Day-3 Hackathon

Marketeplace Bandage E-Commerce

API INTEGRATION & DATA MIGRATION

API INTEGRATION:

API integration is the process of connecting different software applications to communicate and work together. It involves using Application Programming Interfaces (APIs) to allow one system to access the functions and data of another system. This enables seamless data sharing and functionality between applications, enhancing efficiency and automation. For example, integrating a payment gateway API into your website allows customers to make secure payments directly on your site.

DATA MIGRATION:

Data migration is the process of moving data from one location to another, which can involve transferring it from an old system to a new one. This can include databases, applications, or storage systems. The goal is to ensure that the data remains intact and usable in the new location. It involves planning, transferring, testing, and validating the data to make sure everything works properly.

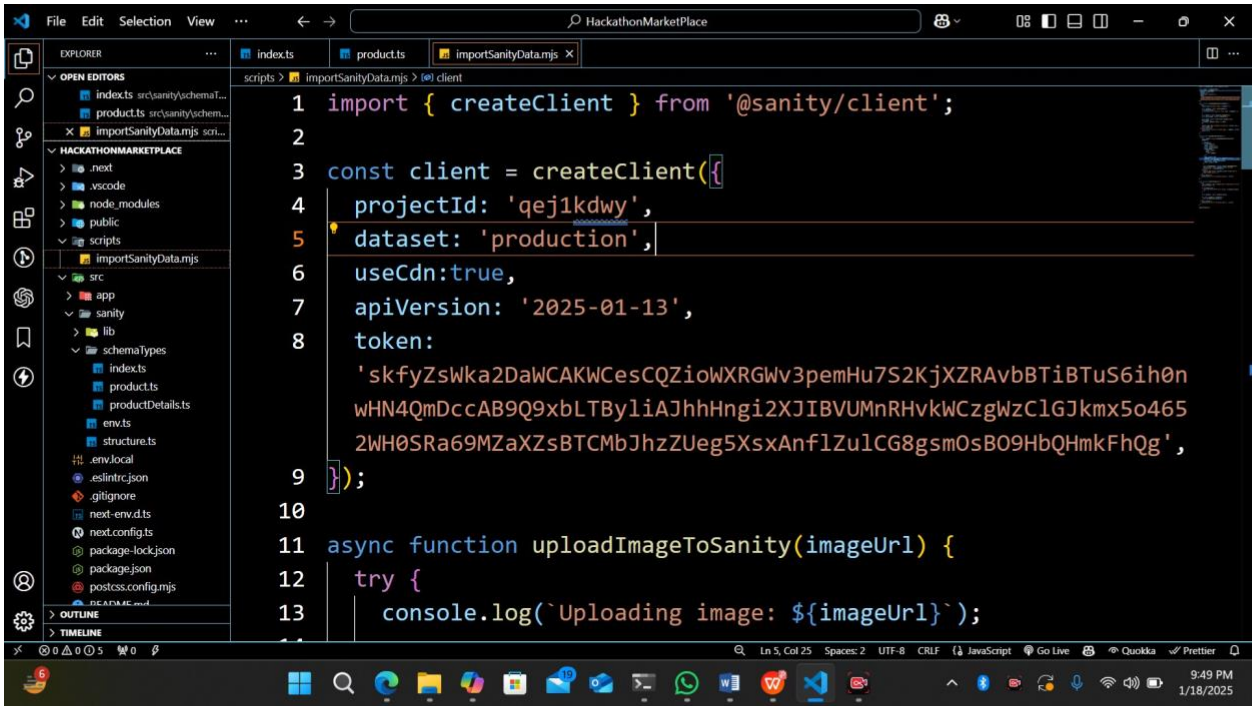
I practice data migration because it's very useful for my business goals. It helps keep data up-to-date, improves system performance, and supports the adoption of new technologies, making operations more efficient and reliable.

API INTEGRATION & DATA MIGRATION:

Data migration is the process of moving data from one location to another, which can involve transferring it from an old system to a new one. This can include databases, applications, or storage systems. The goal is to ensure that the data remains intact and usable in the new location. It involves planning, transferring, testing, and validating the data to make sure everything works properly.

I practice data migration because it's very useful for my business goals. It helps keep data up-to-date, improves system performance, and supports the adoption of new technologies, making operations more efficient and reliable.

I practice API integration because it's incredibly useful for achieving my business goals. It helps streamline operations, reduce manual work, and improve customer experience by enabling real-time data updates and smooth interactions across various platforms and services. I put my sanity project Id, and Token in **(importSanityData.mjs)** file.

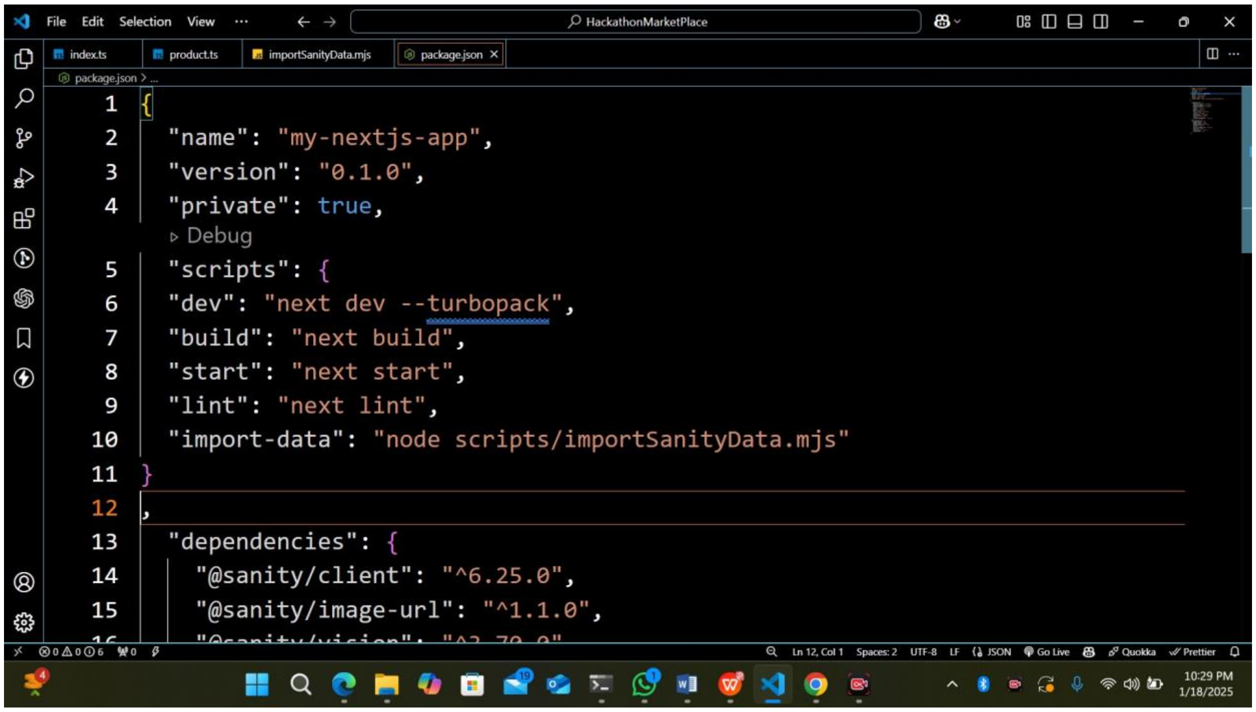


```
import { createClient } from '@sanity/client';

const client = createClient({
  projectId: 'qej1kdwy',
  dataset: 'production',
  useCdn:true,
  apiVersion: '2025-01-13',
  token:
  'skfyZsWka2DaWCAKWCesCQZioWXRGWv3pemHu7S2KjXZRAvbBTiBTuS6ih0n
  wHN4QmDccAB9Q9xbLTByliAJhhHngi2XJIBVUMnRHvkWCzgWzClGJkmx5o465
  2WH0SRa69MZaXZsBTCMbJhzZUeg5XsxAnflZulCG8gsmOsBO9HbQHmkFhQg',
});

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

Running the Import Script:



```
{
  "name": "my-nextjs-app",
  "version": "0.1.0",
  "private": true,
  ▷ Debug
  "scripts": {
  "dev": "next dev --turbopack",
  "build": "next build",
  "start": "next start",
  "lint": "next lint",
  "import-data": "node scripts/importSanityData.mjs"
  }
,
  "dependencies": {
    "@sanity/client": "^6.25.0",
    "@sanity/image-url": "^1.1.0",
```

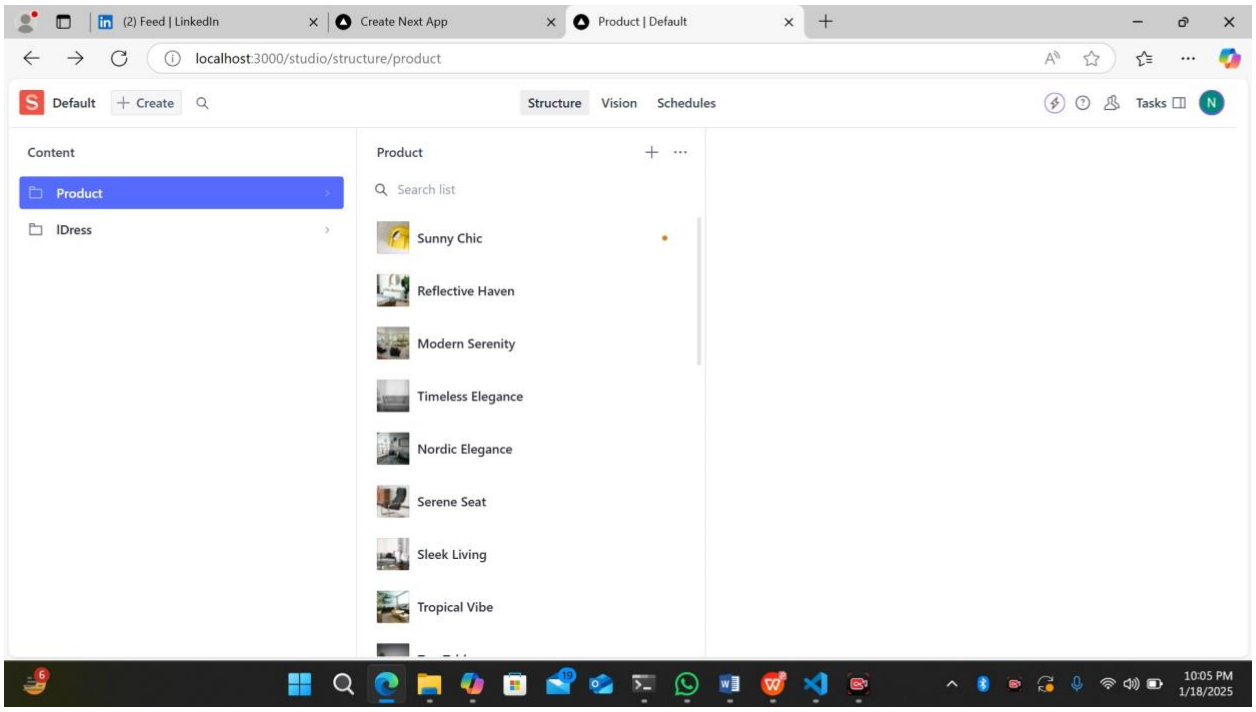
Then run the following commands import script using:

```
npm run scripts/importSanityData.mjs
```

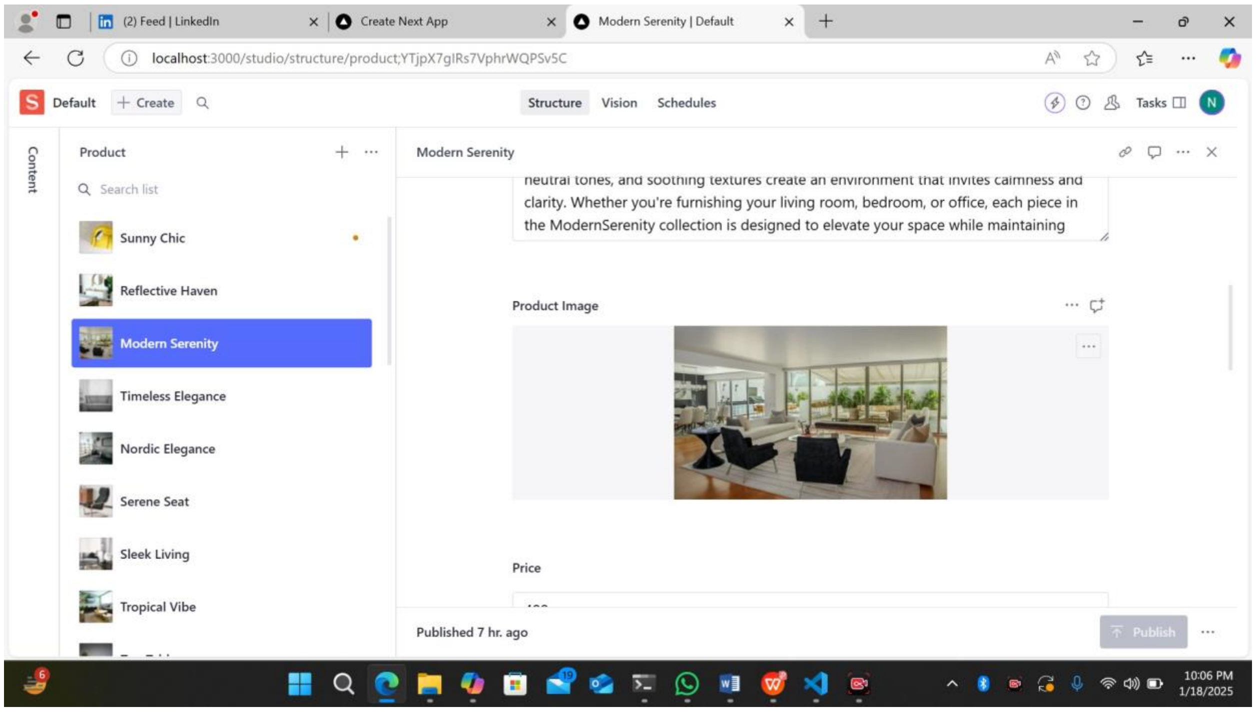
This script fetch products from the FakeStoreAPI, upload any associated images to Sanity's asset store, and then create new product documents in your Sanity dataset.

Sanity Studio Screen Shoot:

Sanity Migration in Sanity Studio:



Sanity Structure For Products Details:



neutral tones, and soothing textures create an environment that invites calmness and clarity. Whether you're furnishing your living room, bedroom, or office, each piece in the ModernSerenity collection is designed to elevate your space while maintaining

Conclusion:

By this task , I've learned how to configure environment variables for Sanity integration, retrieve my Sanity Project ID and API token, create a custom product schema, and import data from an external API into my Sanity content .

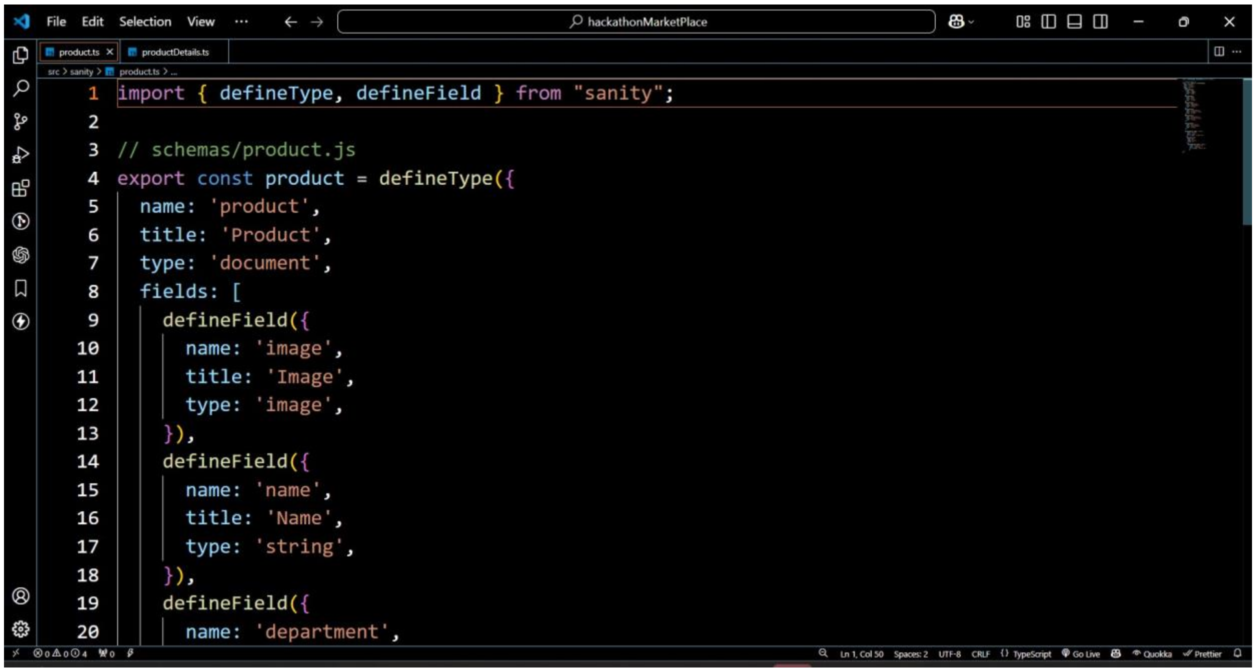
Created My Own Sanity Data in website:

First, I define the structure of my data with the Sanity schema. This lets me customize fields like text, numbers, images, and references. Once My schema is set up, I input data into Sanity's content studio.

In my Next.js project, I use the Sanity client to fetch and display this data. I created dynamic pages that show the updated content from Sanity, ensuring my website always has the latest information. This approach makes managing and updating content a breeze, saving time and effort.

Practicing this is very useful for my business goals because it allows for efficient content management and a smooth user experience.

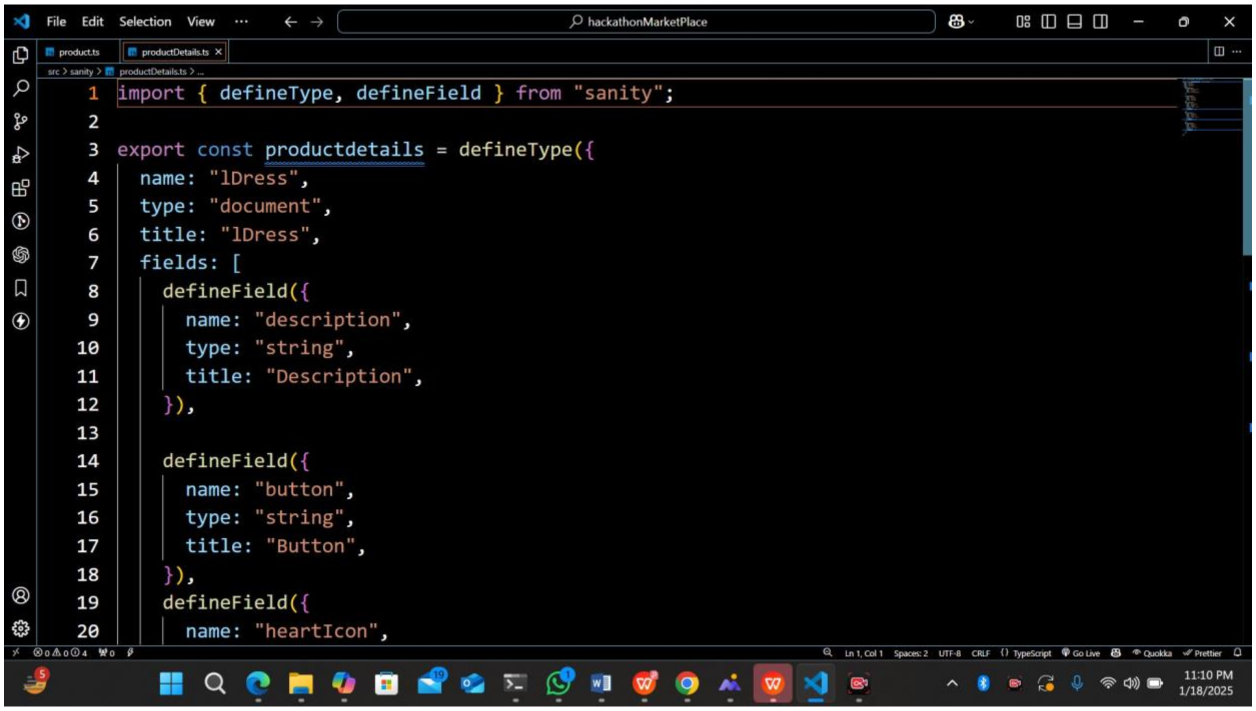
Product.ts (SCHEMA) :



```
import { defineType, defineField } from "sanity";

// schemas/product.js
export const product = defineType({
  name: 'product',
  title: 'Product',
  type: 'document',
  fields: [
    defineField({
      name: 'image',
      title: 'Image',
      type: 'image',
    }),
    defineField({
      name: 'name',
      title: 'Name',
      type: 'string',
    }),
    defineField({
      name: 'department',
```

ProductDetails.ts (SCHEMA) :

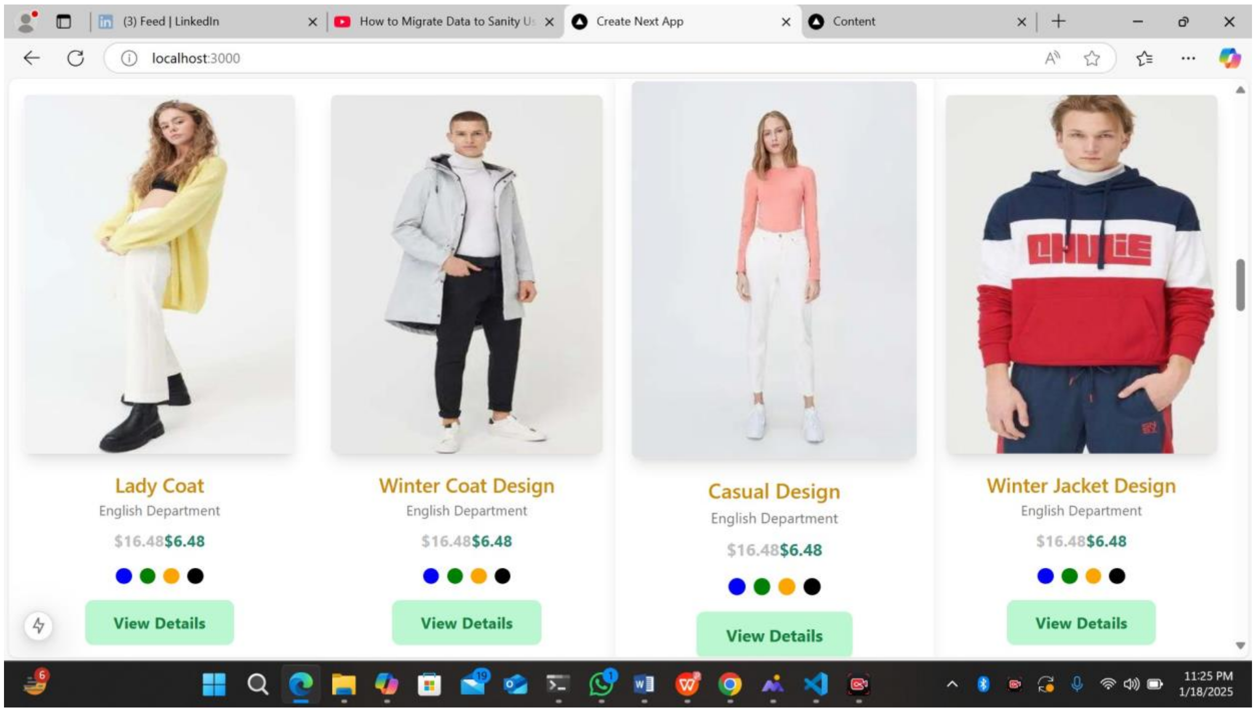


```
import { defineType, defineField } from "sanity";

export const productdetails = defineType({
  name: "lDress",
  type: "document",
  title: "lDress",
  fields: [
    defineField({
      name: "description",
      type: "string",
      title: "Description",
    }),

    defineField({
      name: "button",
      type: "string",
      title: "Button",
    }),
    defineField({
      name: "heartIcon",
```

UI Products Cards Using Sanity Schema



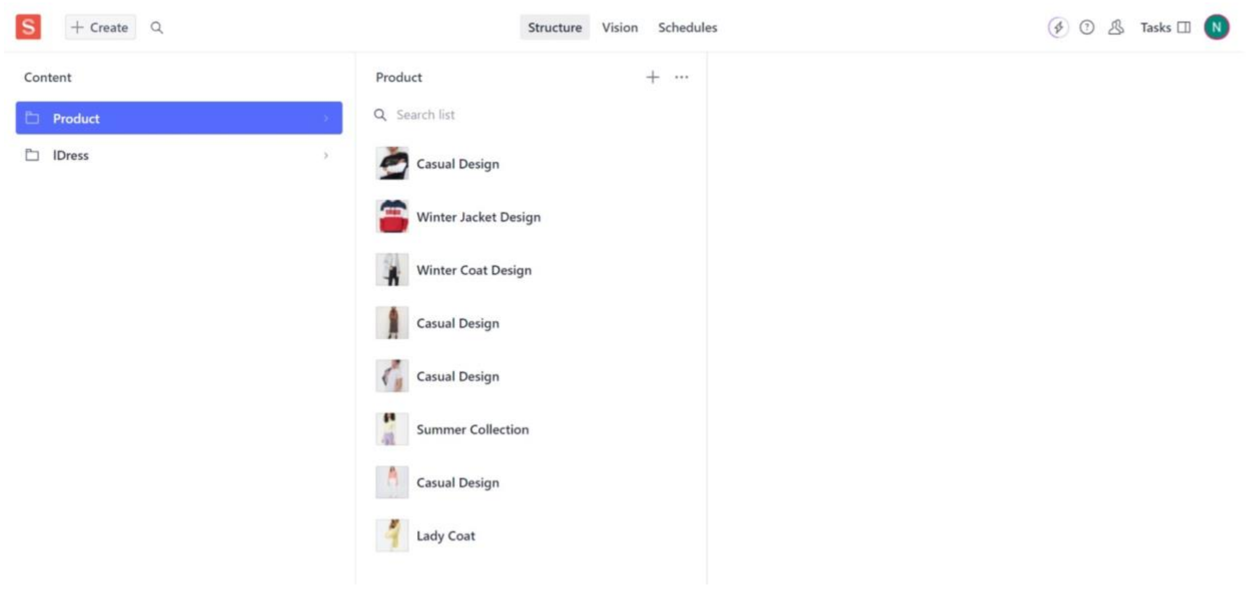
Lady Coat
English Department
$16.48$6.48
● ● ● ●
View Details

Winter Coat Design
English Department
$16.48$6.48
● ● ● ●
View Details

Casual Design
English Department
$16.48$6.48
● ● ● ●
View Details

Winter Jacket Design
English Department
$16.48$6.48
● ● ● ●
View Details

Sanity Studio Display



Content

Product

IDress

Product

Search list

Casual Design

Winter Jacket Design

Winter Coat Design

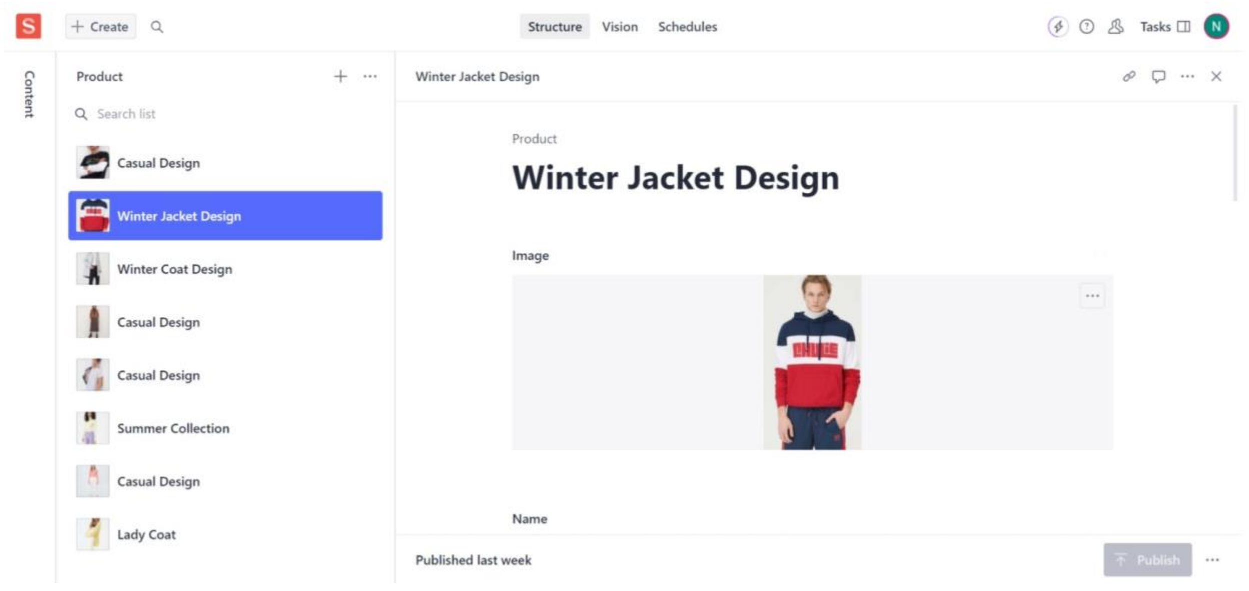
Casual Design

Casual Design

Summer Collection

Casual Design

Lady Coat



Product

Search list

Casual Design

Winter Jacket Design

Winter Coat Design

Casual Design

Casual Design

Summer Collection

Casual Design

Lady Coat

Winter Jacket Design

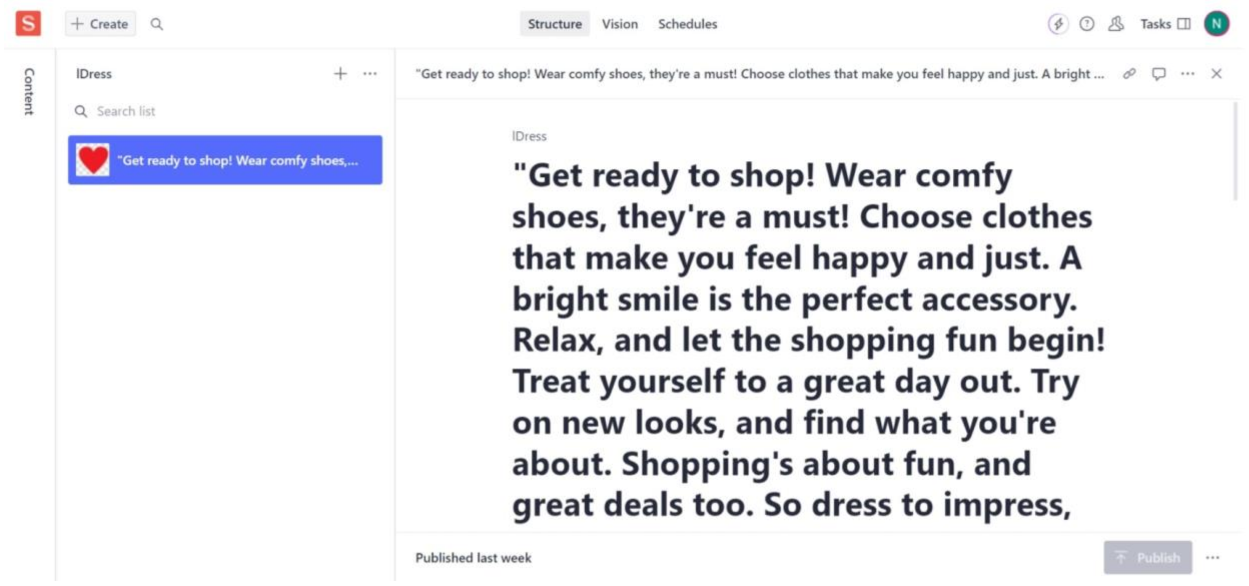
Product

Winter Jacket Design

Image

Name

Published last week


S
Create
Structure
Vision
Schedules
Tasks
N
Content
IDress
Search list
"Get ready to shop! Wear comfy shoes,...
"Get ready to shop! Wear comfy shoes, they're a must! Choose clothes that make you feel happy and just. A bright ...
Published last week
Publish

IDress

"Get ready to shop! Wear comfy shoes, they're a must! Choose clothes that make you feel happy and just. A bright smile is the perfect accessory. Relax, and let the shopping fun begin! Treat yourself to a great day out. Try on new looks, and find what you're about. Shopping's about fun, and great deals too. So dress to impress,

GITHUB LINK:

https://github.com/Nazia-naz90/milestone-3-ecommerce-website.git

Conclusion:

By creating my own Sanity schemas and updating data in Next.js, I learned the importance of customizing data structures to fit specific needs. This experience showed me how to efficiently manage and display dynamic content. I also gained valuable skills in integrating back-end services with front-end frameworks, enhancing both my technical abilities and my project's functionality. This has been a significant step toward achieving my business goals by ensuring up-to-date and seamless content management.