

# Day 3 - API Integration Report - [FURNITURE STORE]

## API Integration Process

### Selected API:

- **API Name:** Sanity CMS API
- **Purpose:** To manage and store product data for the furniture marketplace
- **\*\*Authentication Method:** API Token stored in **\*\*\***

### Integration Steps:

#### Sanity Installation

- Installed Sanity CMS using `npm install -g @sanity/cli`.

## Running Data Import

- Executed `node scripts/importSanityData.mjs` to migrate products.

## Migration Steps and Tools Used

### Database Backup

- Created a backup of the existing database before applying changes.

### Schema Updates

- Updated Sanity schema to match API response.

### Data Migration

- Wrote and executed the `importSanityData.mjs` script to transform existing data.

### Testing and Validation

- Installed Sanity CMS using `npm install -g @sanity/cli`.
- Initialized Sanity Studio using `sanity init`.

### Setting Up Environment

- Created `.env.local` and added `SANITY_PROJECT_ID` and `SANITY_API_TOKEN`.

### Schema Setup

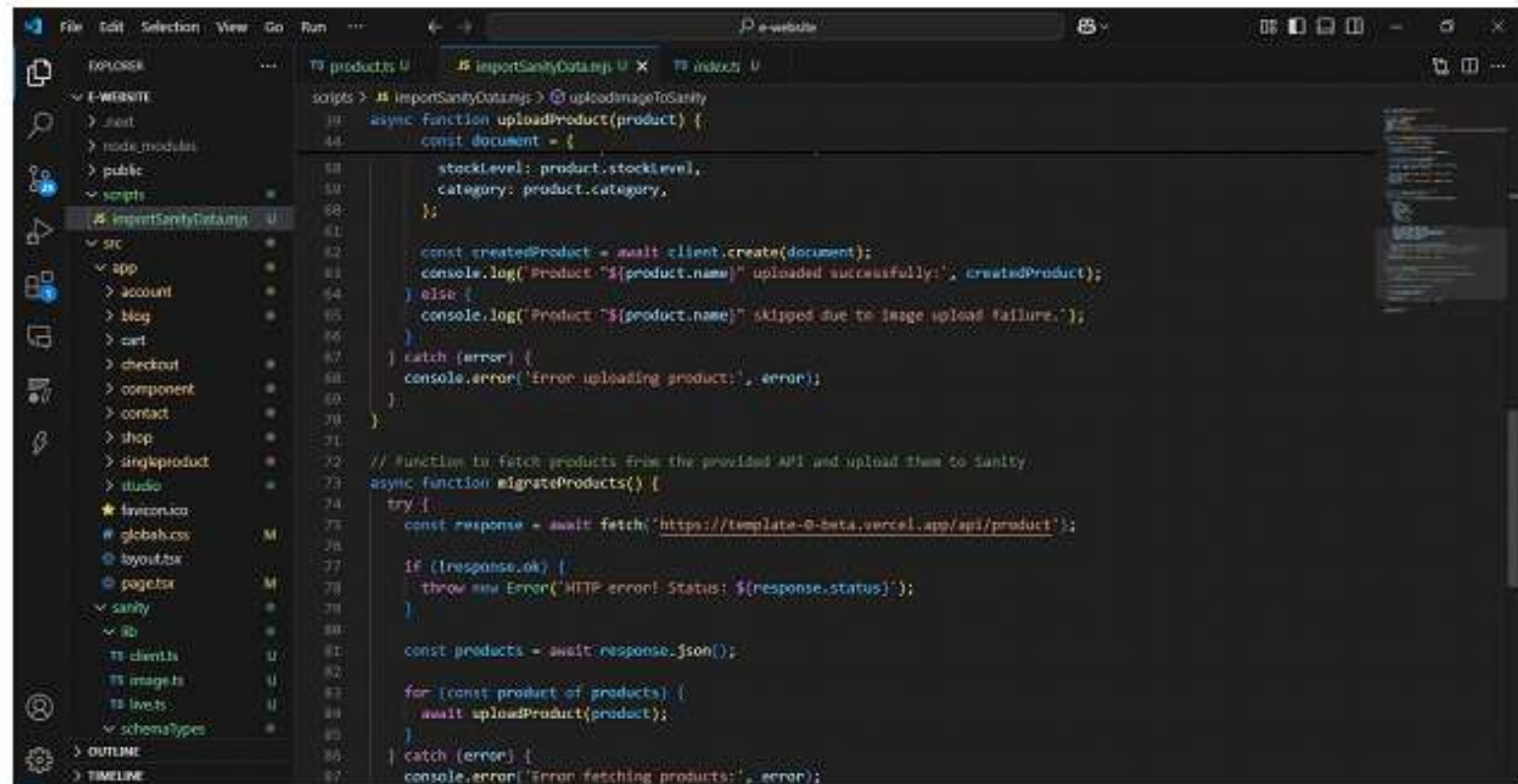
- Created `product.ts` schema file and defined product structure.
- Updated `schemaTypes` index file to include the new product schema.

### Data Import Script

- Created `scripts/importSanityData.mjs` for importing data.
- Used `fetch` to retrieve products and upload them to Sanity.

### Running Data Import

## API Calls



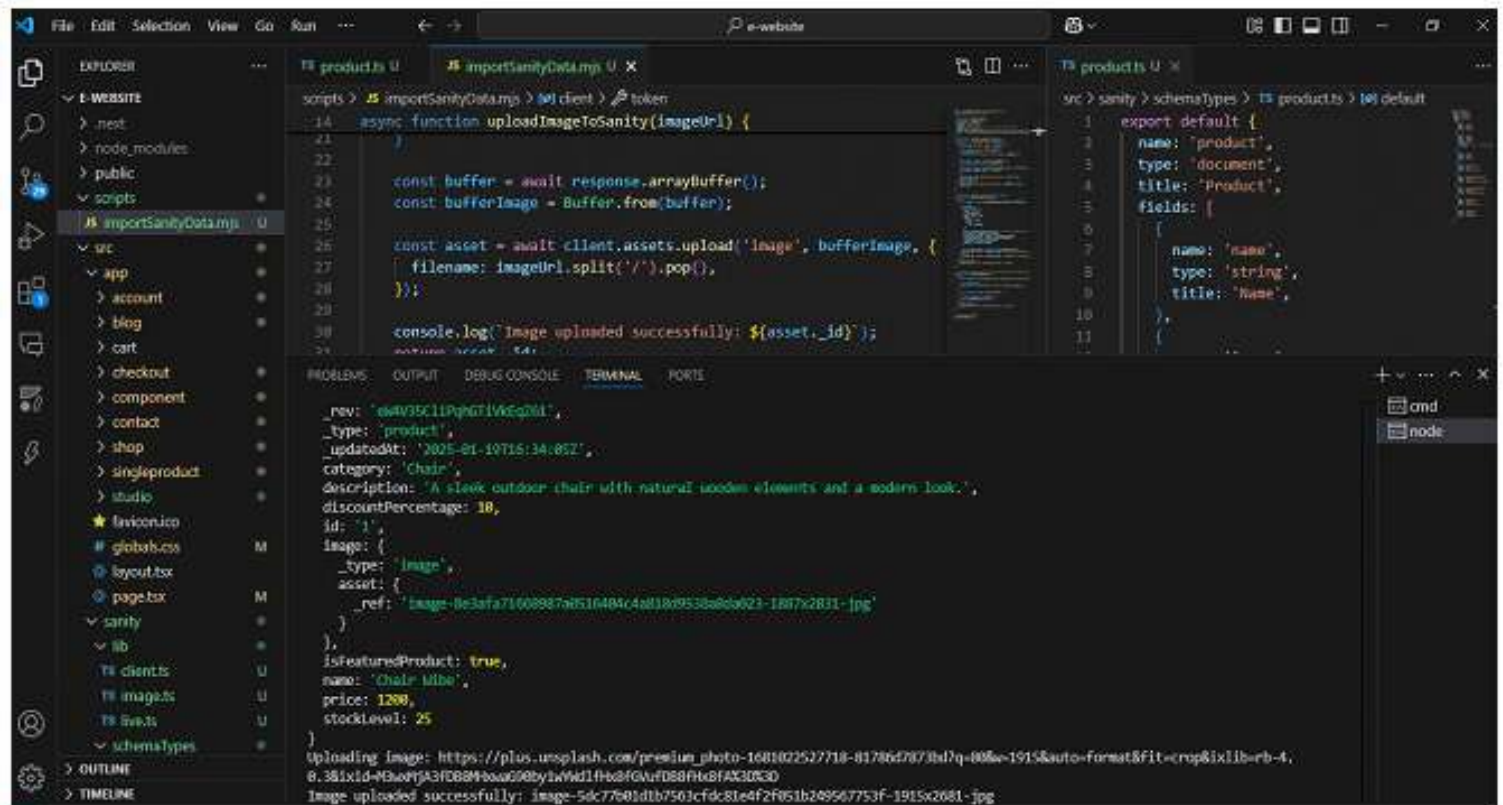
The screenshot shows the VS Code editor with the file explorer on the left, the Explorer view in the middle, and the editor window on the right. The Explorer view shows the project structure with the file `scripts/importSanityData.js` selected. The editor window displays the following code:

```
10 async function uploadProduct(product) {
11   const document = {
12     stockLevel: product.stockLevel,
13     category: product.category,
14   };
15
16   const createdProduct = await client.create(document);
17   console.log('Product "${product.name}" uploaded successfully:', createdProduct);
18 } else {
19   console.log('Product "${product.name}" skipped due to image upload failure.');
```



The screenshot shows the VS Code editor with the file explorer on the left, the Explorer view in the middle, and the editor window on the right. The Explorer view shows the project structure with the file `scripts/importSanityData.js` selected. The editor window displays the following code:

```
14 async function uploadImageToSanity(imageUrl) {
15   // ...
16   const buffer = await response.arrayBuffer();
17   const bufferImage = Buffer.from(buffer);
18
19   const asset = await client.assets.upload('image', bufferImage, {
20     filename: imageUrl.split('/').pop(),
21   });
22 }
```



## Data Successfully Displayed in Frontend





# Data Successfully Displayed in Frontend

