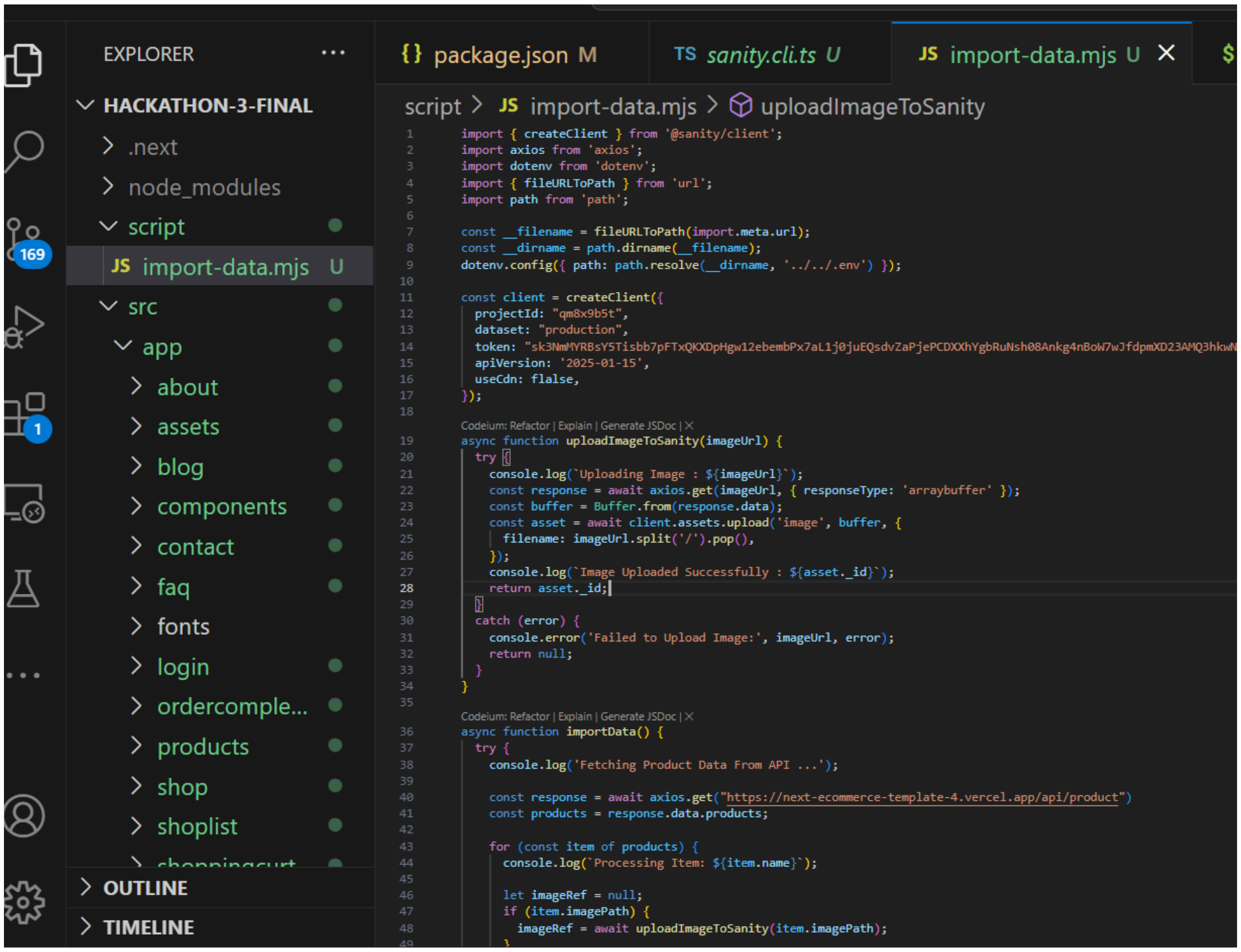
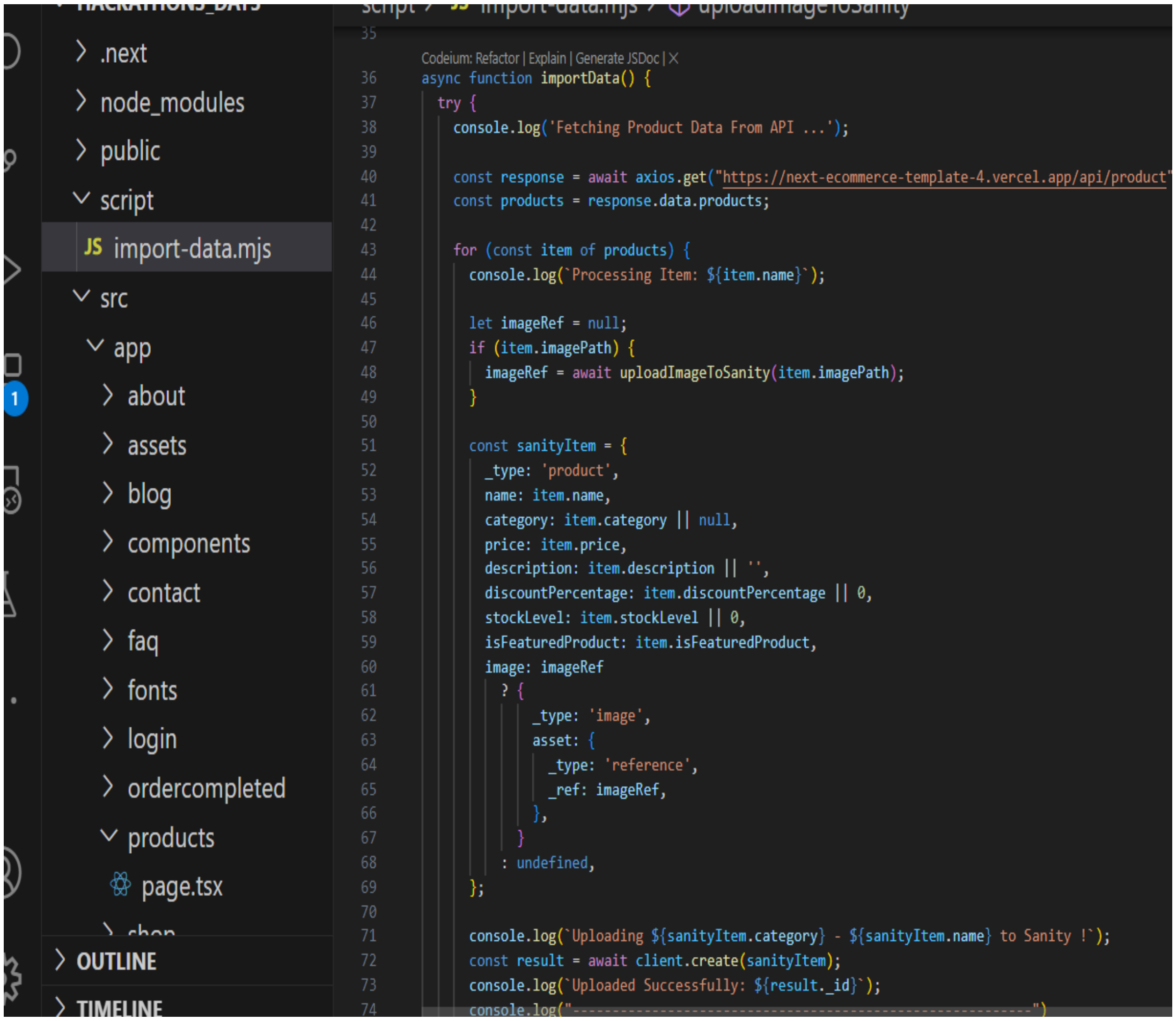
Day 3 – API Integration and Data Migration:

API Integration Process

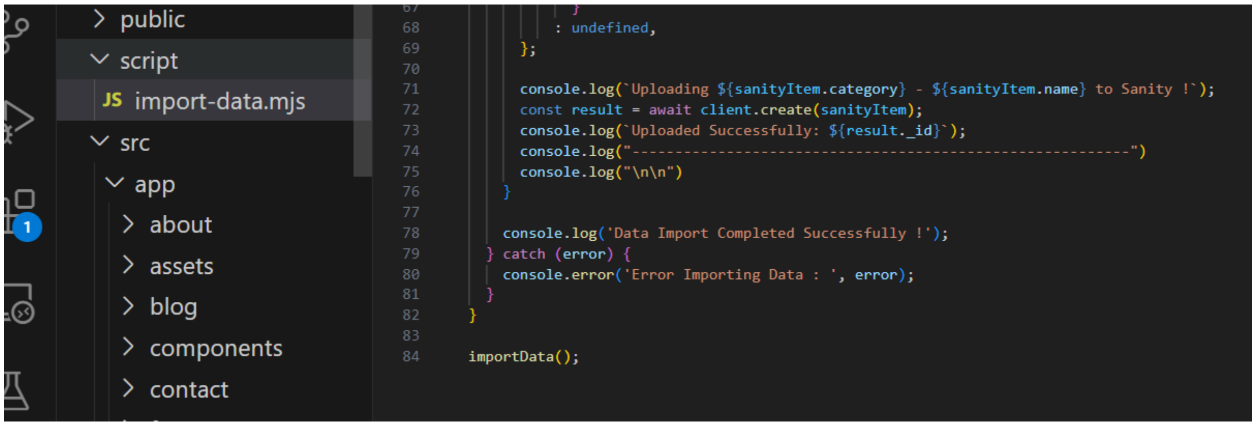
1. **Reviewed API Documentation**
   - Examined the provided API documentation to understand its structure and available endpoints.
   - Identified the necessary endpoints for retrieving and managing data effectively.
2. **Executed Data Migration Script**
   - Ran a custom script tailored to migrate data from the API into Sanity CMS.
   - Confirmed the migration's success by verifying the populated fields in Sanity CMS.



```
import { createClient } from '@sanity/client';
import axios from 'axios';
import dotenv from 'dotenv';
import { fileURLToPath } from 'url';
import path from 'path';

const __filename = fileURLToPath(import.meta.url);
const __dirname = path.dirname(__filename);
dotenv.config({ path: path.resolve(__dirname, '../../.env') });

const client = createClient({
  projectId: "qm8x9b5t",
  dataset: "production",
  token: "sk3NmMYRBsY5Tisbb7pFTxQKXDpHgw12ebembPx7aL1j0juEQsdvZaPjePCDXXhYgbRuNsh08Ankg4nBoW7wJfdpmXD23AMQ3hkwN
  apiVersion: '2025-01-15',
  useCdn: flalse,
});

Codeium: Refactor | Explain | Generate JSDoc | ×
async function uploadImageToSanity(imageUrl) {
  try {
    console.log(`Uploading Image : ${imageUrl}`);
    const response = await axios.get(imageUrl, { responseType: 'arraybuffer' });
    const buffer = Buffer.from(response.data);
    const asset = await client.assets.upload('image', buffer, {
      filename: imageUrl.split('/').pop(),
    });
    console.log(`Image Uploaded Successfully : ${asset._id}`);
    return asset._id;
  }
  catch (error) {
    console.error('Failed to Upload Image:', imageUrl, error);
    return null;
  }
}

Codeium: Refactor | Explain | Generate JSDoc | ×
async function importData() {
  try {
    console.log('Fetching Product Data From API ...');

    const response = await axios.get("https://next-ecommerce-template-4.vercel.app/api/product")
    const products = response.data.products;

    for (const item of products) {
      console.log(`Processing Item: ${item.name}`);

      let imageRef = null;
      if (item.imagePath) {
        imageRef = await uploadImageToSanity(item.imagePath);
```

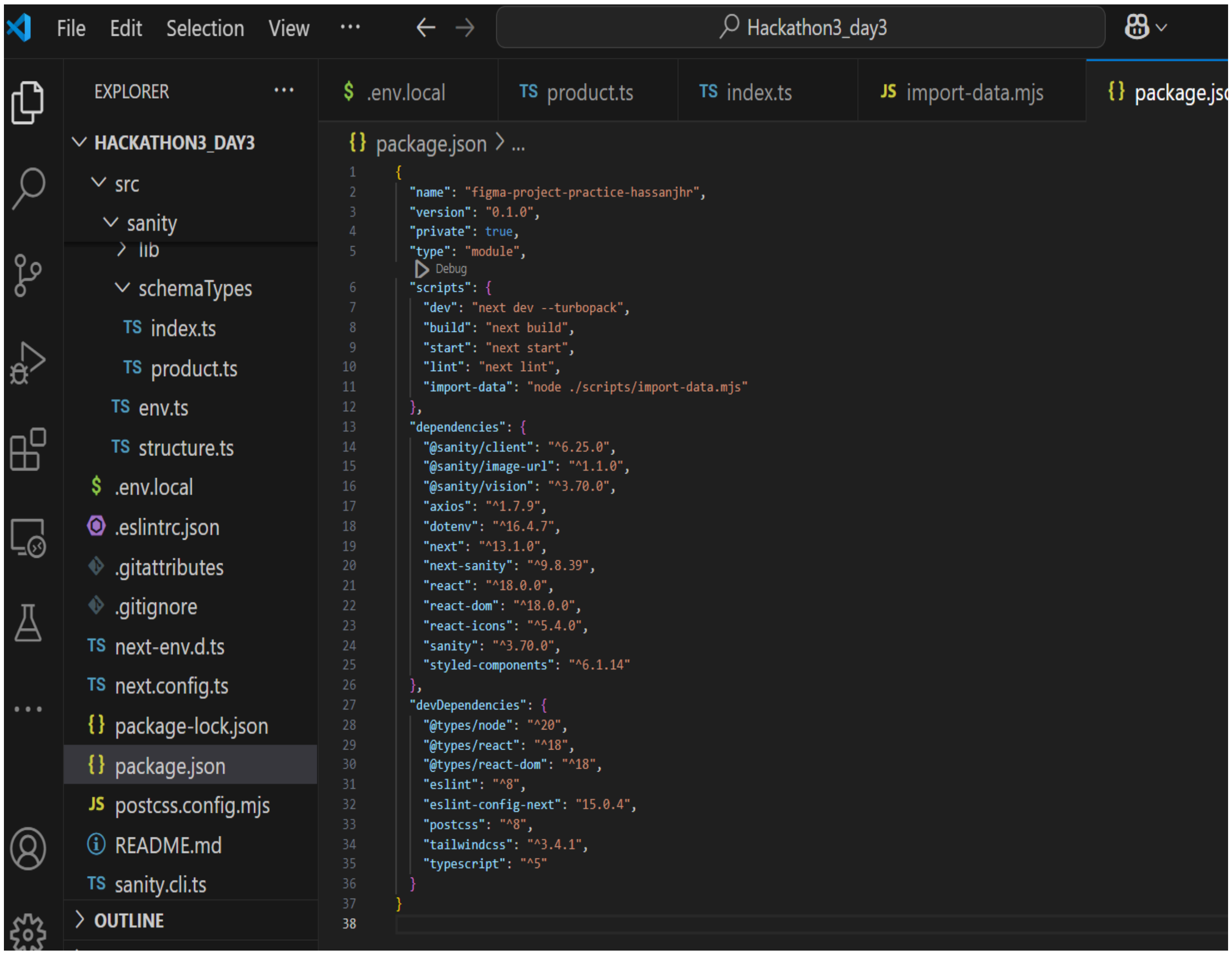


```
async function importData() {
  try {
    console.log('Fetching Product Data From API ...');

    const response = await axios.get("https://next-ecommerce-template-4.vercel.app/api/product"
    const products = response.data.products;

    for (const item of products) {
      console.log(`Processing Item: ${item.name}`);

      let imageRef = null;
      if (item.imagePath) {
        imageRef = await uploadImageToSanity(item.imagePath);
      }

      const sanityItem = {
        _type: 'product',
        name: item.name,
        category: item.category || null,
        price: item.price,
        description: item.description || '',
        discountPercentage: item.discountPercentage || 0,
        stockLevel: item.stockLevel || 0,
        isFeaturedProduct: item.isFeaturedProduct,
        image: imageRef
          ? {
              _type: 'image',
              asset: {
                _type: 'reference',
                _ref: imageRef,
              },
            }
          : undefined,
      };

      console.log(`Uploading ${sanityItem.category} - ${sanityItem.name} to Sanity !`);
      const result = await client.create(sanityItem);
      console.log(`Uploaded Successfully: ${result._id}`);
      console.log("----------------------------------------------------------")
```



```
            : undefined,
        };

        console.log(`Uploading ${sanityItem.category} - ${sanityItem.name} to Sanity !`);
        const result = await client.create(sanityItem);
        console.log(`Uploaded Successfully: ${result._id}`);
        console.log("-----------------------------------------------------------")
        console.log("\n\n")
      }

      console.log('Data Import Completed Successfully !');
    } catch (error) {
      console.error('Error Importing Data : ', error);
    }
  }

  importData();
```

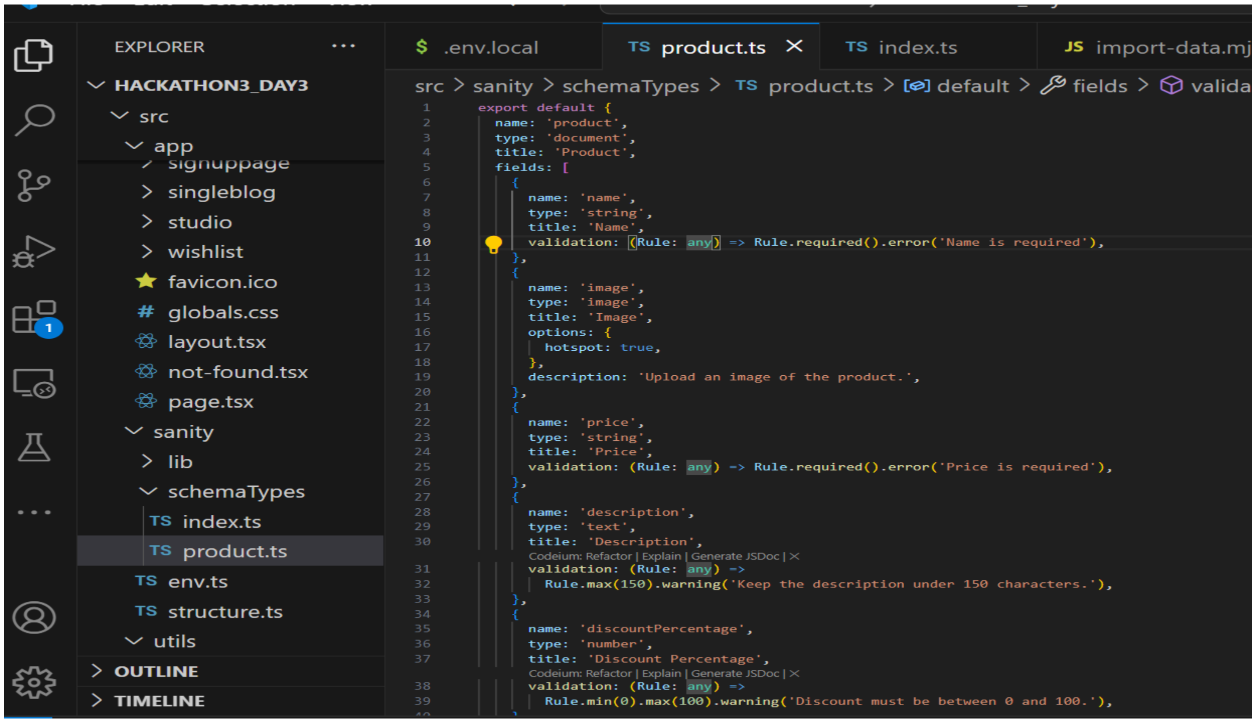
Changes in package.json



```
{
  "name": "figma-project-practice-hassanjhr",
  "version": "0.1.0",
  "private": true,
  "type": "module",
  "scripts": {
    "dev": "next dev --turbopack",
    "build": "next build",
    "start": "next start",
    "lint": "next lint",
    "import-data": "node ./scripts/import-data.mjs"
  },
  "dependencies": {
    "@sanity/client": "^6.25.0",
    "@sanity/image-url": "^1.1.0",
    "@sanity/vision": "^3.70.0",
    "axios": "^1.7.9",
    "dotenv": "^16.4.7",
    "next": "^13.1.0",
    "next-sanity": "^9.8.39",
    "react": "^18.0.0",
    "react-dom": "^18.0.0",
    "react-icons": "^5.4.0",
    "sanity": "^3.70.0",
    "styled-components": "^6.1.14"
  },
  "devDependencies": {
    "@types/node": "^20",
    "@types/react": "^18",
    "@types/react-dom": "^18",
    "eslint": "^8",
    "eslint-config-next": "15.0.4",
    "postcss": "^8",
    "tailwindcss": "^3.4.1",
    "typescript": "^5"
  }
}
```

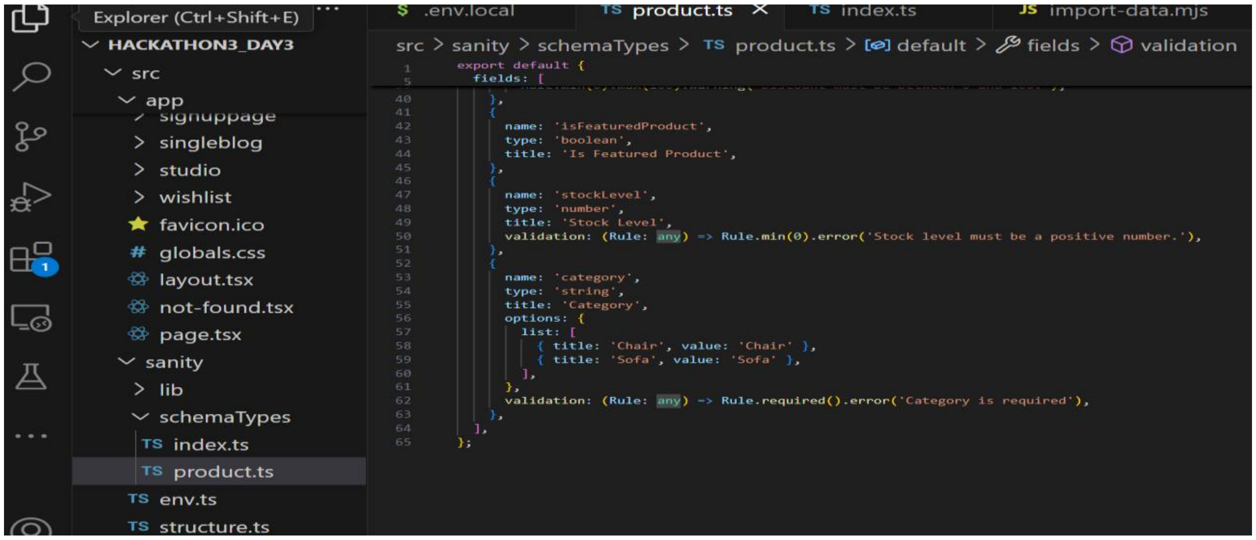
Product Schemas

- Made necessary modifications to the default schemas to align with the requirements of the migrated data.
- Updated the schema structure to ensure seamless compatibility with the API data format.



src > sanity > schemaTypes > product.ts > default > fields > valida

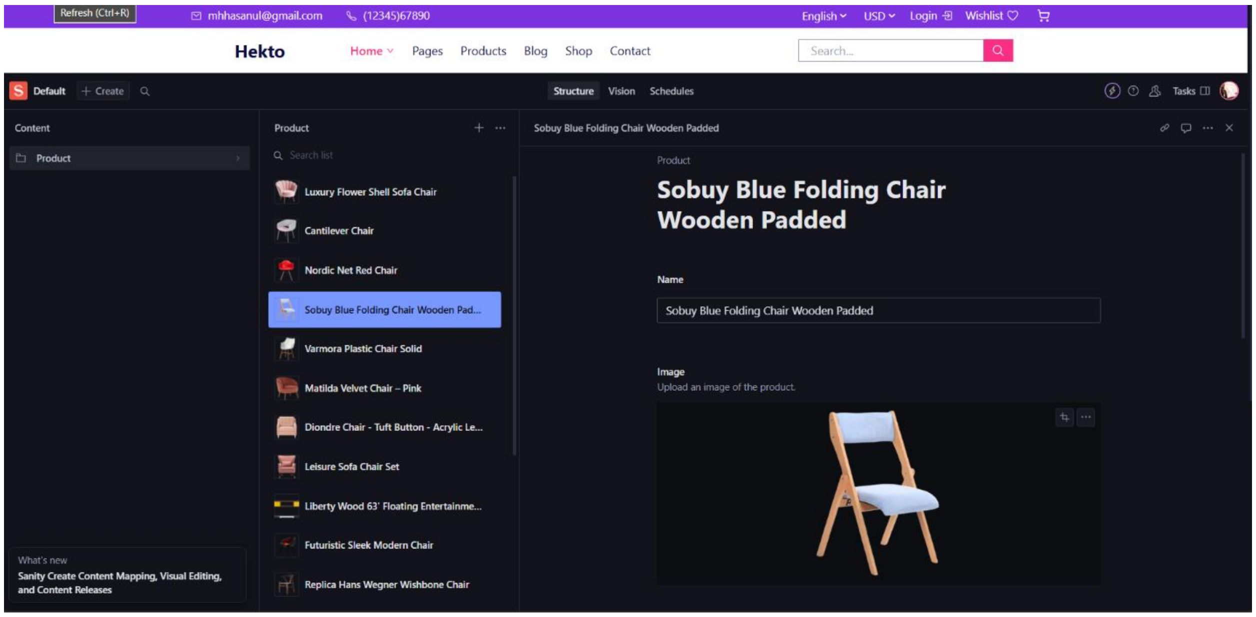
```ts
export default {
  name: 'product',
  type: 'document',
  title: 'Product',
  fields: [
    {
      name: 'name',
      type: 'string',
      title: 'Name',
      validation: (Rule: any) => Rule.required().error('Name is required'),
    },
    {
      name: 'image',
      type: 'image',
      title: 'Image',
      options: {
        hotspot: true,
      },
      description: 'Upload an image of the product.',
    },
    {
      name: 'price',
      type: 'string',
      title: 'Price',
      validation: (Rule: any) => Rule.required().error('Price is required'),
    },
    {
      name: 'description',
      type: 'text',
      title: 'Description',
      validation: (Rule: any) =>
        Rule.max(150).warning('Keep the description under 150 characters.'),
    },
    {
      name: 'discountPercentage',
      type: 'number',
      title: 'Discount Percentage',
      validation: (Rule: any) =>
        Rule.min(0).max(100).warning('Discount must be between 0 and 100.'),
```



src > sanity > schemaTypes > product.ts > default > fields > validation

```ts
export default {
  fields: [
    },
    {
      name: 'isFeaturedProduct',
      type: 'boolean',
      title: 'Is Featured Product',
    },
    {
      name: 'stockLevel',
      type: 'number',
      title: 'Stock Level',
      validation: (Rule: any) => Rule.min(0).error('Stock level must be a positive number.'),
    },
    {
      name: 'category',
      type: 'string',
      title: 'Category',
      options: {
        list: [
          { title: 'Chair', value: 'Chair' },
          { title: 'Sofa', value: 'Sofa' },
        ],
      },
      validation: (Rule: any) => Rule.required().error('Category is required'),
    },
  ],
};
```

Migrated Data in Sanity:



Product

- Luxury Flower Shell Sofa Chair
- Cantilever Chair
- Nordic Net Red Chair
- Sobuy Blue Folding Chair Wooden Pad...
- Varmora Plastic Chair Solid
- Matilda Velvet Chair – Pink
- Diondre Chair - Tuft Button - Acrylic Le...
- Leisure Sofa Chair Set
- Liberty Wood 63' Floating Entertainme...
- Futuristic Sleek Modern Chair
- Replica Hans Wegner Wishbone Chair

Sobuy Blue Folding Chair Wooden Padded

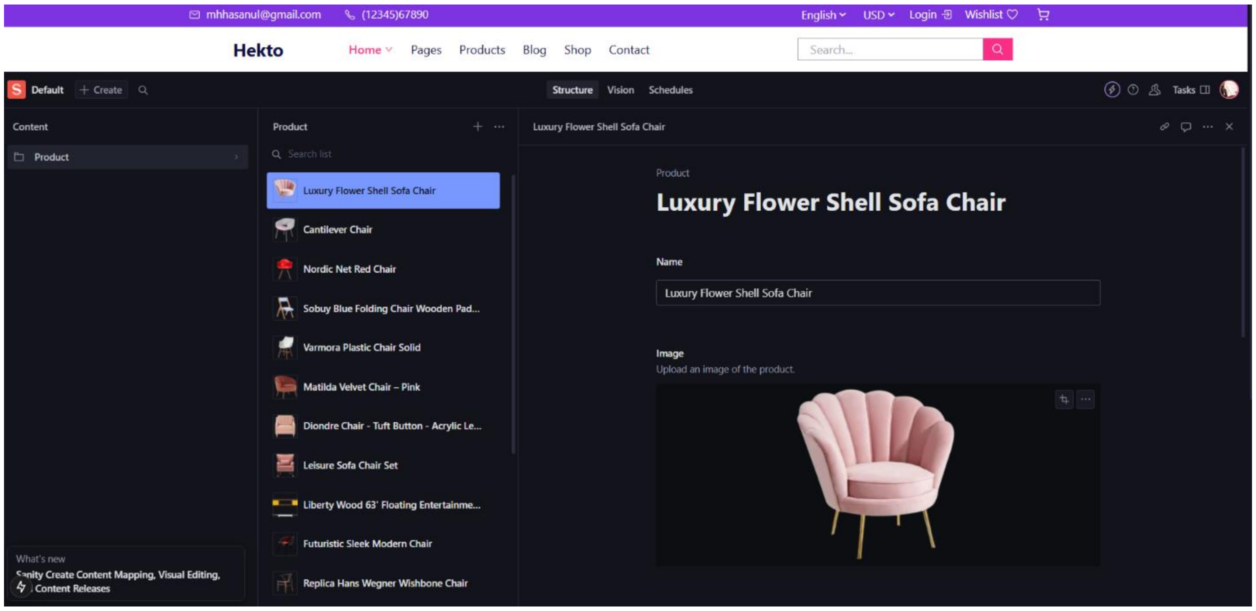
Product

Sobuy Blue Folding Chair Wooden Padded

Name: Sobuy Blue Folding Chair Wooden Padded

Image: Upload an image of the product.

What's new: Sanity Create Content Mapping, Visual Editing, and Content Releases



Product

- Luxury Flower Shell Sofa Chair
- Cantilever Chair
- Nordic Net Red Chair
- Sobuy Blue Folding Chair Wooden Pad...
- Varmora Plastic Chair Solid
- Matilda Velvet Chair – Pink
- Diondre Chair - Tuft Button - Acrylic Le...
- Leisure Sofa Chair Set
- Liberty Wood 63' Floating Entertainme...
- Futuristic Sleek Modern Chair
- Replica Hans Wegner Wishbone Chair

Luxury Flower Shell Sofa Chair

Product

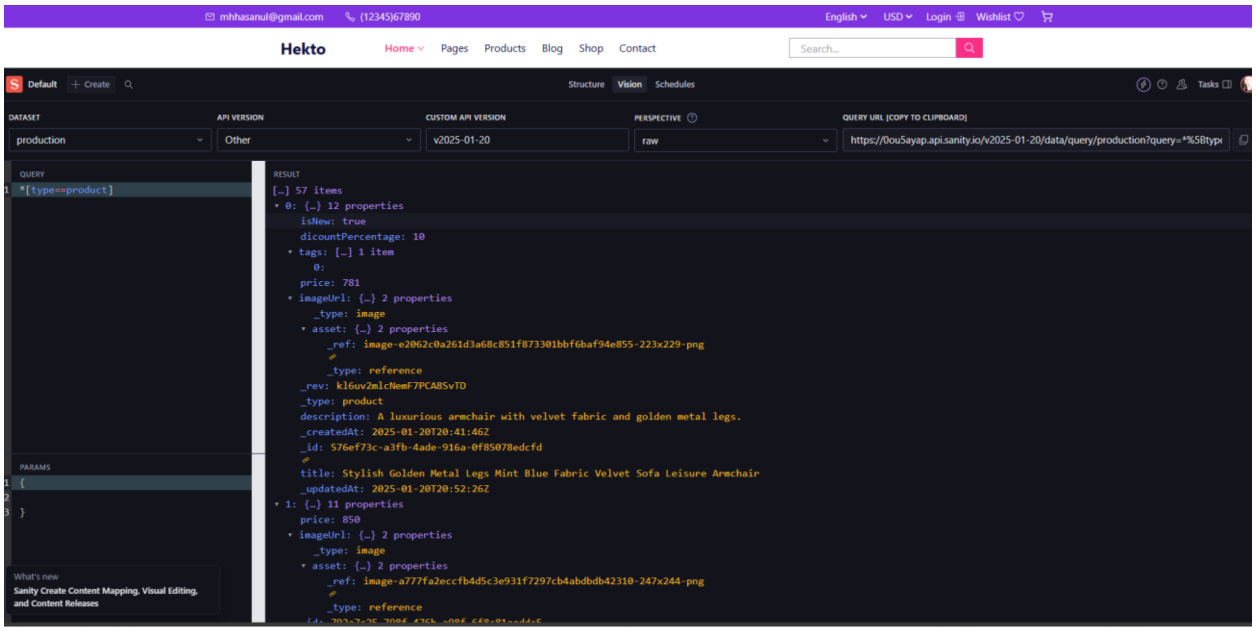
Luxury Flower Shell Sofa Chair

Name: Luxury Flower Shell Sofa Chair

Image: Upload an image of the product.

What's new: Sanity Create Content Mapping, Visual Editing, and Content Releases

GROQ Query Implementation:



DATASET: production
API VERSION: Other
CUSTOM API VERSION: v2025-01-20
PERSPECTIVE: raw
QUERY URL [COPY TO CLIPBOARD]: https://0ou5ayap.api.sanity.io/v2025-01-20/data/query/production?query=*%5Btype

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

PARAMS
```
{

}
```

RESULT
```
[…] 57 items
▾ 0: {…} 12 properties
    isNew: true
    dicountPercentage: 10
  ▾ tags: […] 1 item
      0:
    price: 781
  ▾ imageUrl: {…} 2 properties
      _type: image
    ▾ asset: {…} 2 properties
        _ref: image-e2062c0a261d3a68c851f873301bbf6baf94e855-223x229-png
        _type: reference
    _rev: kl6uv2mlcNemF7PCA8SvTD
    _type: product
    description: A luxurious armchair with velvet fabric and golden metal legs.
    _createdAt: 2025-01-20T20:41:46Z
    _id: 576ef73c-a3fb-4ade-916a-0f85078edcfd
    title: Stylish Golden Metal Legs Mint Blue Fabric Velvet Sofa Leisure Armchair
    _updatedAt: 2025-01-20T20:52:26Z
▾ 1: {…} 11 properties
    price: 850
  ▾ imageUrl: {…} 2 properties
      _type: image
    ▾ asset: {…} 2 properties
        _ref: image-a777fa2eccfb4d5c3e931f7297cb4abdbdb42310-247x244-png
        _type: reference
```

What's new
Sanity Create Content Mapping, Visual Editing, and Content Releases