# Hackathon

## DAY 3 - API INTEGRATION AND DATA MIGRATION

1. **Downloading a Sanity in our project:**
   - To install the sanity use the below-mentioned command
     npm i --global sanity@latest

   - To create a sanity project use the below-mentioned command
     npm create sanity@latest -- --template clean --create-project "Furniro" --dataset production
   - Now add the token in the ".env.local" file.

2. **Setting up Environment Variable:**


3. **Creating the Sanity Schema:**

   In folder sanity/schemaTypes create a file named product.ts

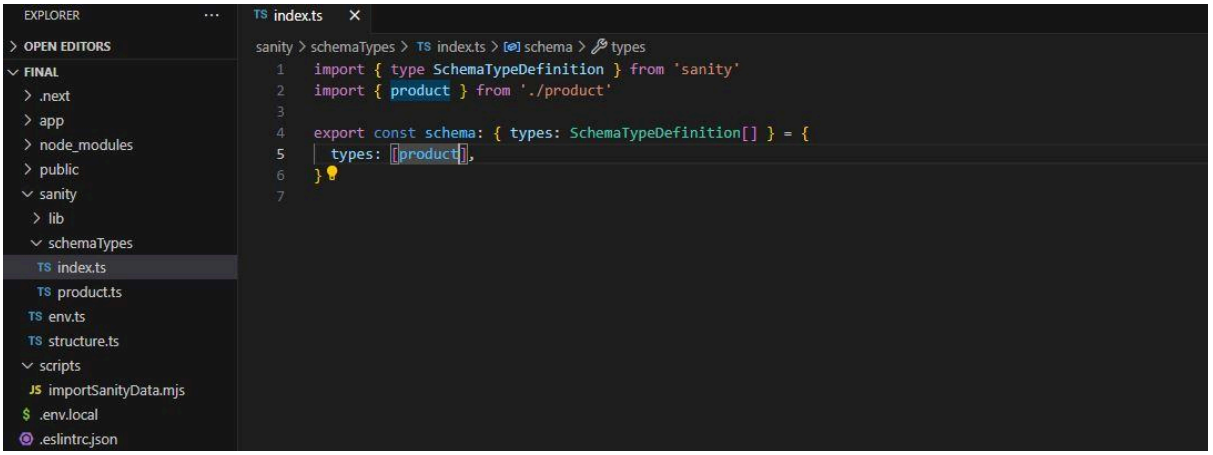
```
import { defineType } from "sanity"



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: "productImage",
            type: "image",
            validation: (rule) => rule.required(),
            title: "Product Image"
        },
        {
            name: "price",
            type: "number",
            validation: (rule) => rule.required(),
            title: "Price",
        },
        {
            name: "tags",
            type: "array",
            title: "Tags",
```

```
            of: [{ type: "string" }]

        },

        {

            name:"discountPercentage",

            type:"number",

            title:"Discount Percentage",

        },

        {

            name:"isNew",

            type:"boolean",

            title:"New Badge",

        }

    ]

})
```

4. **<u>Adding product component:</u>**

   Adding product components in index.ts




5. **<u>Setting up the Data Import Script:</u>**

Create a folder named scripts and file inside this folder named importSanityData.mjs

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


const client = createClient({
  // add you project id & apitoken
  projectId: NEXT_PUBLIC_SANITY_PROJECT_ID,
  dataset: "production",
  useCdn: true,
  apiVersion: "2025-01-13",
  token: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);
```

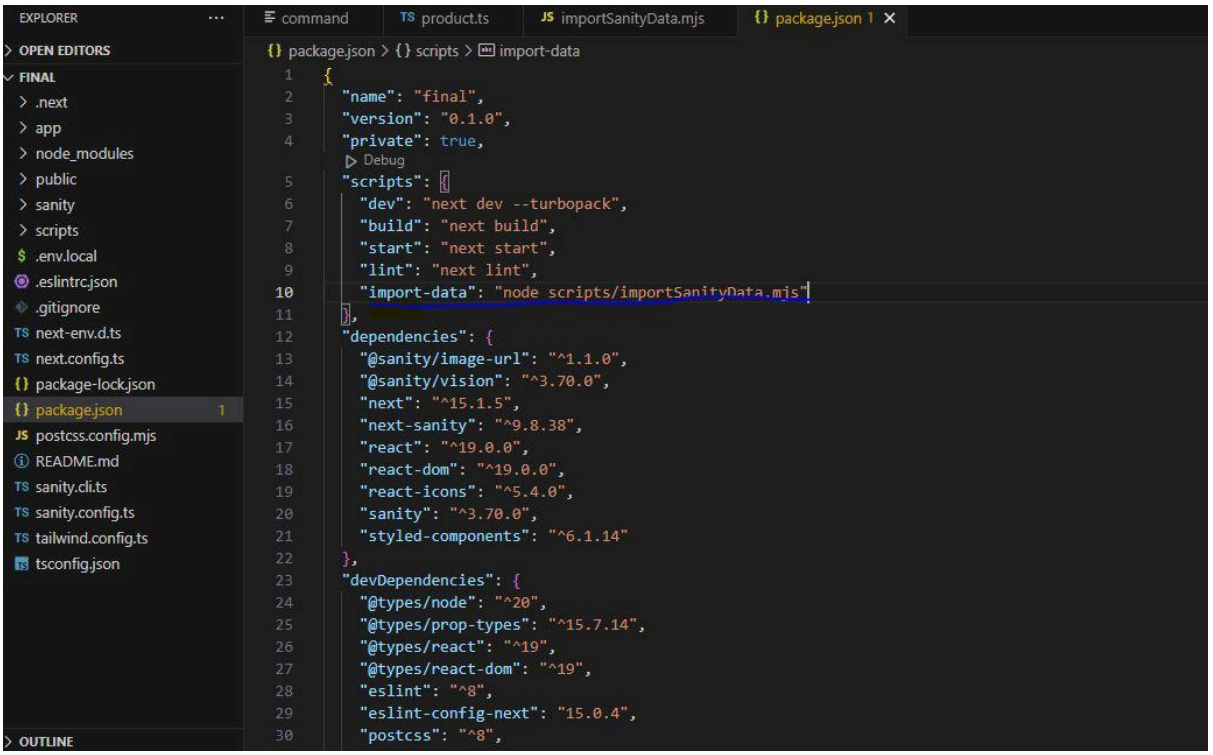
```
    const asset = await client.assets.upload("image", bufferImage, {
      filename: imageUrl.split("/").pop(),
    });

    console.log(`Image uploaded successfully: ${asset._id}`);
    return asset._id;
  } catch (error) {
    console.error("Failed to upload image:", imageUrl, error);
    return null;
  }
}

async function uploadProduct(product) {
  try {
    const imageId = await uploadImageToSanity(product.imageUrl);

    if (imageId) {
      const document = {
        _type: "product",
        title: product.title,
        price: product.price,
        productImage: {
          _type: "image",
          asset: {
```

```
            _ref: imageId,
          },
        },
        tags: product.tags,
        dicountPercentage: product.dicountPercentage, // Typo in field
name: dicountPercentage -> discountPercentage
        description: product.description,
        isNew: product.isNew,
      };


      const createdProduct = await client.create(document);
      console.log(
        `Product ${product.title} uploaded successfully:`,
        createdProduct
      );
    } else {
      console.log(
        `Product ${product.title} skipped due to image upload failure.`
      );
    }
  } catch (error) {
    console.error("Error uploading product:", error);
  }
}
```

```
async function importProducts() {
  try {
    const response = await fetch(
      "https://template6-six.vercel.app/api/products"
    );

    if (!response.ok) {
      throw new Error(`HTTP error! Status: ${response.status}`);
    }

    const products = await response.json();

    for (const product of products) {
      await uploadProduct(product);
    }
  } catch (error) {
    console.error("Error fetching products:", error);
  }
}

importProducts();
```

4. **<u>Adding import-data in script:</u>**

we need to add a new script to our `package.json` file. Open your `package.json`and add the following to the `"scripts"` section:




6. **<u>Running command for adding data in sanity:</u>**

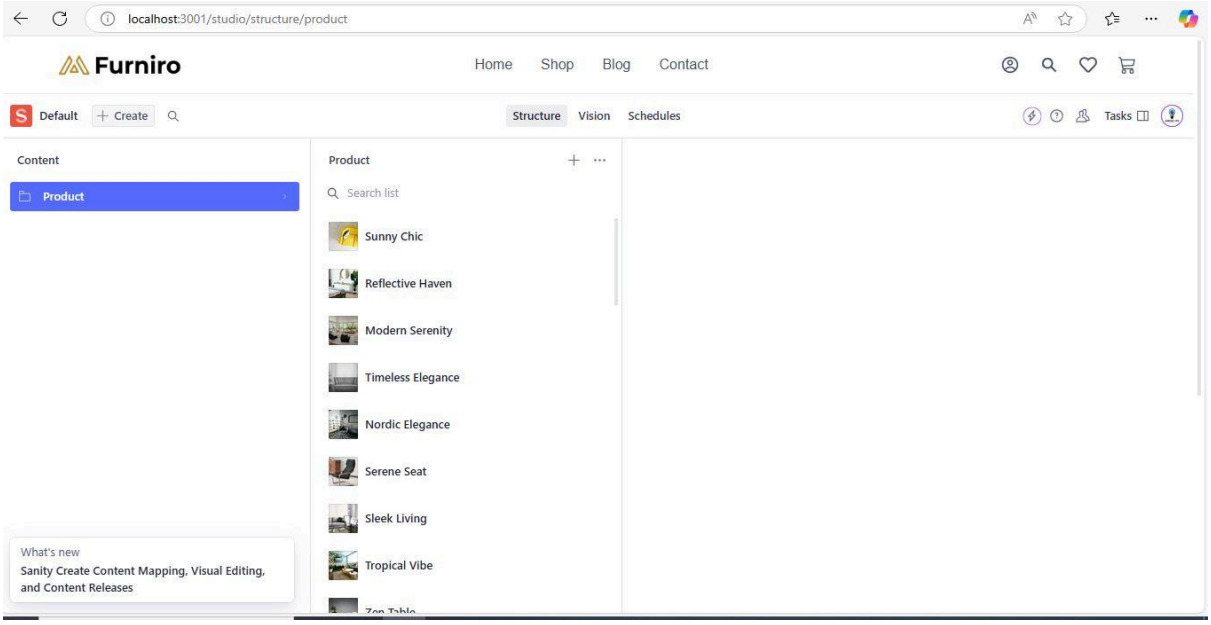
   Run npm run import-data command in your  terminal

7. **<u>Run command for checking data:</u>**

   Run npm dev in your terminal

8. **<u>Open google:</u>**

   Run [http://localhost:3000/studio](http://localhost:3000/studio) in google




9. **<u>For Checking product count:</u>**

**Query**

Using query for checking how many products we have in sanity.