**  
\$280
  -  **Serene Seat**  
\$350
  -  **Sleek Living**  
\$300
  -  **Tropical Vibe**  
\$550
  -  **Zen Table**  
\$250