

---

**Prepared By ASAD ALI**

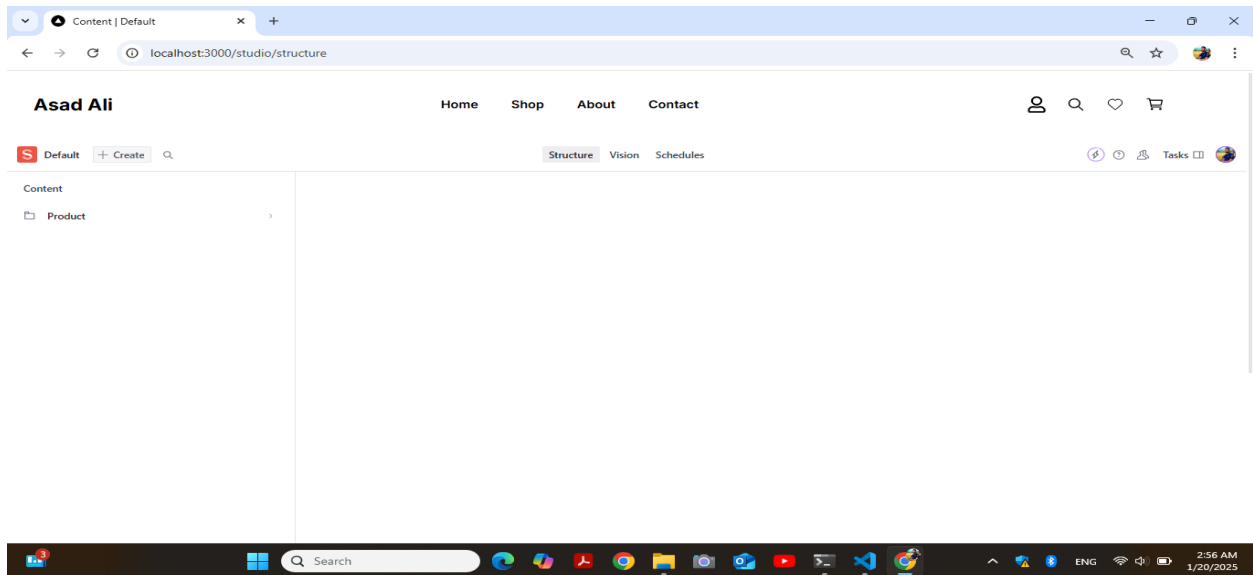
**Student of GIAIC**

---

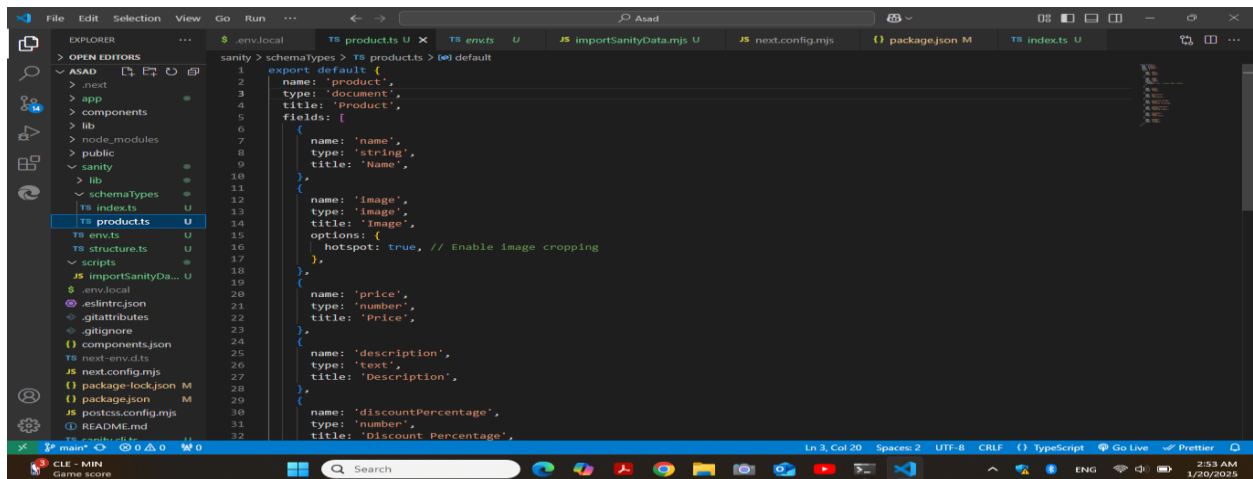
**Day 3**

**Marketplace Type: General E-Commerce Furniture Website**

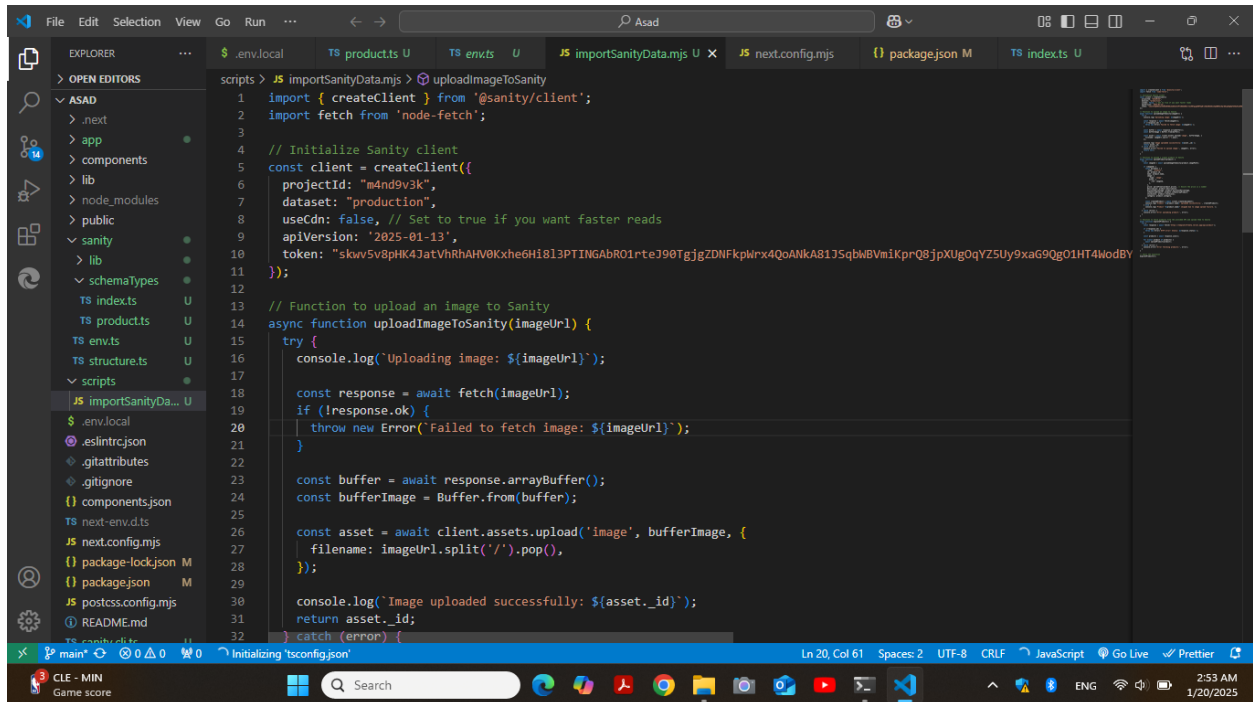
## 1. Created Sanity Project



## 2. Define Schema

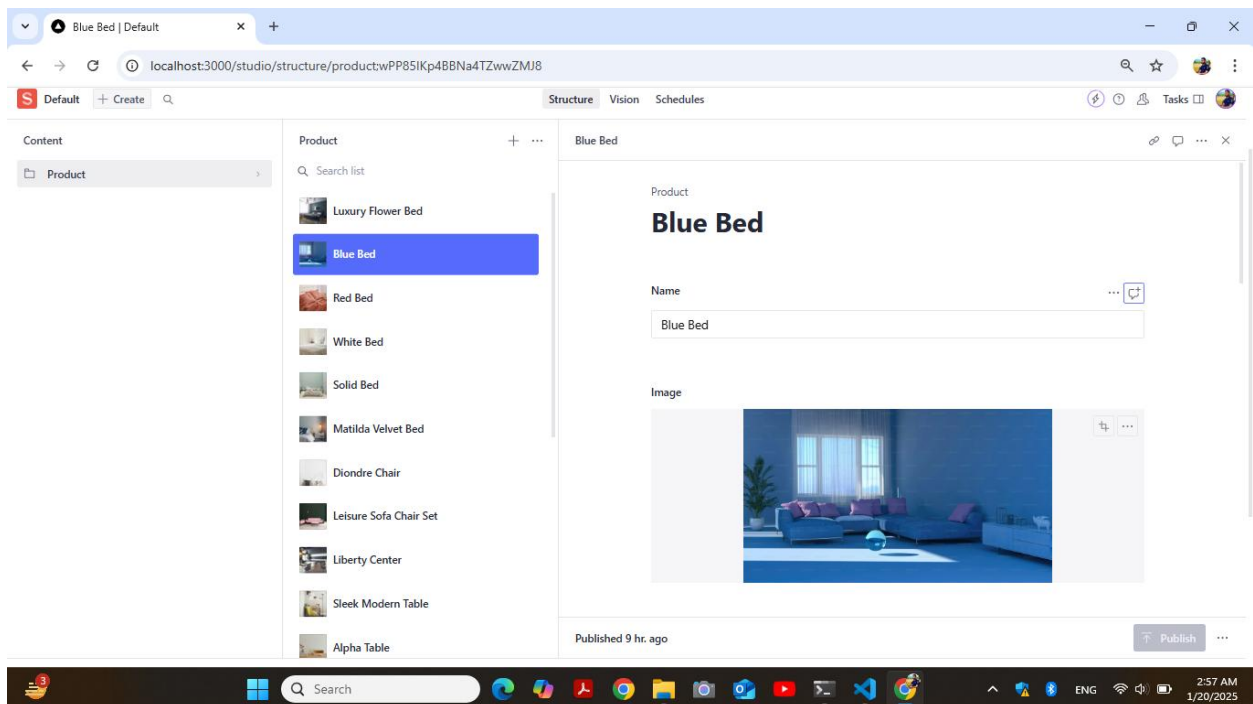


### 3.API



```
1 import { createClient } from '@sanity/client';
2 import fetch from 'node-fetch';
3
4 // Initialize Sanity client
5 const client = createClient({
6   projectId: "m4nd9v3k",
7   dataset: "production",
8   useCdn: false, // Set to true if you want faster reads
9   apiVersion: '2025-01-13',
10   token: "skav5v8pHK4JatVhRhAHV0Kxhe6Hi813PTINGAbR01rteJ90TgJgZDNFkpWrx4QoANKA81J5qbWbVmiKprQ8jpxUgOqYZ5Uy9xaG9G01HT4WodBY",
11 });
12
13 // Function to upload an image to Sanity
14 async function uploadImageToSanity(imageUrl) {
15   try {
16     console.log(`Uploading image: ${imageUrl}`);
17
18     const response = await fetch(imageUrl);
19     if (!response.ok) {
20       throw new Error(`Failed to fetch image: ${imageUrl}`);
21     }
22
23     const buffer = await response.arrayBuffer();
24     const bufferImage = Buffer.from(buffer);
25
26     const asset = await client.assets.upload('image', bufferImage, {
27       filename: imageUrl.split('/').pop(),
28     });
29
30     console.log(`Image uploaded successfully: ${asset._id}`);
31     return asset._id;
32   } catch (error) {
33     console.error(error);
34   }
35 }
```

### 4.Data import in Sanity



The screenshot shows the Databricks Studio interface. At the top, there's a navigation bar with 'Structure', 'Vision', and 'Schedules' tabs. Below this is a toolbar with icons for search, star, and user profile. The main workspace is divided into several sections:

- Dataset:** 'production' (dropdown)
- API Version:** 'Other' (dropdown)
- Custom API Version:** 'v2025-01-17' (dropdown)
- Perspective:** 'raw' (dropdown)
- Query URL:** 'https://m4nd9v3kapi.sanity.io/v2025-01-17/data/query/product' (text input)

The central area displays a SQL query and its results:

```

1  QUERY
2  *[_type=="product"]{
3    name,
4    image,
5    price,
6    description,
7    discountPercentage,
8    isFeaturedProduct,
9    stockLevel,
10   category
11 }

12  PARAMS
13  {
14  }

15  RESULT
16  [-] 20 items
17  - 0: {...} 8 properties
18    isFeaturedProduct: true
19    stockLevel: 25
20    category: Chair
21    name: Chair Wibe
22  - image: {...} 2 properties
23    _type: image
24    - asset: {...} 1 property
25      _ref: image-8e3afa71668987a0516404c4a818d9538a8da023-1887x2831-jpg
26    price: 1200
27    description: A sleek outdoor chair with natural wooden elements and a modern look.
28    discountPercentage: 10
29  - 1: {...} 8 properties
30    description: An elegant armchair with plush cushions and a curved design for comfort.
31    discountPercentage: 0
32    isFeaturedProduct: false
33    stockLevel: 10
34    category: Chair
  
```

At the bottom, there are buttons for 'Fetch' and 'Listen', and a status bar showing 'Execution: 4ms' and 'End-to-end: 1344ms'. On the right, there are buttons for 'Save result as', 'JSON', and 'CSV'.

The screenshot shows a VS Code editor with the following details:

- Explorer Panel:** Shows the project structure with folders like 'ASAD', '.next', 'app', 'components', 'lib', 'node\_modules', 'public', 'sanity', 'scripts', and files like 'favicon.ico', 'globals.css', 'layout.tsx', 'page.tsx', 'components.json', 'next-env.d.ts', '.env.local', 'eslint.config.json', 'gitattributes', 'gitignore', 'package.json', 'tsconfig.json', and 'tsconfig.json'.
- Editor Panel:** Displays the content of 'page.tsx' in the 'app' folder. The code includes:
 

```

import { client } from '@sanity/lib/client';
import { urlFor } from '@sanity/lib/image';

interface IPProduct {
  name: string; // Product name
  price: number; // Product price
  description: string; // Product description
  discountPercentage: number; // Discount percentage
  isFeaturedProduct: boolean; // Whether it's a featured product
  stockLevel: number; // Stock level
  category: string; // Category
  imageUrl: string; // Image URL for the product
}

export default async function Shop() {
  // Fetch products using GROQ query
  const query = `
    *[_type == "product"]{
      name,
      price,
      description,
      discountPercentage,
      isFeaturedProduct,
      stockLevel,
      category,
      "imageUrl": image.asset->url
    }
  `;

  const products: IPProduct[] = await client.fetch(query);

  return (
    <div className="bg-gray-50 min-h-screen">

```
- Status Bar:** Shows 'Ln & Col 45', 'Spaces 2', 'UTF-8', 'CRLF', 'TypeScript JSX', 'Go Live', and 'Prettier'.

## 7. Product Display

