Day 3: “API Integration”

This documentation is a guide for the setup and implementation of a Sanity schema, a migration script for transferring data from API created on Mock.api to another Sanity account, and overall project setup.

Sanity Installation:

installing Sanity in my Project Beadedwear,

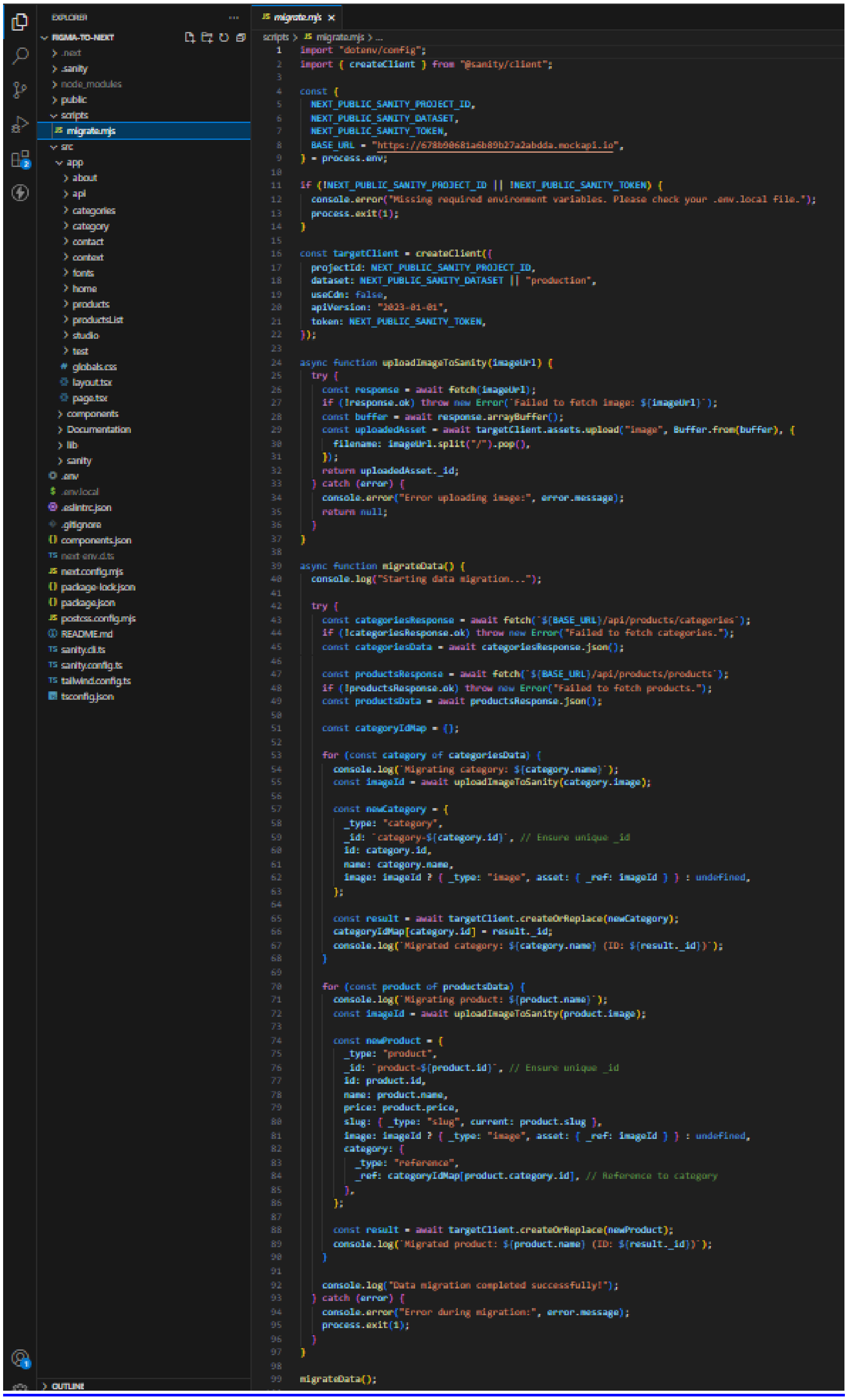
Sanity Schema:

1. After the installation Navigate to your schema folder:
    - If you have a src folder, go to /src/sanity/schemaTypes.
2. Create two new files and add the following code:
    - products.ts
        1. /src/sanity/schemaTypes/products.ts
    - categories.ts
        1. /src/sanity/schemaTypes/categories.ts
3. **Importing Schemas in index.ts**
    - src/sanity/schemaTypes/index.ts

Data Migration Script

The migration script is available in the project repository:

Migration Script



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

const {
  NEXT_PUBLIC_SANITY_PROJECT_ID,
  NEXT_PUBLIC_SANITY_DATASET,
  NEXT_PUBLIC_SANITY_TOKEN,
  BASE_URL = "https://678b08681a6b89b27a2abdda.mockapi.io",
} = process.env;

if (!NEXT_PUBLIC_SANITY_PROJECT_ID || !NEXT_PUBLIC_SANITY_TOKEN) {
  console.error("Missing required environment variables. Please check your .env.local file.");
  process.exit(1);
}

const targetClient = createClient({
  projectId: NEXT_PUBLIC_SANITY_PROJECT_ID,
  dataset: NEXT_PUBLIC_SANITY_DATASET || "production",
  useCdn: false,
  apiVersion: "2023-01-01",
  token: NEXT_PUBLIC_SANITY_TOKEN,
});

async function uploadImageToSanity(imageUrl) {
  try {
    const response = await fetch(imageUrl);
    if (!response.ok) throw new Error(`Failed to fetch image: ${imageUrl}`);
    const buffer = await response.arrayBuffer();
    const uploadedAsset = await targetClient.assets.upload("image", Buffer.from(buffer), {
      filename: imageUrl.split("/").pop(),
    });
    return uploadedAsset._id;
  } catch (error) {
    console.error("Error uploading image:", error.message);
    return null;
  }
}

async function migrateData() {
  console.log("Starting data migration...");

  try {
    const categoriesResponse = await fetch(`${BASE_URL}/api/products/categories`);
    if (!categoriesResponse.ok) throw new Error("Failed to fetch categories.");
    const categoriesData = await categoriesResponse.json();

    const productsResponse = await fetch(`${BASE_URL}/api/products/products`);
    if (!productsResponse.ok) throw new Error("Failed to fetch products.");
    const productsData = await productsResponse.json();

    const categoryIdMap = {};

    for (const category of categoriesData) {
      console.log(`Migrating category: ${category.name}`);
      const imageId = await uploadImageToSanity(category.image);

      const newCategory = {
        _type: "category",
        _id: `category-${category.id}`, // Ensure unique _id
        id: category.id,
        name: category.name,
        image: imageId ? { _type: "image", asset: { _ref: imageId } } : undefined,
      };

      const result = await targetClient.createOrReplace(newCategory);
      categoryIdMap[category.id] = result._id;
      console.log(`Migrated category: ${category.name} (ID: ${result._id})`);
    }

    for (const product of productsData) {
      console.log(`Migrating product: ${product.name}`);
      const imageId = await uploadImageToSanity(product.image);

      const newProduct = {
        _type: "product",
        _id: `product-${product.id}`, // Ensure unique _id
        id: product.id,
        name: product.name,
        price: product.price,
        slug: { _type: "slug", current: product.slug },
        image: imageId ? { _type: "image", asset: { _ref: imageId } } : undefined,
        category: {
          _type: "reference",
          _ref: categoryIdMap[product.category.id], // Reference to category
        },
      };

      const result = await targetClient.createOrReplace(newProduct);
      console.log(`Migrated product: ${product.name} (ID: ${result._id})`);
    }

    console.log("Data migration completed successfully!");
  } catch (error) {
    console.error("Error during migration:", error.message);
    process.exit(1);
  }
}

migrateData();
```

This script automates the process of transferring data from the provided REST APIs to my Sanity dataset.

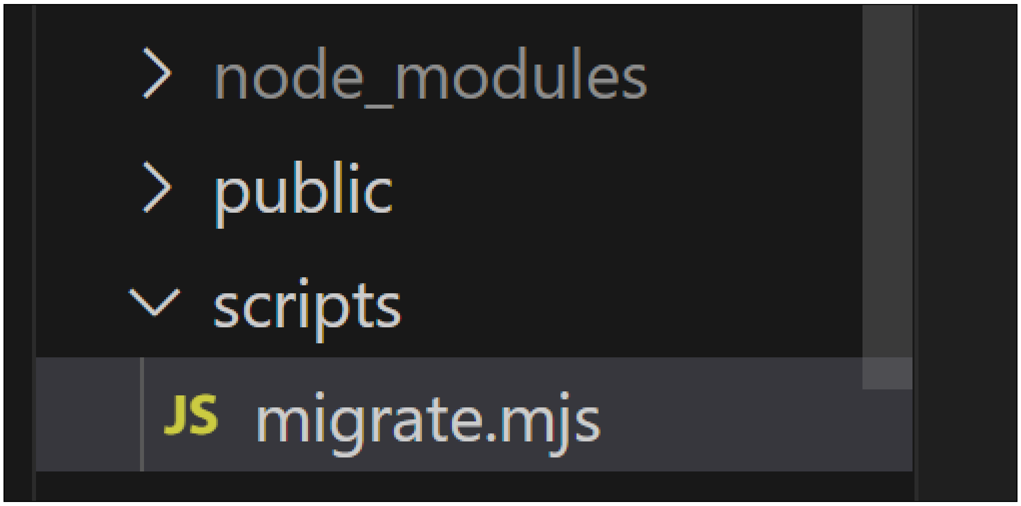
1. Setting Up Environment Variables

    1. Create a .env file in the root of your project.

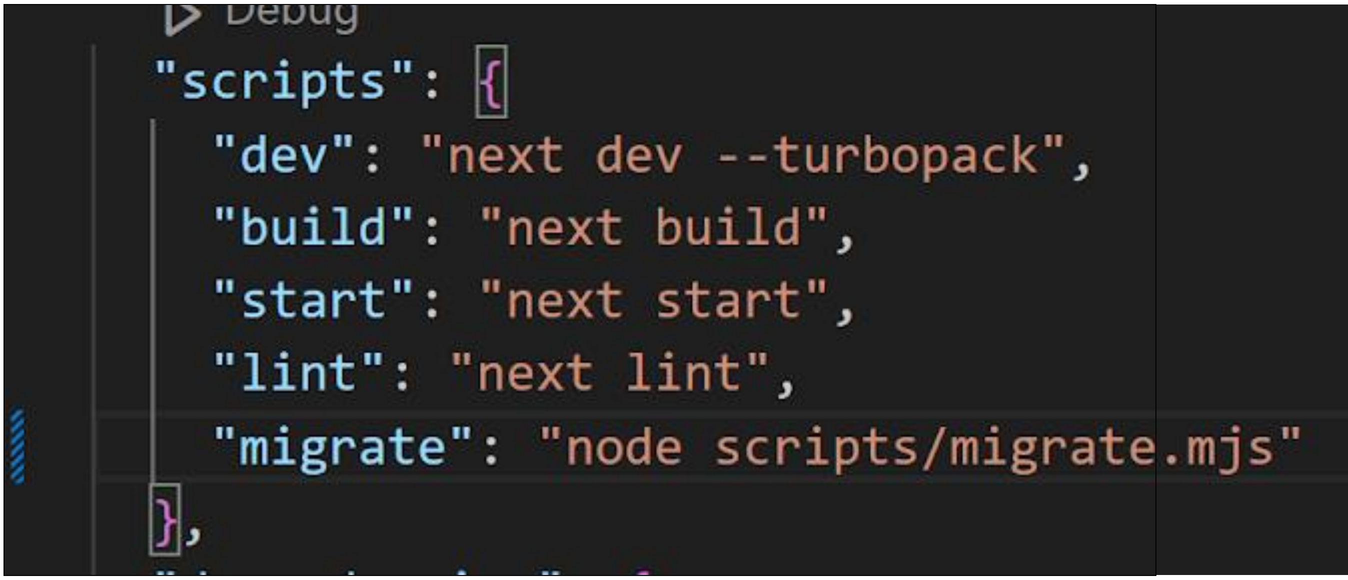
Create .env file and add the following variables:

```
NEXT_PUBLIC_SANITY_PROJECT_ID="<Project ID>" # Add your project Id
NEXT_PUBLIC_SANITY_DATASET="production"
NEXT_PUBLIC_SANITY_AUTH_TOKEN="<Auth Token>" # Add your token
```

1. Create migrate.mjs inside of the script folder



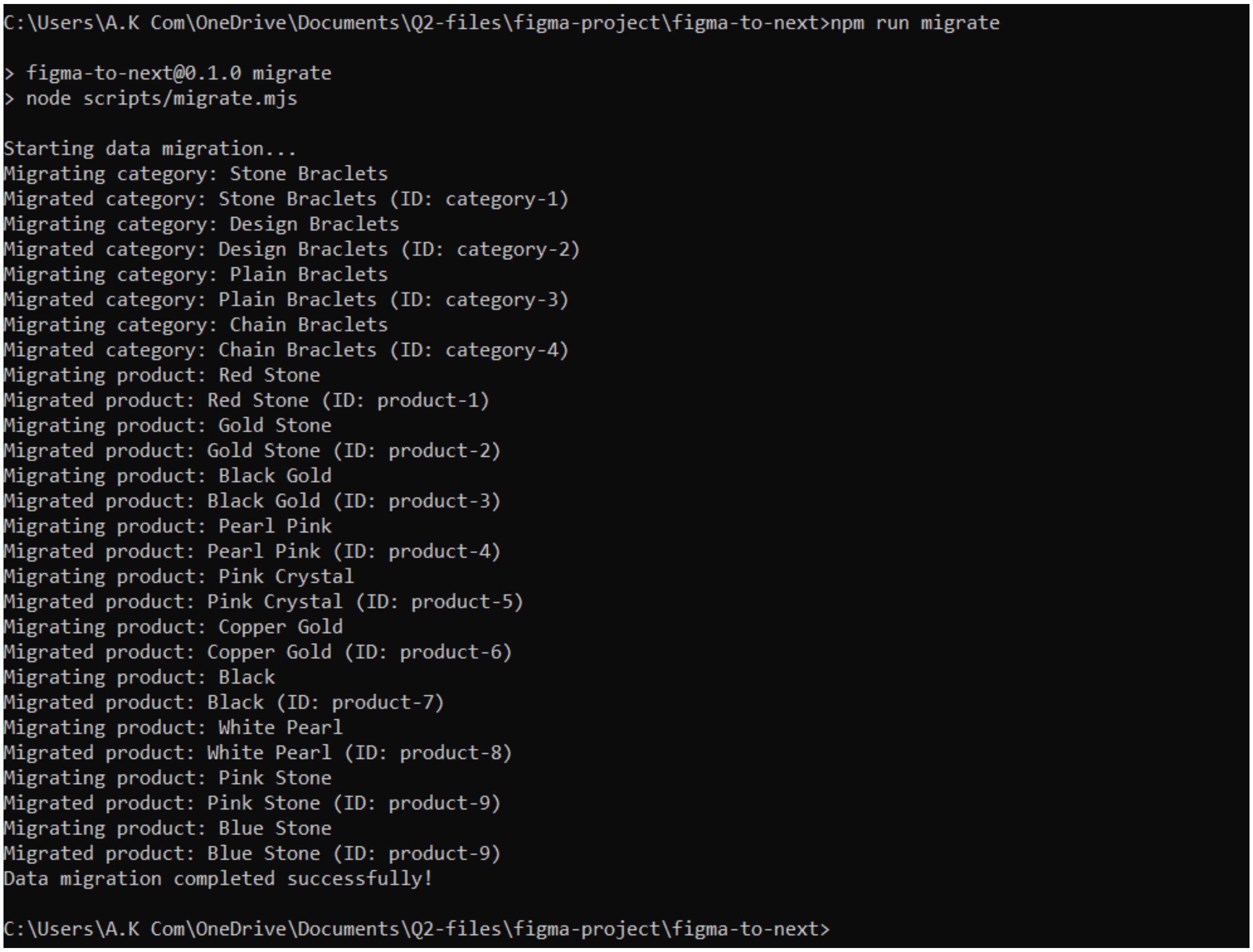
4. Open `package.json` file and add the following code inside of scripts: `"migrate": "node scripts/migrate.mjs"`



```
"scripts": {
  "dev": "next dev --turbopack",
  "build": "next build",
  "start": "next start",
  "lint": "next lint",
  "migrate": "node scripts/migrate.mjs"
},
```

5. Install the following package before running the script
   1. npm install dotenv

6. Now run the command npm run migrate
7. This will insert the data from the rest api to your sanity studio.



```
C:\Users\A.K Com\OneDrive\Documents\Q2-files\figma-project\figma-to-next>npm run migrate

> figma-to-next@0.1.0 migrate
> node scripts/migrate.mjs

Starting data migration...
Migrating category: Stone Braclets
Migrated category: Stone Braclets (ID: category-1)
Migrating category: Design Braclets
Migrated category: Design Braclets (ID: category-2)
Migrating category: Plain Braclets
Migrated category: Plain Braclets (ID: category-3)
Migrating category: Chain Braclets
Migrated category: Chain Braclets (ID: category-4)
Migrating product: Red Stone
Migrated product: Red Stone (ID: product-1)
Migrating product: Gold Stone
Migrated product: Gold Stone (ID: product-2)
Migrating product: Black Gold
Migrated product: Black Gold (ID: product-3)
Migrating product: Pearl Pink
Migrated product: Pearl Pink (ID: product-4)
Migrating product: Pink Crystal
Migrated product: Pink Crystal (ID: product-5)
Migrating product: Copper Gold
Migrated product: Copper Gold (ID: product-6)
Migrating product: Black
Migrated product: Black (ID: product-7)
Migrating product: White Pearl
Migrated product: White Pearl (ID: product-8)
Migrating product: Pink Stone
Migrated product: Pink Stone (ID: product-9)
Migrating product: Blue Stone
Migrated product: Blue Stone (ID: product-9)
Data migration completed successfully!

C:\Users\A.K Com\OneDrive\Documents\Q2-files\figma-project\figma-to-next>
```

Rest API endpoint for Details:

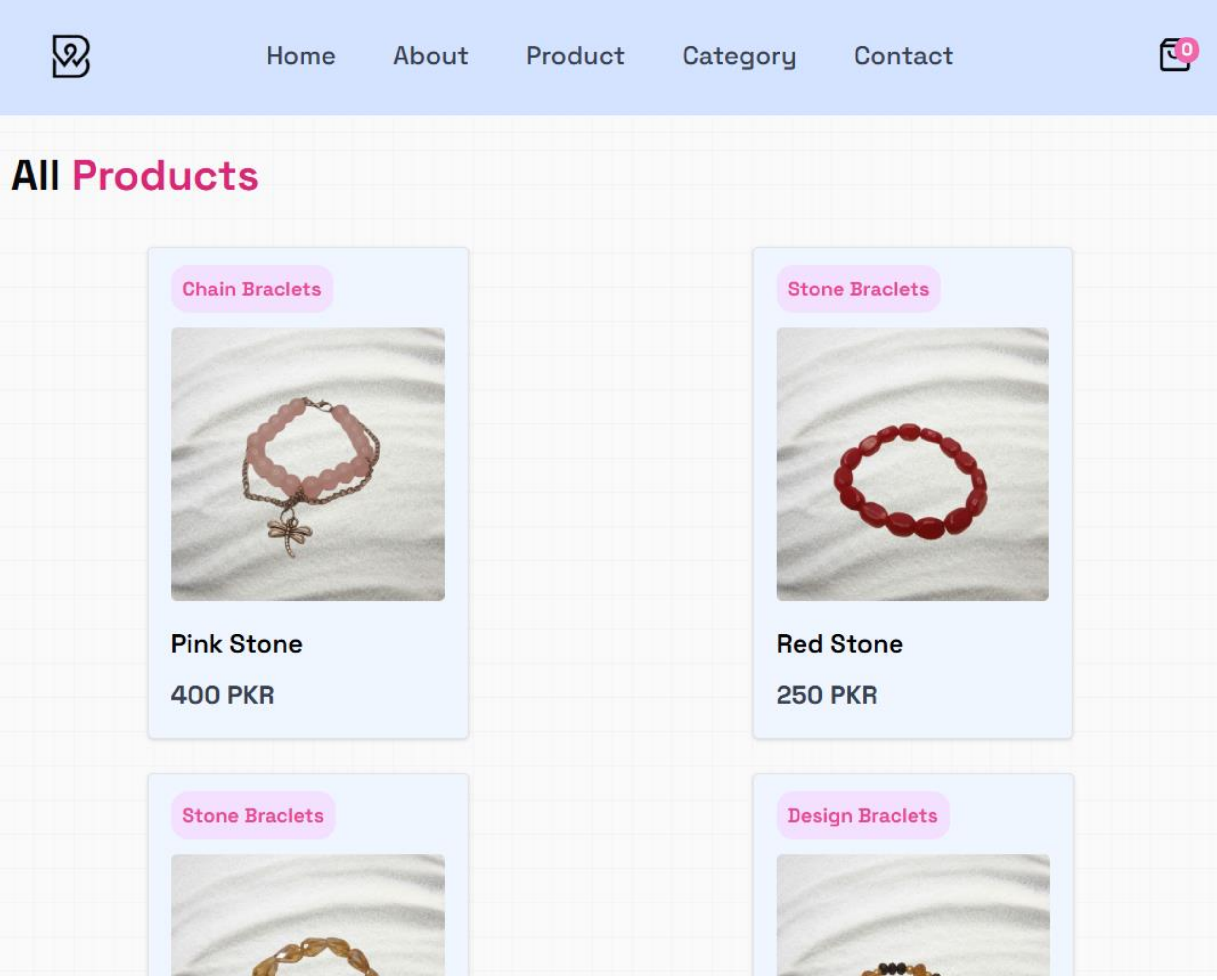
Products API

- Endpoint: https://678b90681a6b89b27a2abdda.mockapi.io/api/products/products

Categories API:

- Endpoint: https://678b90681a6b89b27a2abdda.mockapi.io/api/products/categories

Data displayed on Frontend:



All Products

Chain Braclets

Pink Stone

400 PKR

Stone Braclets

Red Stone

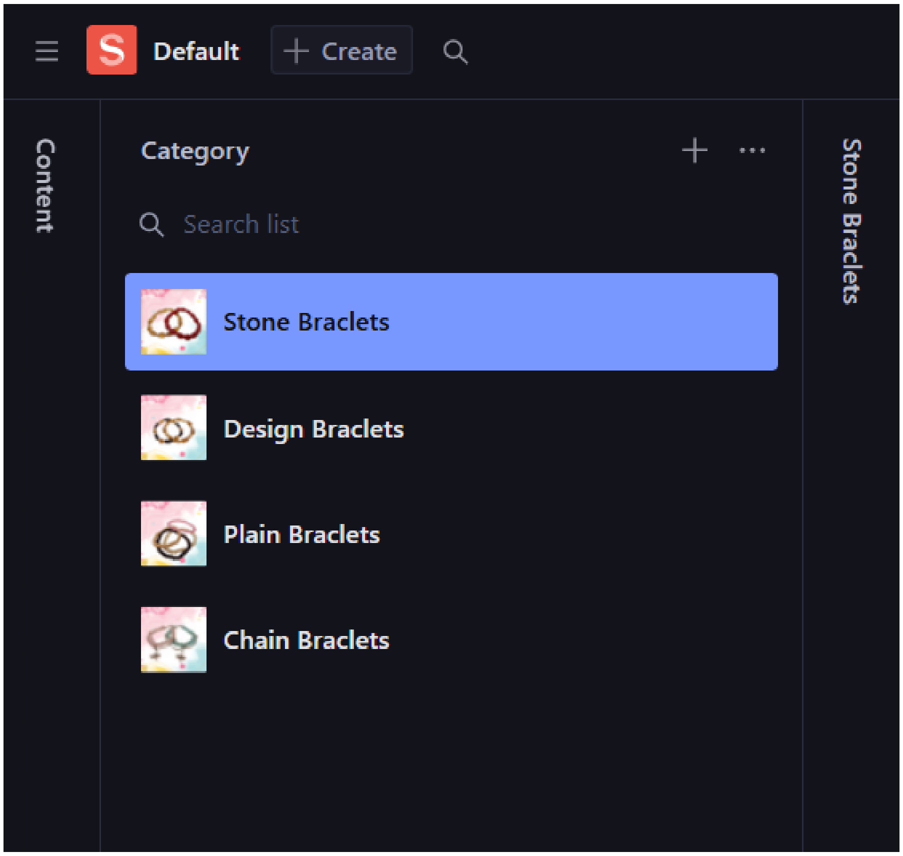
250 PKR

Stone Braclets

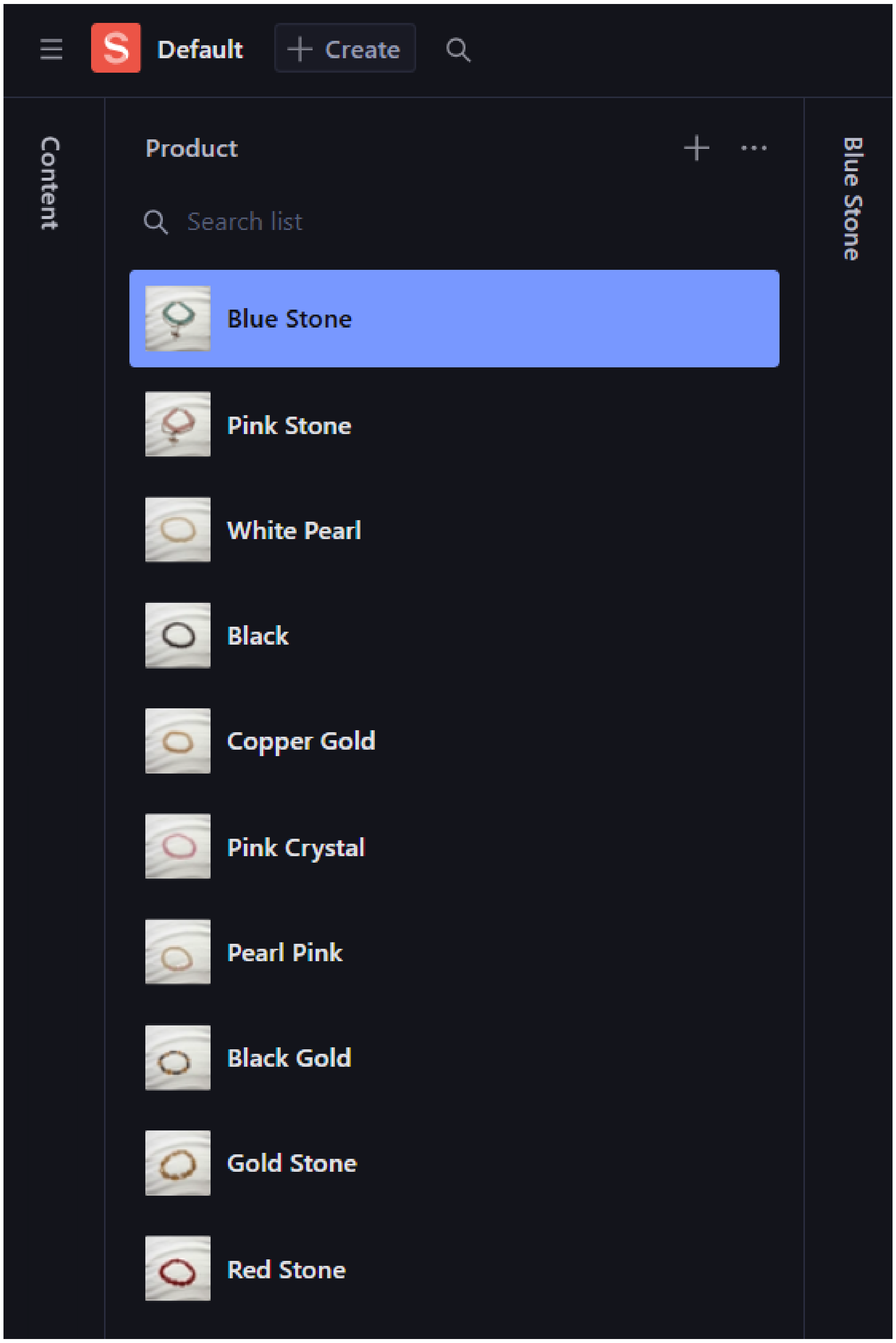
Design Braclets

Data In Sanity (studio):

Categories:


S
Default
+ Create
Content
Category
+
···
Search list
Stone Braclets
Design Braclets
Plain Braclets
Chain Braclets
Stone Braclets

Products:



Blue Stone
Pink Stone
White Pearl
Black
Copper Gold
Pink Crystal
Pearl Pink
Black Gold
Gold Stone
Red Stone

Document by Ilsa Ubaid ***(****Senior Student GIAIC****)***