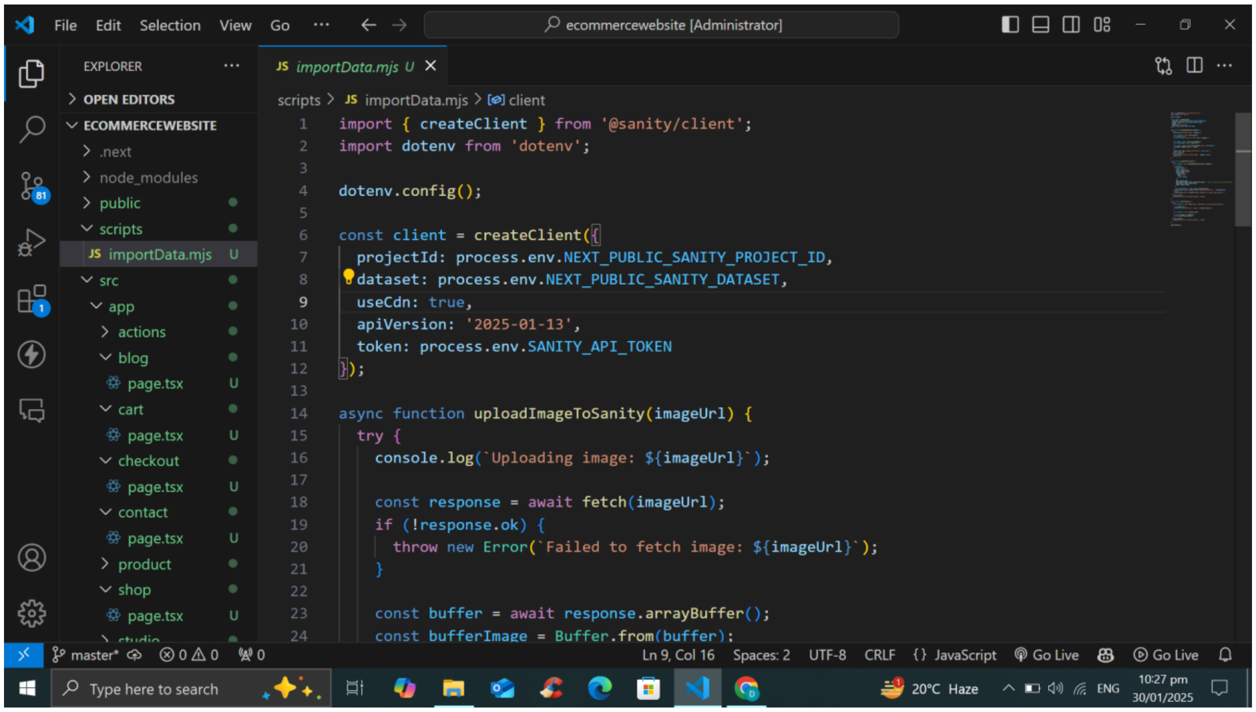
DAY 3 - API INTEGRATION AND DATA MIGRATION

Data Migration:

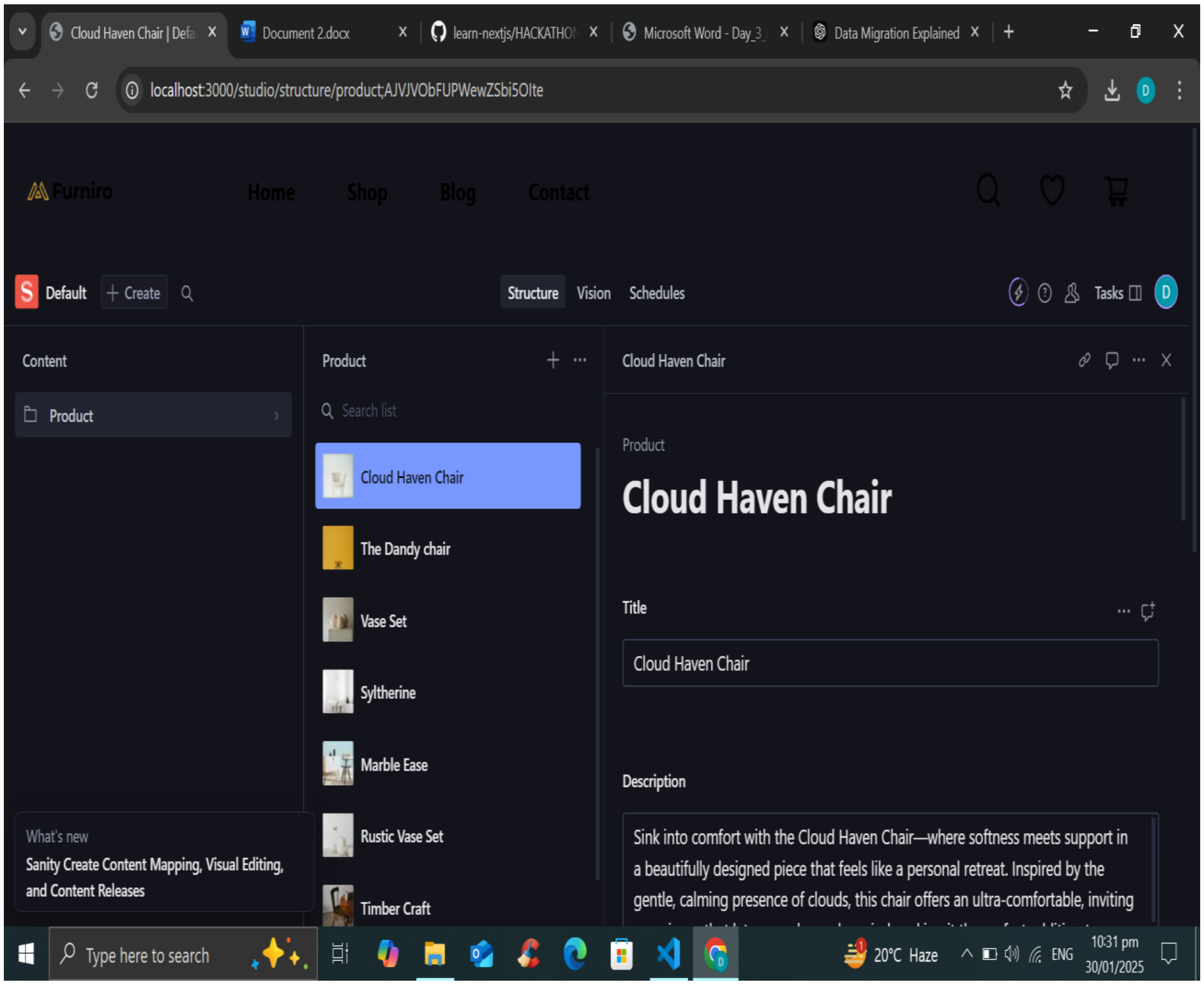
Migrating data into Sanity CMS to build a functional marketplace backend.. Sanity allows developers to structure, manage, and query content in a flexible and API-driven way, often used in modern web and app development. Migrating data into Sanity means you're moving or importing existing content from another system or database into Sanity's content structure.



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

dotenv.config();

const client = createClient({
  projectId: process.env.NEXT_PUBLIC_SANITY_PROJECT_ID,
  dataset: process.env.NEXT_PUBLIC_SANITY_DATASET,
  useCdn: true,
  apiVersion: '2025-01-13',
  token: process.env.SANITY_API_TOKEN
});

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);
```

Sanity Studio:

Here is an Sanity Studio Where I migrated my data

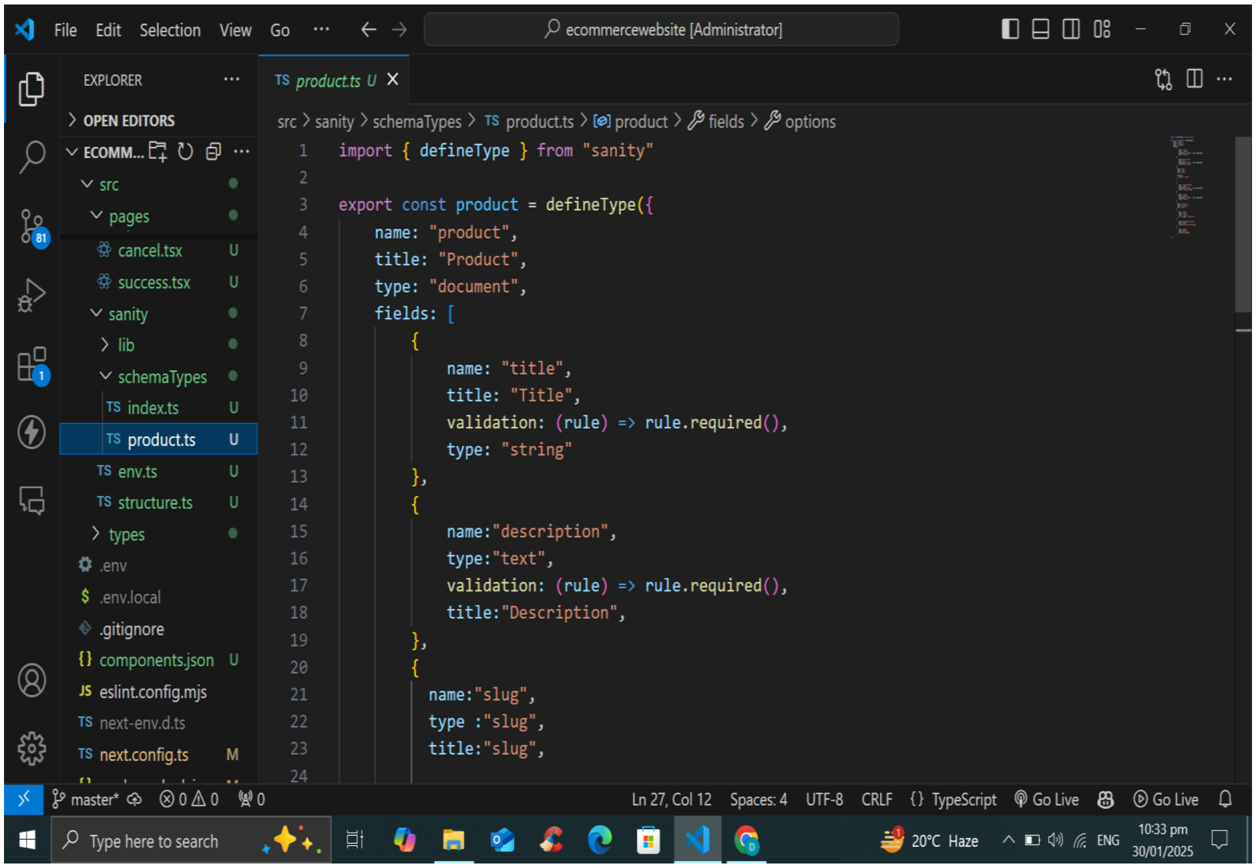


Title: Cloud Haven Chair

Description: Sink into comfort with the Cloud Haven Chair—where softness meets support in a beautifully designed piece that feels like a personal retreat. Inspired by the gentle, calming presence of clouds, this chair offers an ultra-comfortable, inviting

Here is my Schema Of My Product:

In **Sanity**, a **schema** defines the structure of the content you want to manage and how it will be stored. It's essentially a blueprint that specifies the types of content (like blog posts, products, users, etc.) and their respective fields (like titles, images, dates, etc.). The schema determines how content is created, organized, and queried in Sanity.



```
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:"slug",
          type :"slug",
          title:"slug",
```