Day-3 Market-Place Ecommerce Website

API-Integration & Data-Migration----ShopifyX

API Integration:

```js
import { createClient } from "@sanity/client";
import axios from "axios";
import dotenv from "dotenv";
import { fileURLToPath } from "url";
import path from "path";

 Load environment variables from .env.local
const __filename = fileURLToPath(import.meta.url);
const __dirname = path.dirname(__filename);
dotenv.config({ path: path.resolve(__dirname, "../.env.local") });

    Create Sanity client

const client = createClient({
 projectId: process.env.NEXT_PUBLIC_SANITY_PROJECT_ID,
 dataset: process.env.NEXT_PUBLIC_SANITY_DATASET,
 useCdn: false,
 token: process.env.SANITY_API_TOKEN,
 apiVersion: "2021-08-31", Match with your API version
});

   Upload image to Sanity
async function uploadImageToSanity(imageUrl) {
 try {
   console.log(`Uploading image: ${imageUrl}`);
   const response = await axios.get(imageUrl, { responseType: "arraybuffer" });
   const buffer = Buffer.from(response.data);
   const asset = await client.assets.upload("image", buffer, {
     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;
 }
}  Import Data into Sanity

async function importData() {
 try {                                       API Endpoint
   console.log("Fetching products from API...");
   const response = await axios.get("https://fakestoreapi.com/products");
   const products = response.data;
   console.log(`Fetched ${products.length} products`);

   for (const product of products) {
     console.log(`Processing product: ${product.title}`);
     Upload image and get reference

     let imageRef = null;
     if (product.image) {
       imageRef = await uploadImageToSanity(product.image);
     }
     const sanityProduct = {
       _type: "product",
       name: product.title,
       slug: {
         _type: "slug",
         current: product.title.toLowerCase().replace(/\s+/g, "-"),
       },
       description: product.description,
       price: product.price,
       discount: 0,
       categories: [],
       stock: Math.floor(Math.random() * 50) + 1,
       label: product.category || "",
       status: "new",
       image: imageRef
         ? {
             _type: "image",
             asset: {
               _type: "reference",
               _ref: imageRef,
             },
           }
         : undefined,
     } Check if the category exists in Sanity

     if (product.category) {
       const existingCategory = await client.fetch(
         `*[_type == "category" && title == $title][0]._id`,
         { title: product.category }
       );
       if (existingCategory) {
         sanityProduct.categories.push({
           _type: "reference",
           _ref: existingCategory,
         });
       }
       console.log(product.category);
     }

     console.log("Uploading product to Sanity:", sanityProduct.name);
     const result = await client.create(sanityProduct);
     console.log(`Product uploaded successfully: ${result._id}`);
   }

   console.log("Data import completed successfully!");
 } catch (error) {
   console.error("Error importing data:", error);
 }
}

importData();
```

import Data successfully!

Data Comes on Frontend: