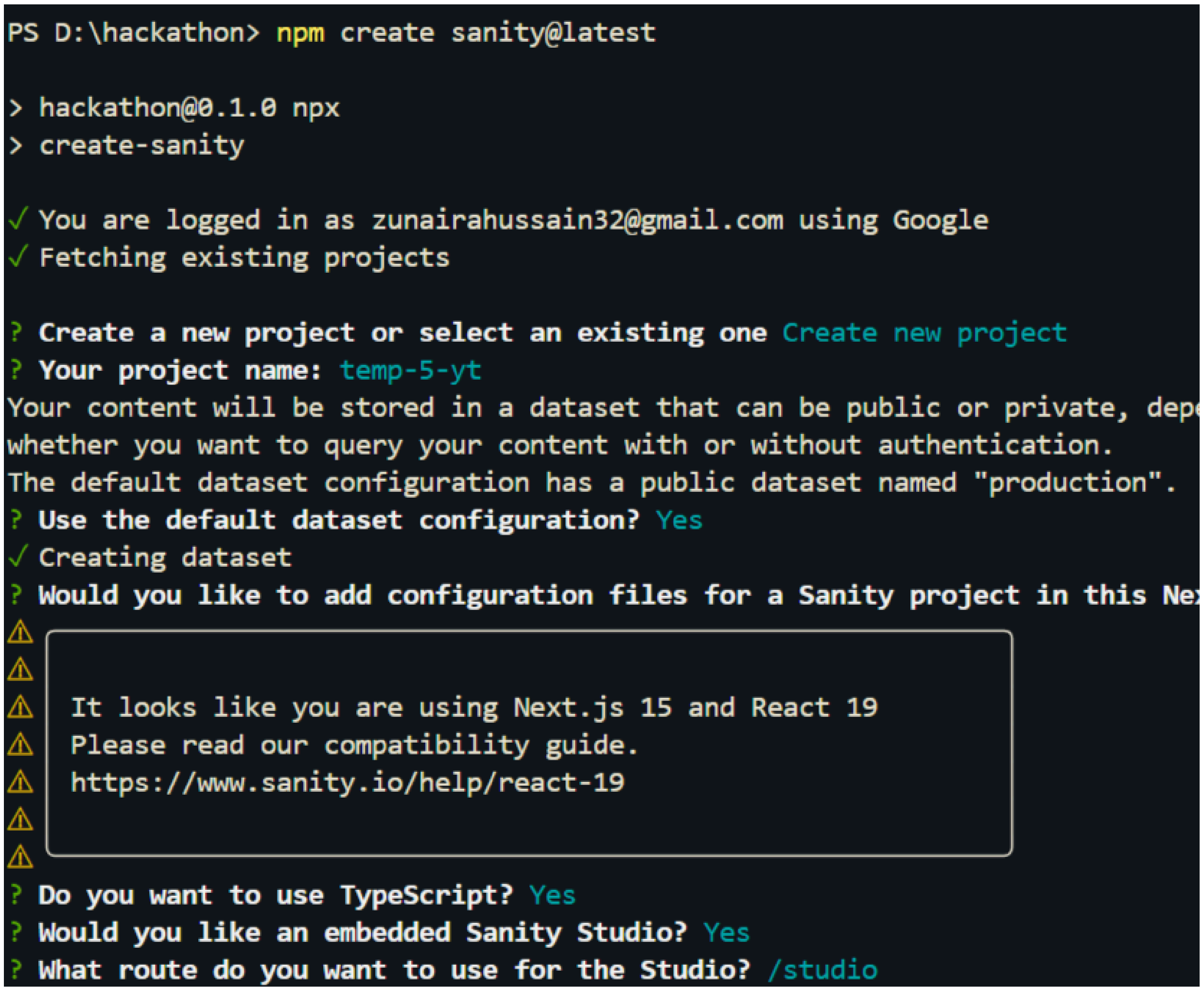
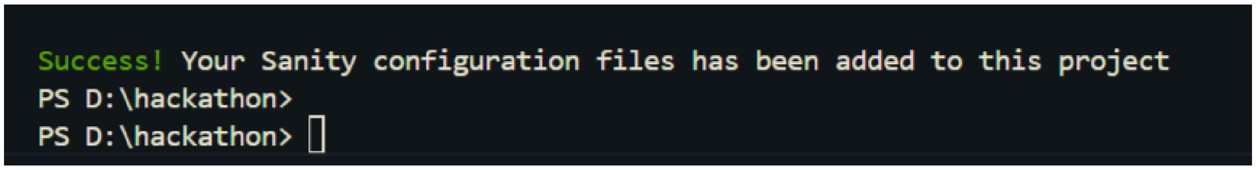
Day 3 - API Integration Report - CLOTHING AND BAG

1.API INTEGRATING PROCESS

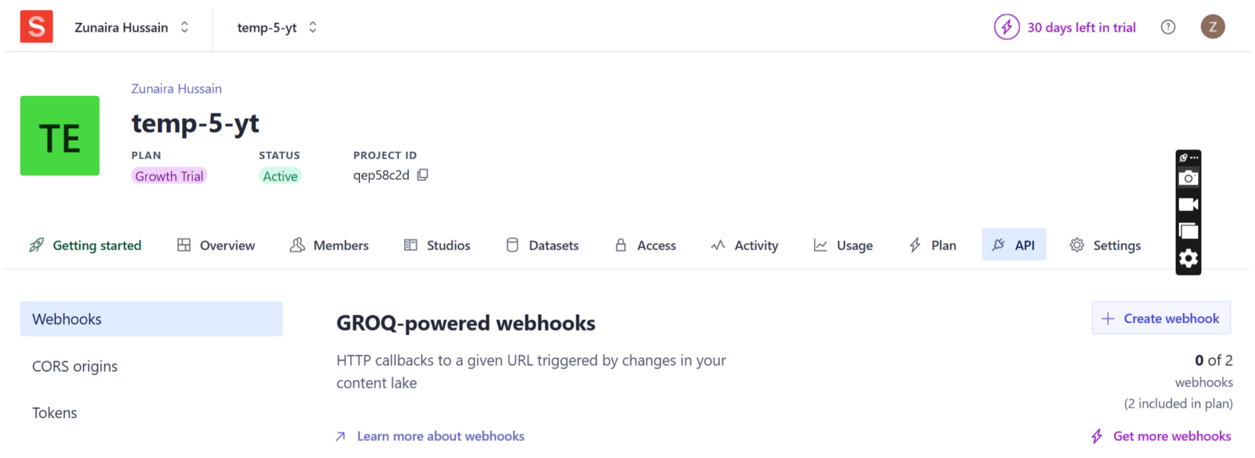


```
PS D:\hackathon> npm create sanity@latest

> hackathon@0.1.0 npx
> create-sanity

✓ You are logged in as zunairahussain32@gmail.com using Google
✓ Fetching existing projects

? Create a new project or select an existing one Create new project
? Your project name: temp-5-yt
Your content will be stored in a dataset that can be public or private, depe
whether you want to query your content with or without authentication.
The default dataset configuration has a public dataset named "production".
? Use the default dataset configuration? Yes
✓ Creating dataset
? Would you like to add configuration files for a Sanity project in this Ne
⚠
⚠
⚠   It looks like you are using Next.js 15 and React 19
⚠   Please read our compatibility guide.
⚠   https://www.sanity.io/help/react-19
⚠
⚠
? Do you want to use TypeScript? Yes
? Would you like an embedded Sanity Studio? Yes
? What route do you want to use for the Studio? /studio
```



```
Success! Your Sanity configuration files has been added to this project
PS D:\hackathon>
PS D:\hackathon>
```

2.SANITY



Zunaira Hussain · temp-5-yt · 30 days left in trial

Zunaira Hussain

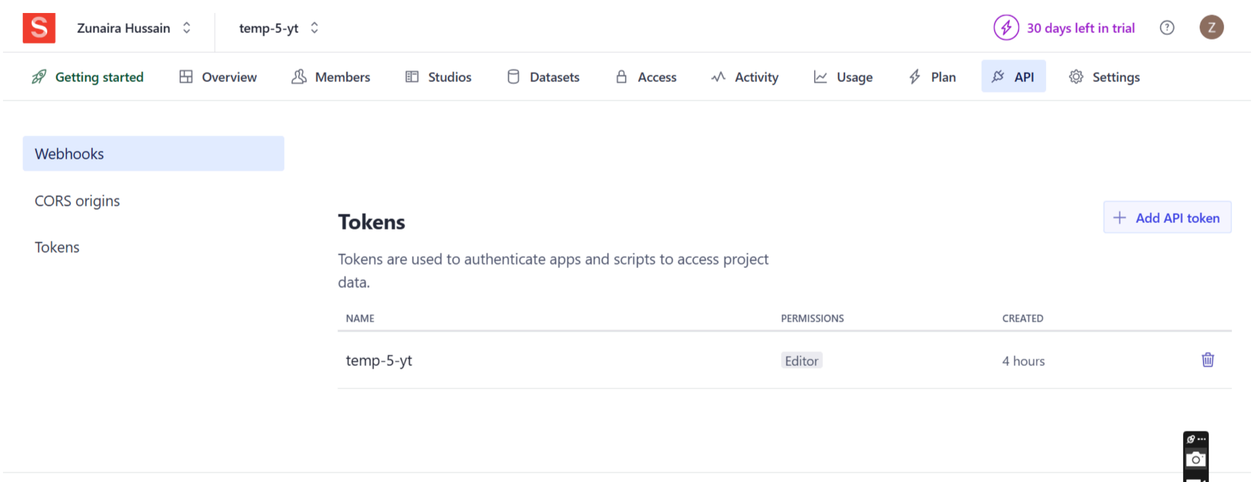
temp-5-yt

PLAN: Growth Trial
STATUS: Active
PROJECT ID: qep58c2d

GROQ-powered webhooks

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

0 of 2 webhooks (2 included in plan)

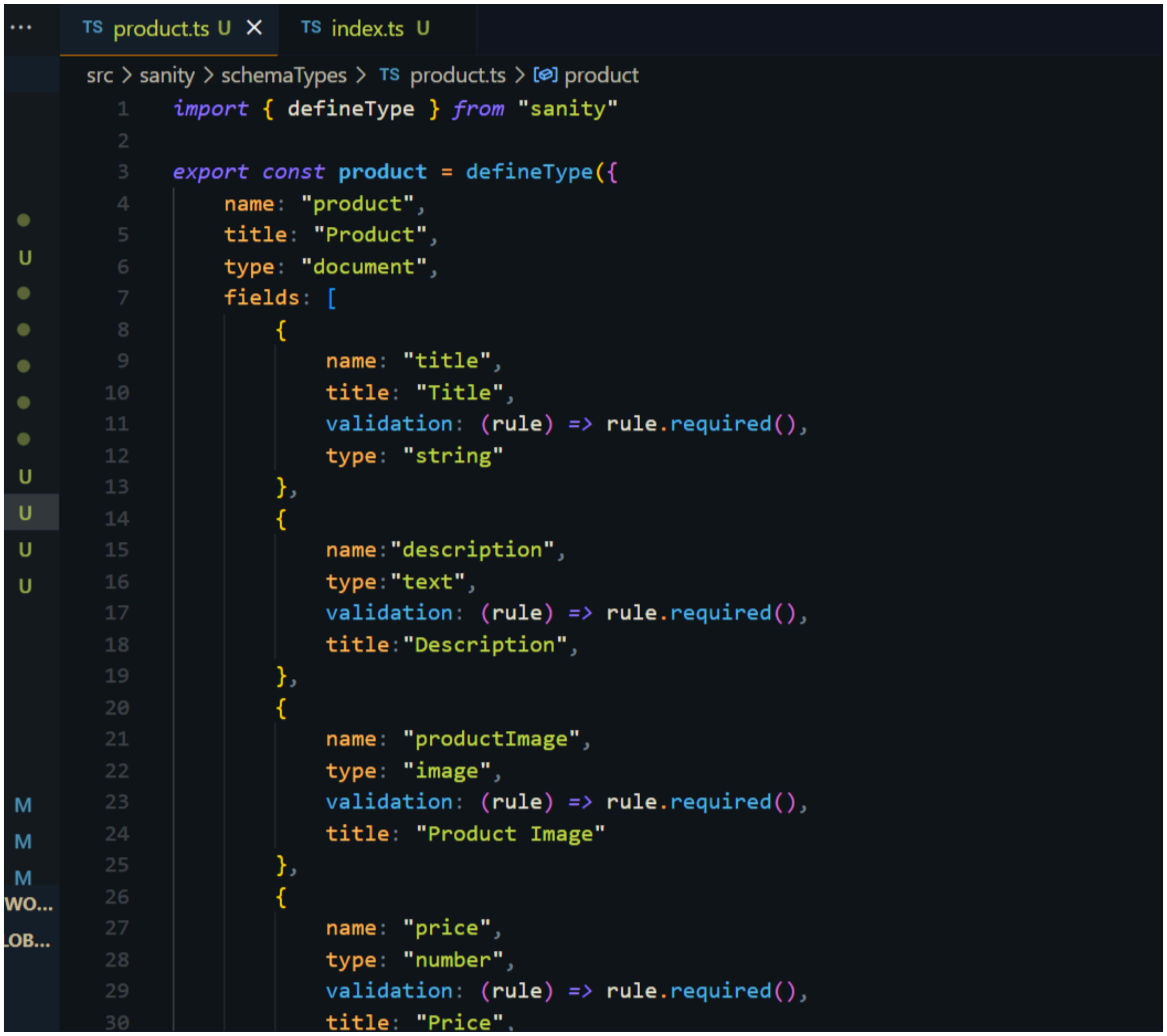


Zunaira Hussain · temp-5-yt · 30 days left in trial

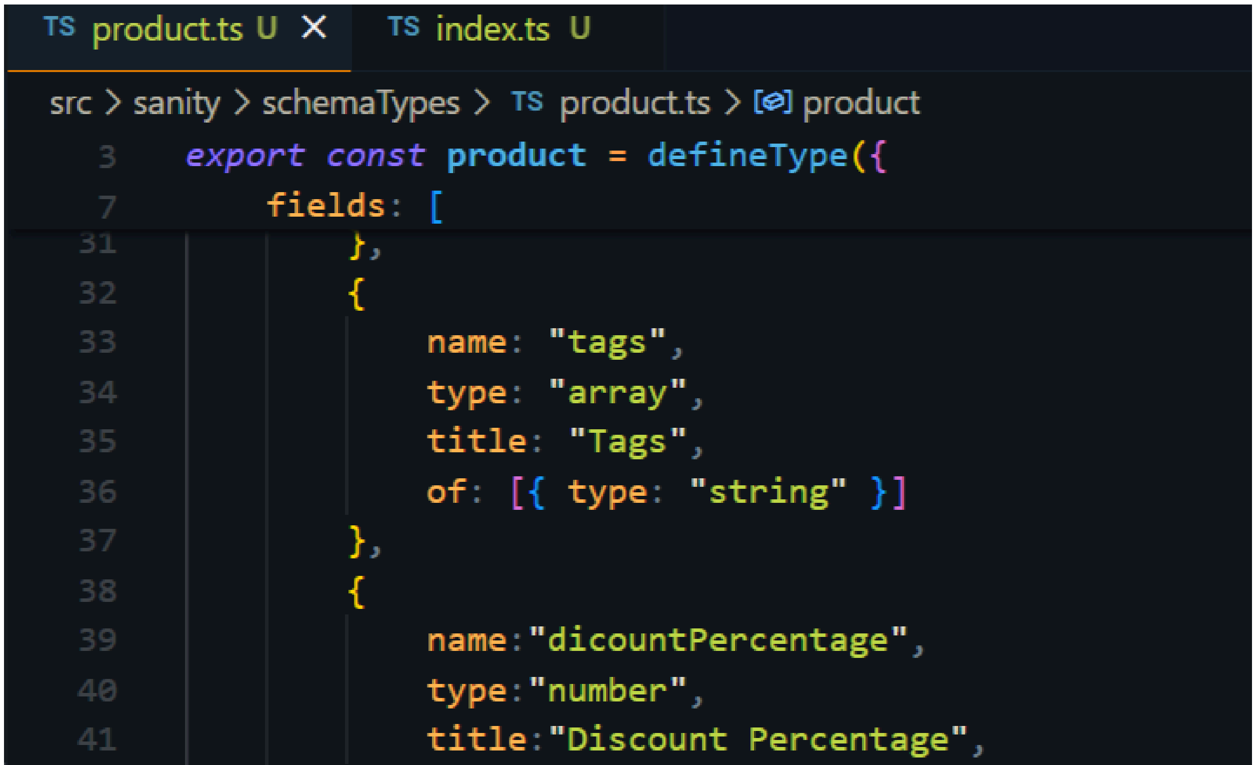
Tokens

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

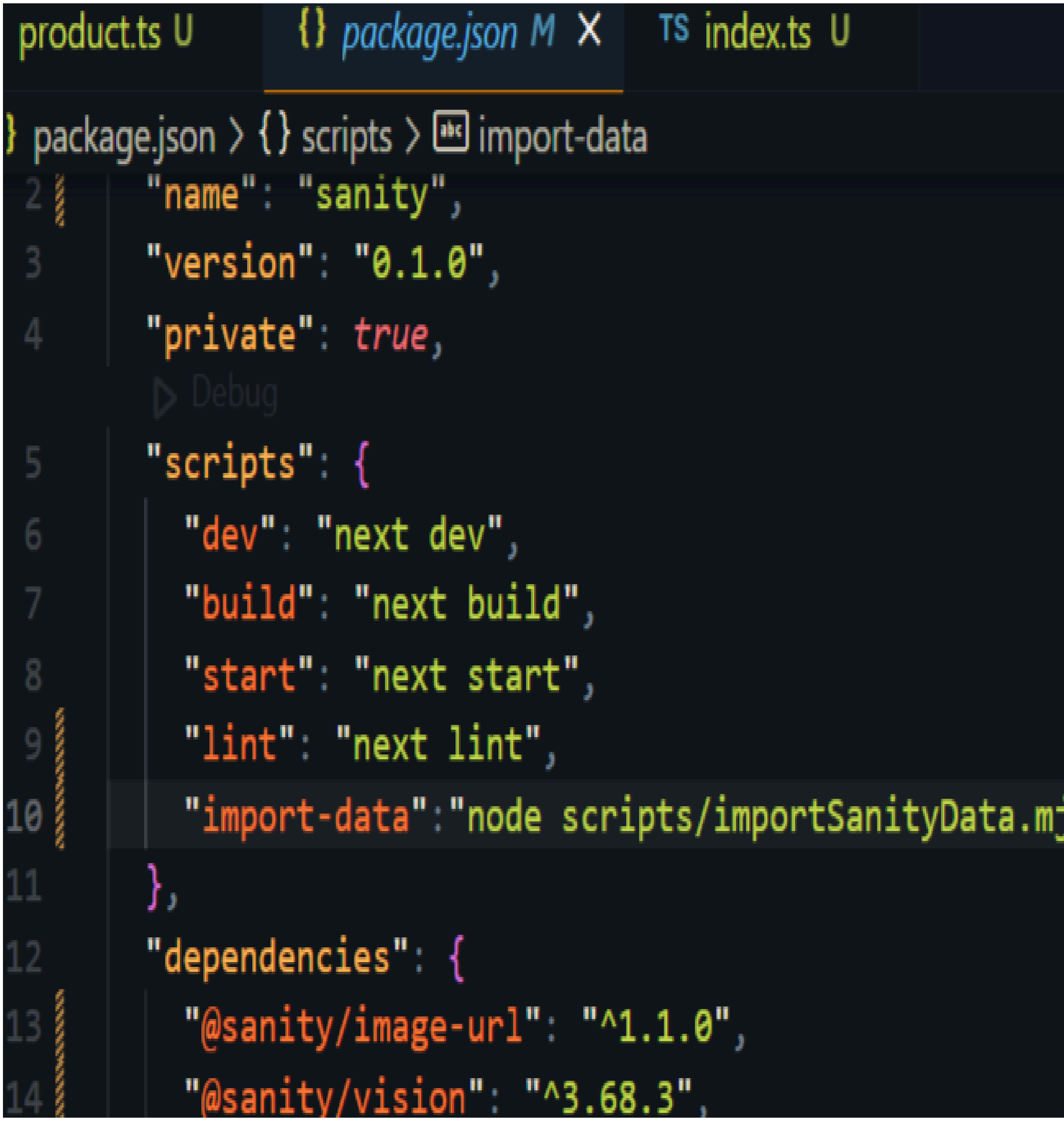
NAME	PERMISSIONS	CREATED
temp-5-yt	Editor	4 hours



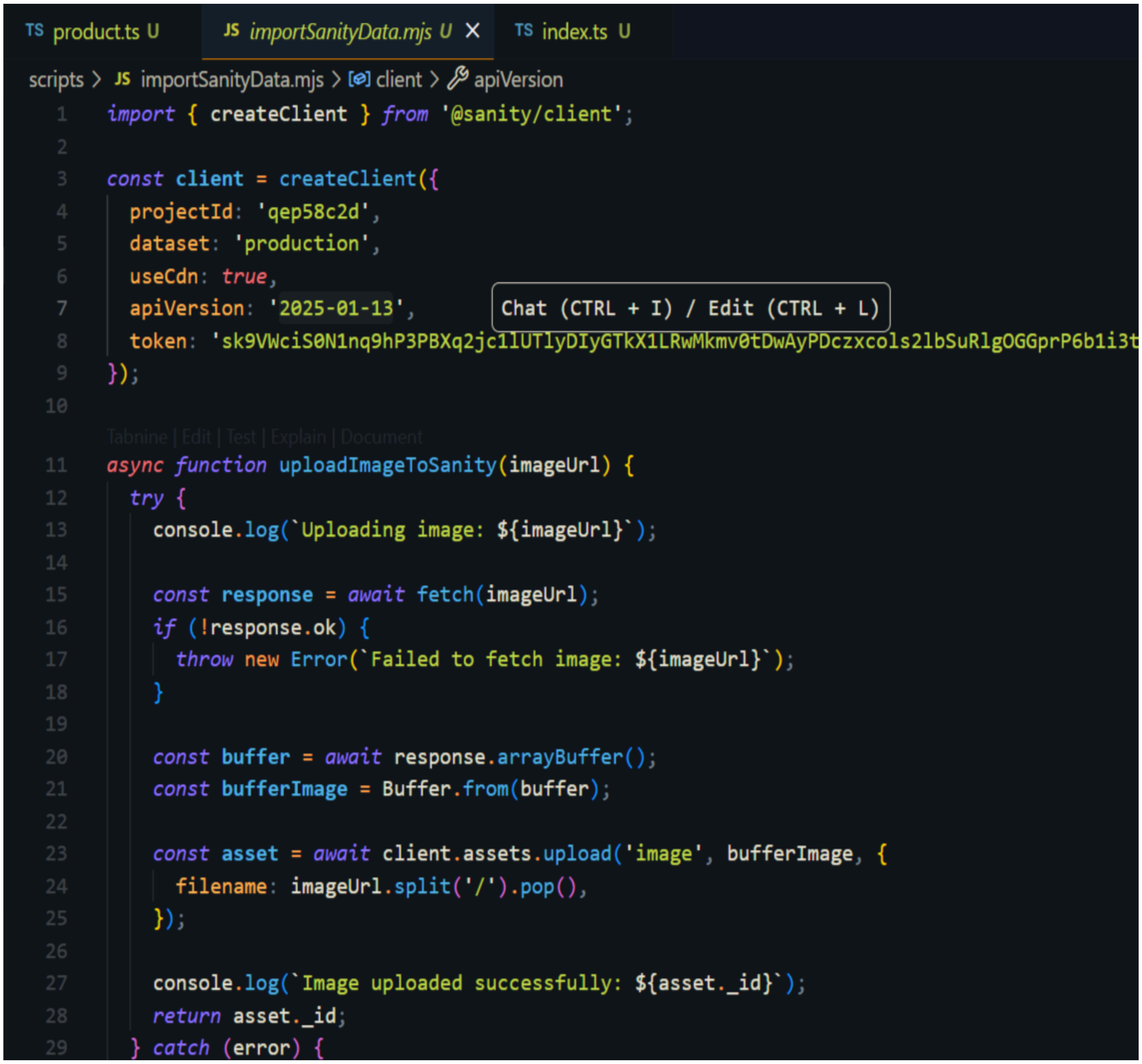
```
import { defineType } from "sanity"

export const product = defineType({
    name: "product",
    title: "Product",
    type: "document",
    fields: [
        {
            name: "title",
            title: "Title",
            validation: (rule) => rule.required(),
            type: "string"
        },
        {
            name:"description",
            type:"text",
            validation: (rule) => rule.required(),
            title:"Description",
        },
        {
            name: "productImage",
            type: "image",
            validation: (rule) => rule.required(),
            title: "Product Image"
        },
        {
            name: "price",
            type: "number",
            validation: (rule) => rule.required(),
            title: "Price",
```



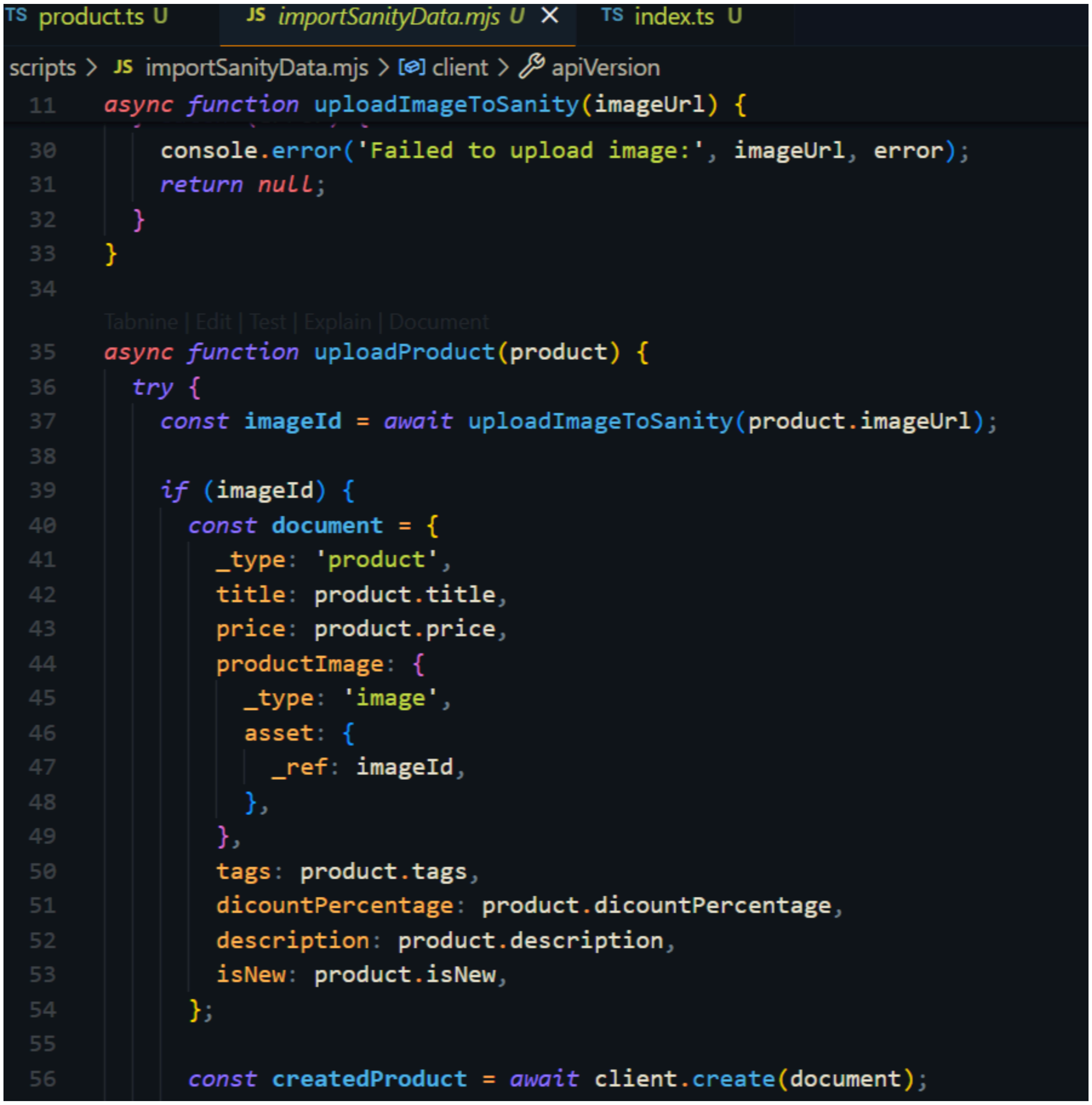
```
export const product = defineType({
    fields: [
        },
        {
            name: "tags",
            type: "array",
            title: "Tags",
            of: [{ type: "string" }]
        },
        {
            name:"dicountPercentage",
            type:"number",
            title:"Discount Percentage",
```



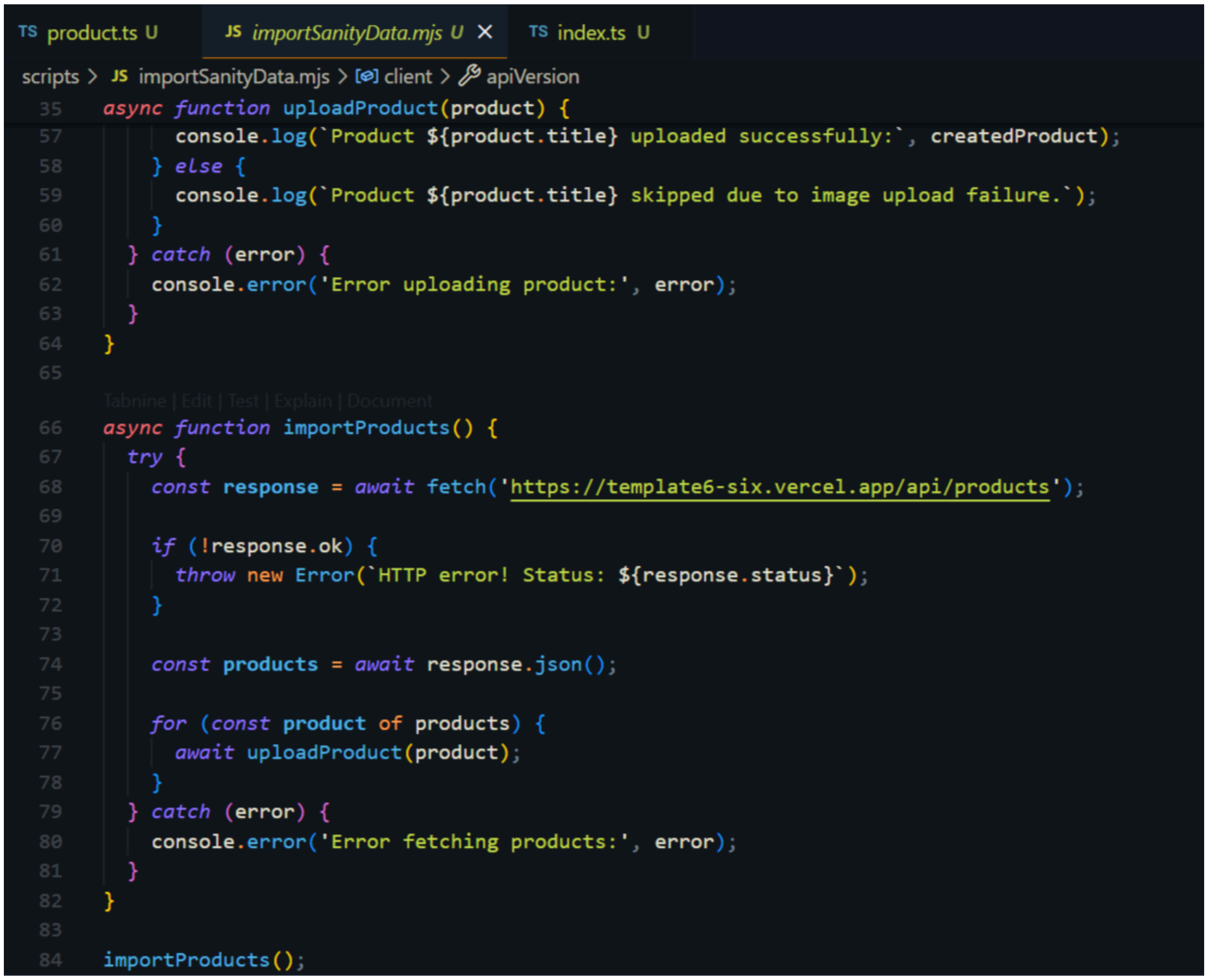
```
  "name": "sanity",
  "version": "0.1.0",
  "private": true,
  "scripts": {
    "dev": "next dev",
    "build": "next build",
    "start": "next start",
    "lint": "next lint",
    "import-data":"node scripts/importSanityData.mj
  },
  "dependencies": {
    "@sanity/image-url": "^1.1.0",
    "@sanity/vision": "^3.68.3",
```


TS product.ts U
JS importSanityData.mjs U
TS index.ts U
scripts > JS importSanityData.mjs > client > apiVersion
Chat (CTRL + I) / Edit (CTRL + L)
Tabnine | Edit | Test | Explain | Document

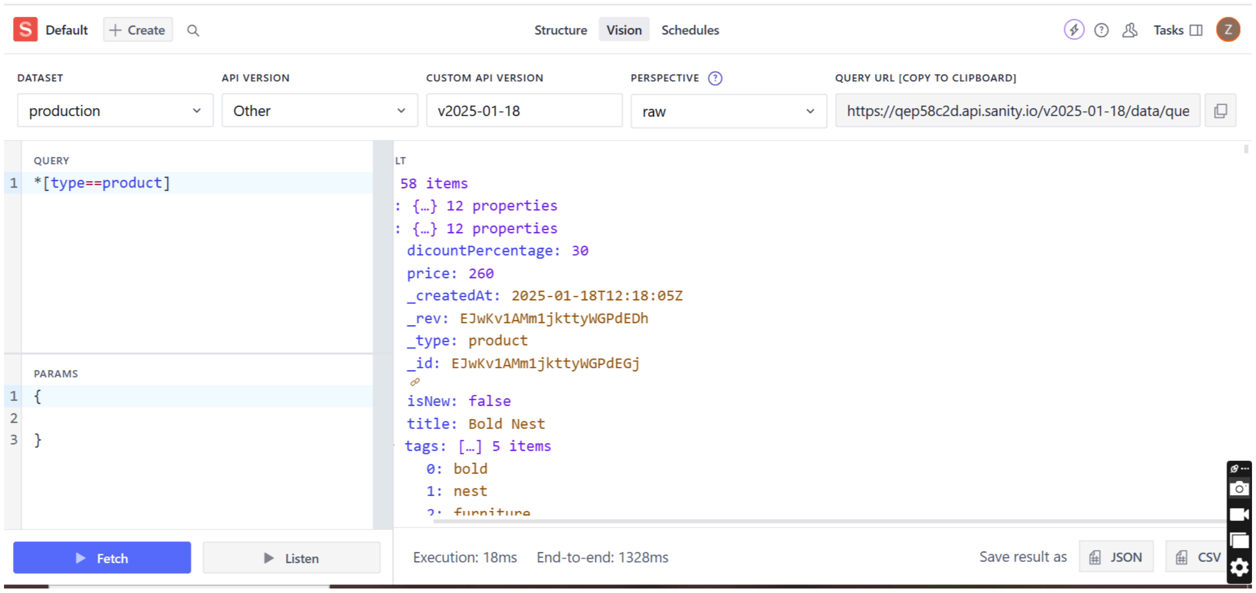
```javascript
import { createClient } from '@sanity/client';

const client = createClient({
  projectId: 'qep58c2d',
  dataset: 'production',
  useCdn: true,
  apiVersion: '2025-01-13',
  token: 'sk9VWciS0N1nq9hP3PBXq2jc1lUTlyDIyGTkX1LRwMkmv0tDwAyPDczxcols2lbSuRlgOGGprP6b1i3t
});

async function uploadImageToSanity(imageUrl) {
  try {
    console.log(`Uploading image: ${imageUrl}`);

    const response = await fetch(imageUrl);
    if (!response.ok) {
      throw new Error(`Failed to fetch image: ${imageUrl}`);
    }

    const buffer = await response.arrayBuffer();
    const bufferImage = Buffer.from(buffer);

    const asset = await client.assets.upload('image', bufferImage, {
      filename: imageUrl.split('/').pop(),
    });

    console.log(`Image uploaded successfully: ${asset._id}`);
    return asset._id;
  } catch (error) {
```


TS product.ts U
JS importSanityData.mjs U
TS index.ts U
scripts > JS importSanityData.mjs > client > apiVersion
Tabnine | Edit | Test | Explain | Document

```javascript
async function uploadImageToSanity(imageUrl) {
    console.error('Failed to upload image:', imageUrl, error);
    return null;
  }
}

async function uploadProduct(product) {
  try {
    const imageId = await uploadImageToSanity(product.imageUrl);

    if (imageId) {
      const document = {
        _type: 'product',
        title: product.title,
        price: product.price,
        productImage: {
          _type: 'image',
          asset: {
            _ref: imageId,
          },
        },
        tags: product.tags,
        dicountPercentage: product.dicountPercentage,
        description: product.description,
        isNew: product.isNew,
      };

      const createdProduct = await client.create(document);
```


TS product.ts U
JS importSanityData.mjs U
TS index.ts U
scripts > JS importSanityData.mjs > client > apiVersion
Tabnine | Edit | Test | Explain | Document

```
async function uploadProduct(product) {
      console.log(`Product ${product.title} uploaded successfully:`, createdProduct);
    } else {
      console.log(`Product ${product.title} skipped due to image upload failure.`);
    }
  } catch (error) {
    console.error('Error uploading product:', error);
  }
}

async function importProducts() {
  try {
    const response = await fetch('https://template6-six.vercel.app/api/products');

    if (!response.ok) {
      throw new Error(`HTTP error! Status: ${response.status}`);
    }

    const products = await response.json();

    for (const product of products) {
      await uploadProduct(product);
    }
  } catch (error) {
    console.error('Error fetching products:', error);
  }
}

importProducts();
```



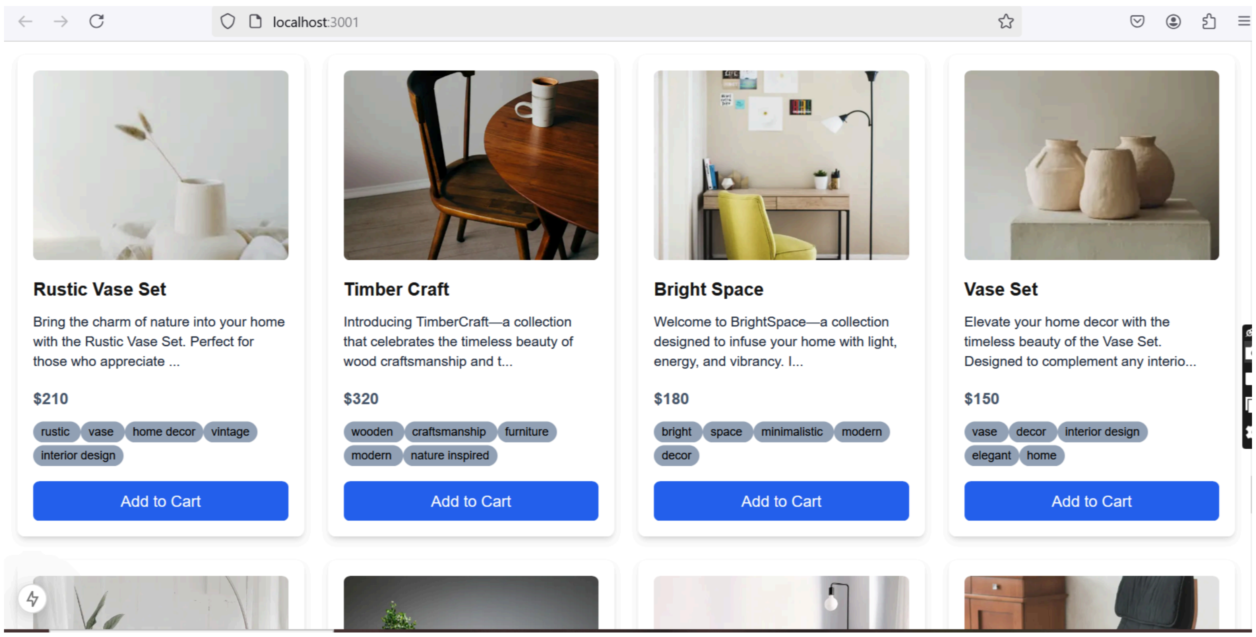
DATASET: production
API VERSION: Other
CUSTOM API VERSION: v2025-01-18
PERSPECTIVE: raw
QUERY URL [COPY TO CLIPBOARD]: https://qep58c2d.api.sanity.io/v2025-01-18/data/que

QUERY
```
*[type==product]
```

PARAMS
```
{

}
```

```
LT
58 items
: {…} 12 properties
: {…} 12 properties
 dicountPercentage: 30
 price: 260
 _createdAt: 2025-01-18T12:18:05Z
 _rev: EJwKv1AMm1jkttyWGPdEDh
 _type: product
 _id: EJwKv1AMm1jkttyWGPdEGj
 isNew: false
 title: Bold Nest
 tags: […] 5 items
    0: bold
    1: nest
    2: furniture
```

Execution: 18ms End-to-end: 1328ms



Rustic Vase Set

Bring the charm of nature into your home with the Rustic Vase Set. Perfect for those who appreciate ...

$210

rustic vase home decor vintage interior design

Add to Cart

Timber Craft

Introducing TimberCraft—a collection that celebrates the timeless beauty of wood craftsmanship and t...

$320

wooden craftsmanship furniture modern nature inspired

Add to Cart

Bright Space

Welcome to BrightSpace—a collection designed to infuse your home with light, energy, and vibrancy. I...

$180

bright space minimalistic modern decor

Add to Cart

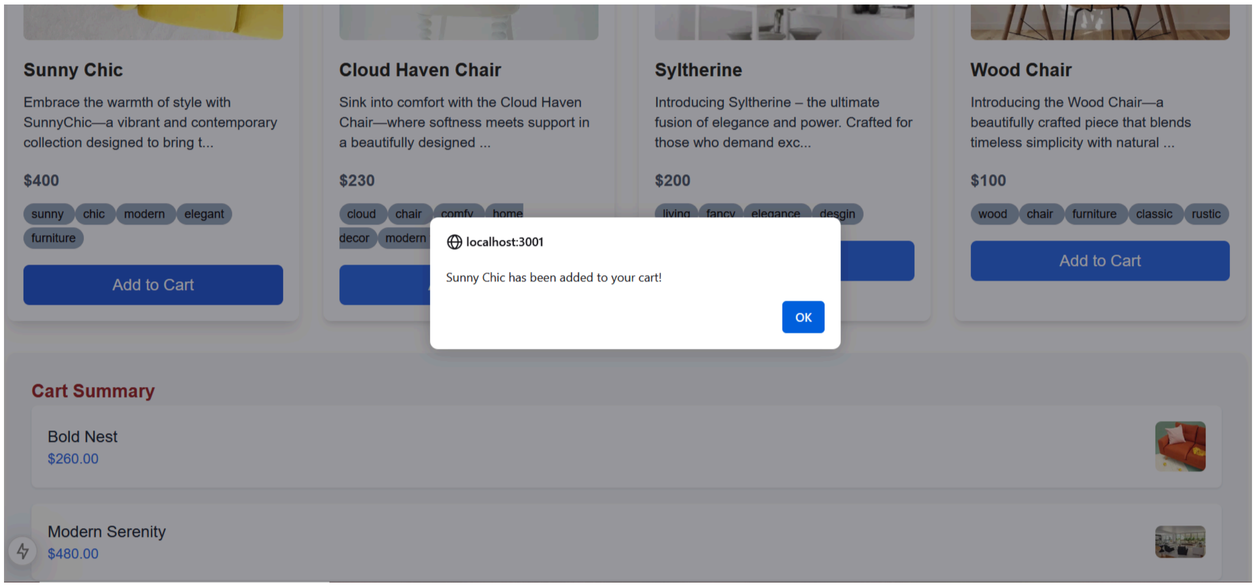
Vase Set

Elevate your home decor with the timeless beauty of the Vase Set. Designed to complement any interio...

$150

vase decor interior design elegant home

Add to Cart



Sunny Chic

Embrace the warmth of style with SunnyChic—a vibrant and contemporary collection designed to bring t...

$400

sunny chic modern elegant furniture

Add to Cart

Cloud Haven Chair

Sink into comfort with the Cloud Haven Chair—where softness meets support in a beautifully designed ...

$230

cloud chair comfy home decor modern

Syltherine

Introducing Syltherine – the ultimate fusion of elegance and power. Crafted for those who demand exc...

$200

living fancy elegance desgin

Wood Chair

Introducing the Wood Chair—a beautifully crafted piece that blends timeless simplicity with natural ...

$100

wood chair furniture classic rustic

Add to Cart

localhost:3001

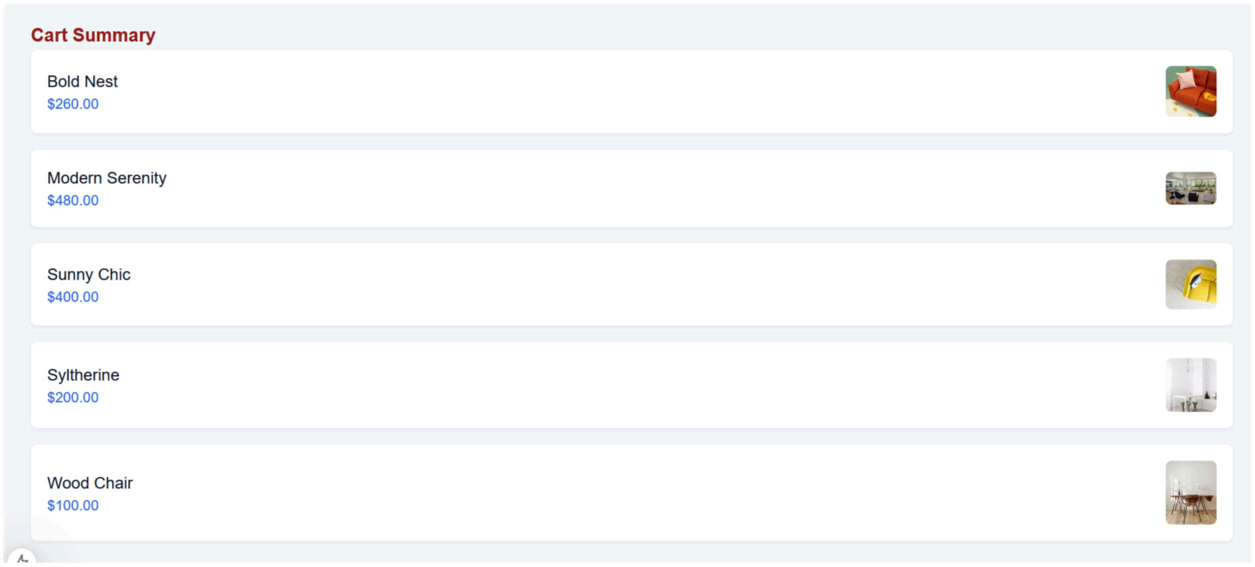
Sunny Chic has been added to your cart!

OK

Cart Summary

Bold Nest
$260.00

Modern Serenity
$480.00



Cart Summary

Bold Nest
$260.00

Modern Serenity
$480.00

Sunny Chic
$400.00

Syltherine
$200.00

Wood Chair
$100.00