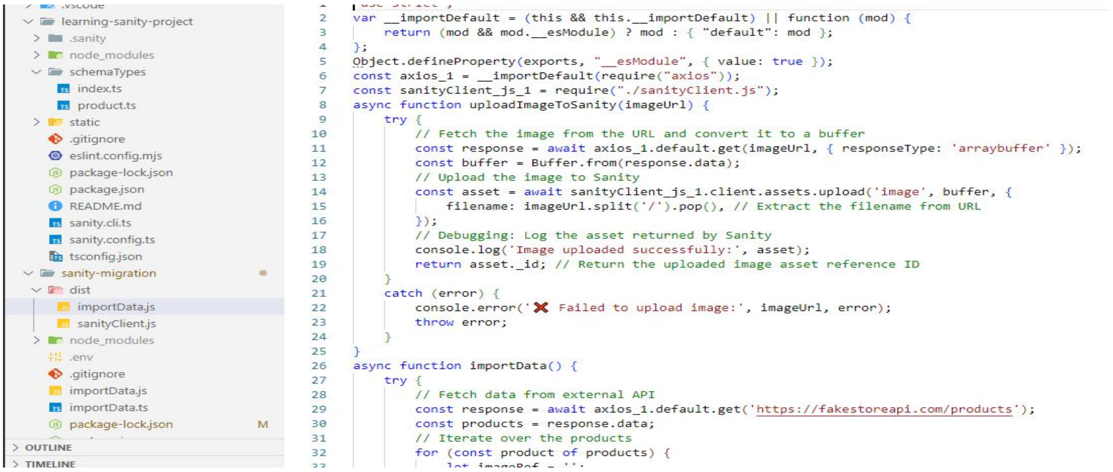
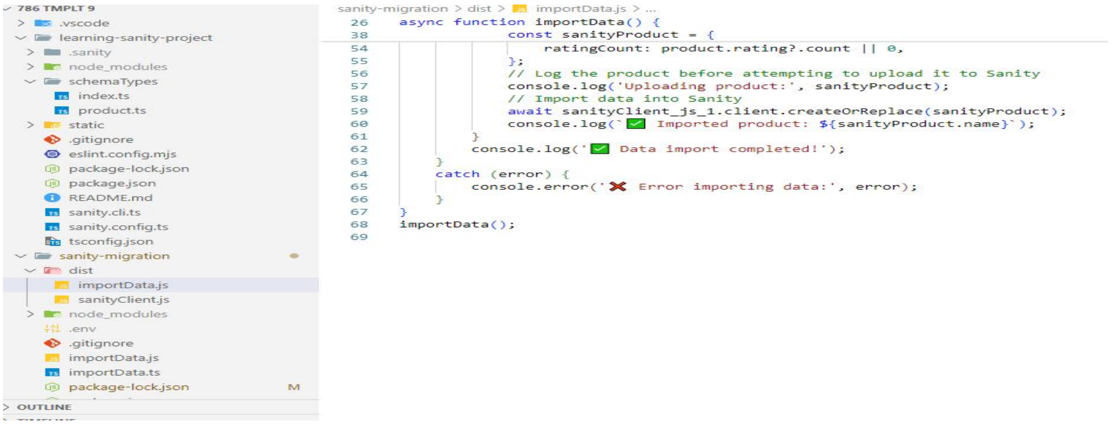
Day 3
API Integration and data Migaration

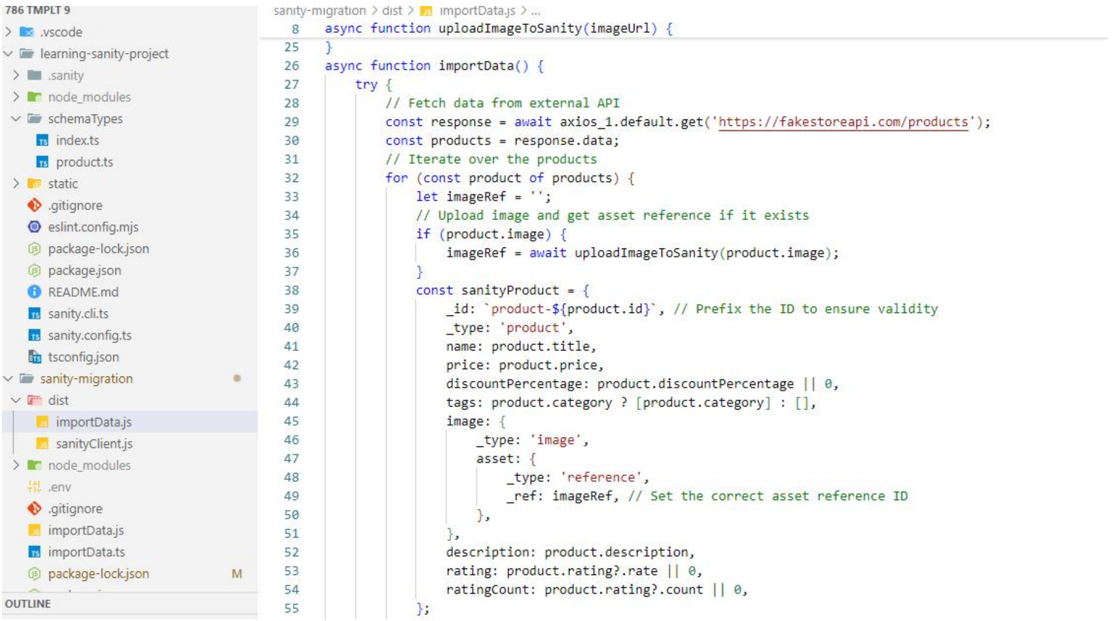
API



```
"use strict";
var __importDefault = (this && this.__importDefault) || function (mod) {
    return (mod && mod.__esModule) ? mod : { "default": mod };
};
Object.defineProperty(exports, "__esModule", { value: true });
const axios_1 = __importDefault(require("axios"));
const sanityClient_js_1 = require("./sanityClient.js");
async function uploadImageToSanity(imageUrl) {
    try {
        // Fetch the image from the URL and convert it to a buffer
        const response = await axios_1.default.get(imageUrl, { responseType: 'arraybuffer' });
        const buffer = Buffer.from(response.data);
        // Upload the image to Sanity
        const asset = await sanityClient_js_1.client.assets.upload('image', buffer, {
            filename: imageUrl.split('/').pop(), // Extract the filename from URL
        });
        // Debugging: Log the asset returned by Sanity
        console.log('Image uploaded successfully:', asset);
        return asset._id; // Return the uploaded image asset reference ID
    }
    catch (error) {
        console.error('❌ Failed to upload image:', imageUrl, error);
        throw error;
    }
}
async function importData() {
    try {
        // Fetch data from external API
        const response = await axios_1.default.get('https://fakestoreapi.com/products');
        const products = response.data;
        // Iterate over the products
        for (const product of products) {
            let imageRef = '';
```

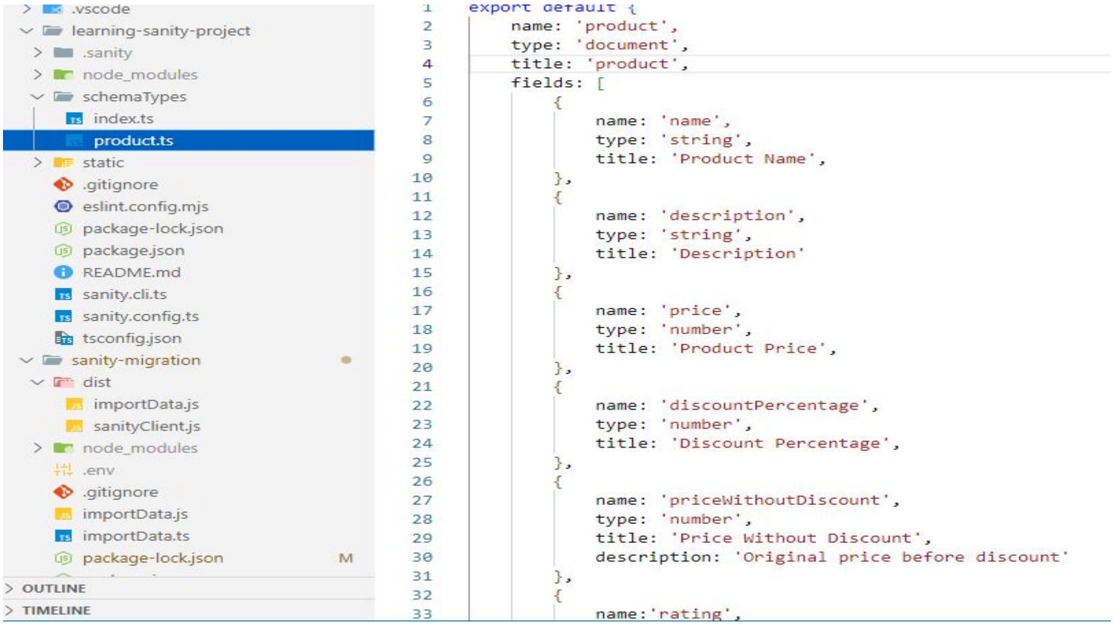


```
async function importData() {
                const sanityProduct = {
                    ratingCount: product.rating?.count || 0,
                };
                // Log the product before attempting to upload it to Sanity
                console.log('Uploading product:', sanityProduct);
                // Import data into Sanity
                await sanityClient_js_1.client.createOrReplace(sanityProduct);
                console.log(`✅ Imported product: ${sanityProduct.name}`);
            }
            console.log('✅ Data import completed!');
        }
        catch (error) {
            console.error('❌ Error importing data:', error);
        }
}
importData();
```

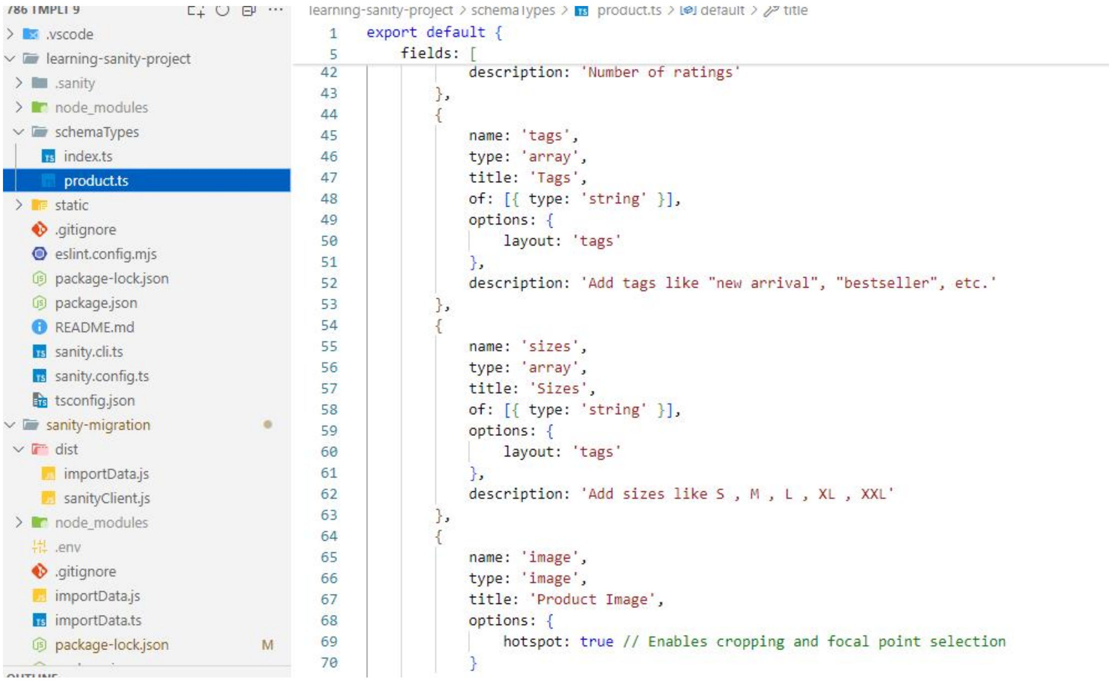


```javascript
async function uploadImageToSanity(imageUrl) {
}
async function importData() {
    try {
        // Fetch data from external API
        const response = await axios_1.default.get('https://fakestoreapi.com/products');
        const products = response.data;
        // Iterate over the products
        for (const product of products) {
            let imageRef = '';
            // Upload image and get asset reference if it exists
            if (product.image) {
                imageRef = await uploadImageToSanity(product.image);
            }
            const sanityProduct = {
                _id: `product-${product.id}`, // Prefix the ID to ensure validity
                _type: 'product',
                name: product.title,
                price: product.price,
                discountPercentage: product.discountPercentage || 0,
                tags: product.category ? [product.category] : [],
                image: {
                    _type: 'image',
                    asset: {
                        _type: 'reference',
                        _ref: imageRef, // Set the correct asset reference ID
                    },
                },
                description: product.description,
                rating: product.rating?.rate || 0,
                ratingCount: product.rating?.count || 0,
            };
```

Schema

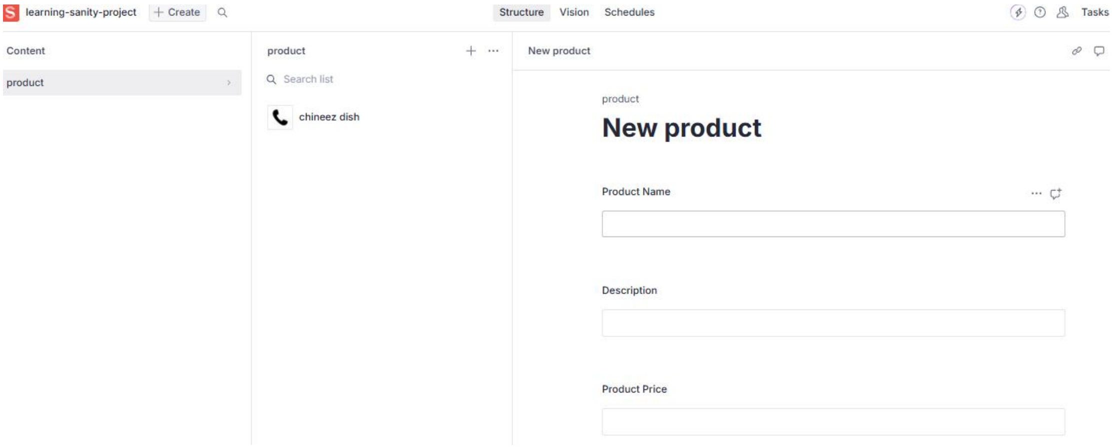


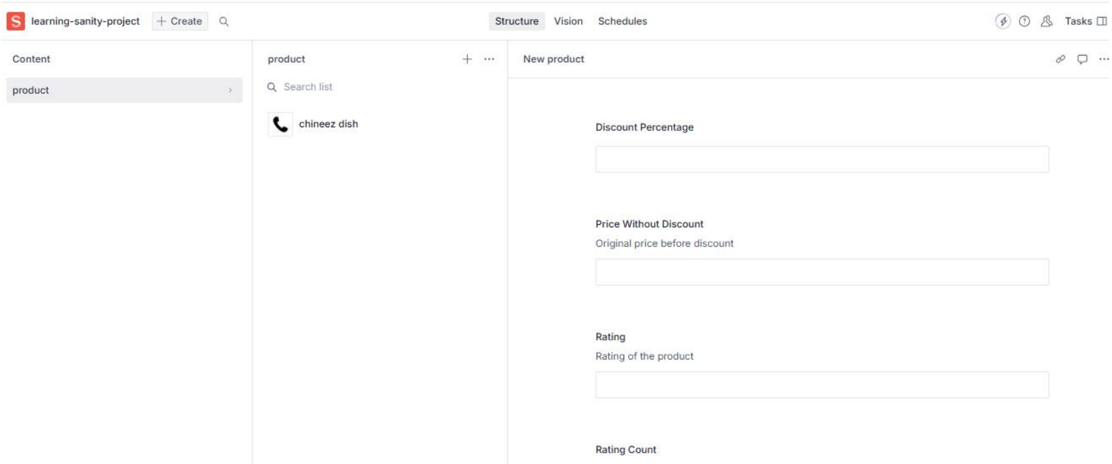
```javascript
export default {
    name: 'product',
    type: 'document',
    title: 'product',
    fields: [
        {
            name: 'name',
            type: 'string',
            title: 'Product Name',
        },
        {
            name: 'description',
            type: 'string',
            title: 'Description'
        },
        {
            name: 'price',
            type: 'number',
            title: 'Product Price',
        },
        {
            name: 'discountPercentage',
            type: 'number',
            title: 'Discount Percentage',
        },
        {
            name: 'priceWithoutDiscount',
            type: 'number',
            title: 'Price Without Discount',
            description: 'Original price before discount'
        },
        {
            name:'rating',
```



```
export default {
    fields: [
            description: 'Number of ratings'
        },
        {
            name: 'tags',
            type: 'array',
            title: 'Tags',
            of: [{ type: 'string' }],
            options: {
                layout: 'tags'
            },
            description: 'Add tags like "new arrival", "bestseller", etc.'
        },
        {
            name: 'sizes',
            type: 'array',
            title: 'Sizes',
            of: [{ type: 'string' }],
            options: {
                layout: 'tags'
            },
            description: 'Add sizes like S , M , L , XL , XXL'
        },
        {
            name: 'image',
            type: 'image',
            title: 'Product Image',
            options: {
                hotspot: true // Enables cropping and focal point selection
            }
```

DAta Miggration



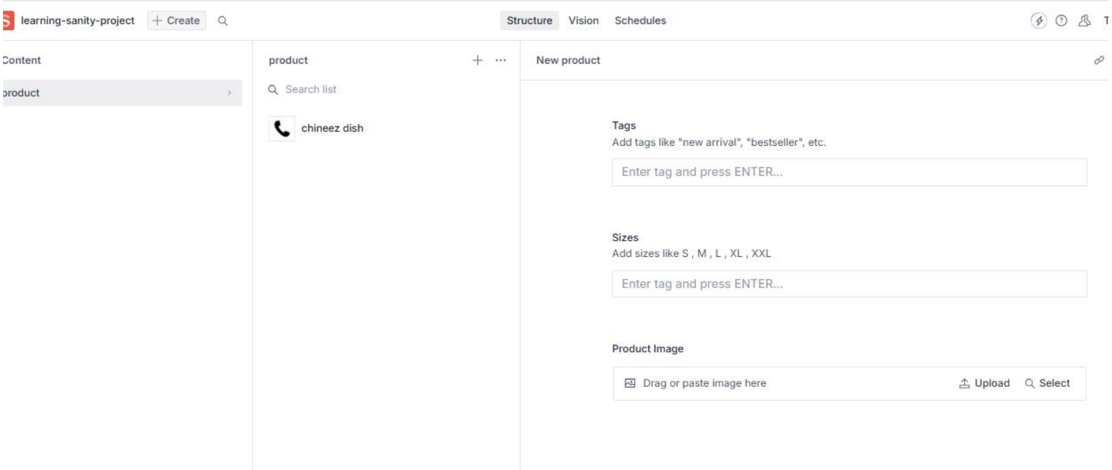


Discount Percentage

Price Without Discount
Original price before discount

Rating
Rating of the product

Rating Count

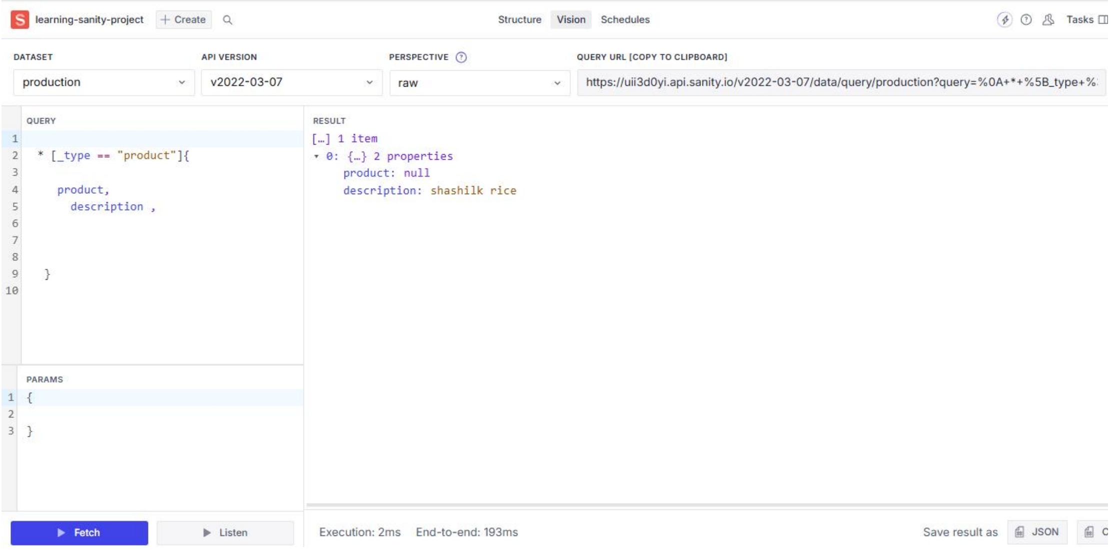


Tags
Add tags like "new arrival", "bestseller", etc.
Enter tag and press ENTER...

Sizes
Add sizes like S , M , L , XL , XXL
Enter tag and press ENTER...

Product Image
Drag or paste image here

Data fetch



DATASET: production

API VERSION: v2022-03-07

PERSPECTIVE: raw

QUERY URL [COPY TO CLIPBOARD]: https://uii3d0yi.api.sanity.io/v2022-03-07/data/query/production?query=%0A+*+%5B_type+%

QUERY

```

 * [_type == "product"]{

    product,
      description ,



  }

```

RESULT

```
[…] 1 item
▾ 0: {…} 2 properties
    product: null
    description: shashilk rice
```

PARAMS

```
{

}
```

Execution: 2ms End-to-end: 193ms

THE End