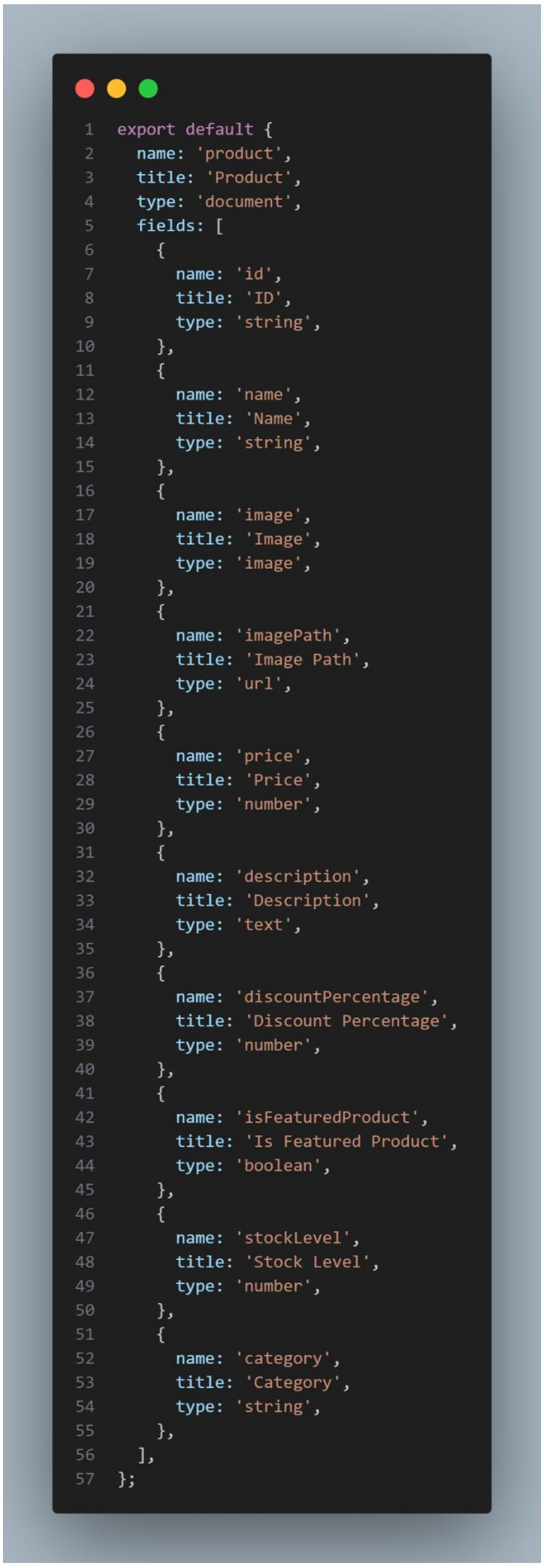
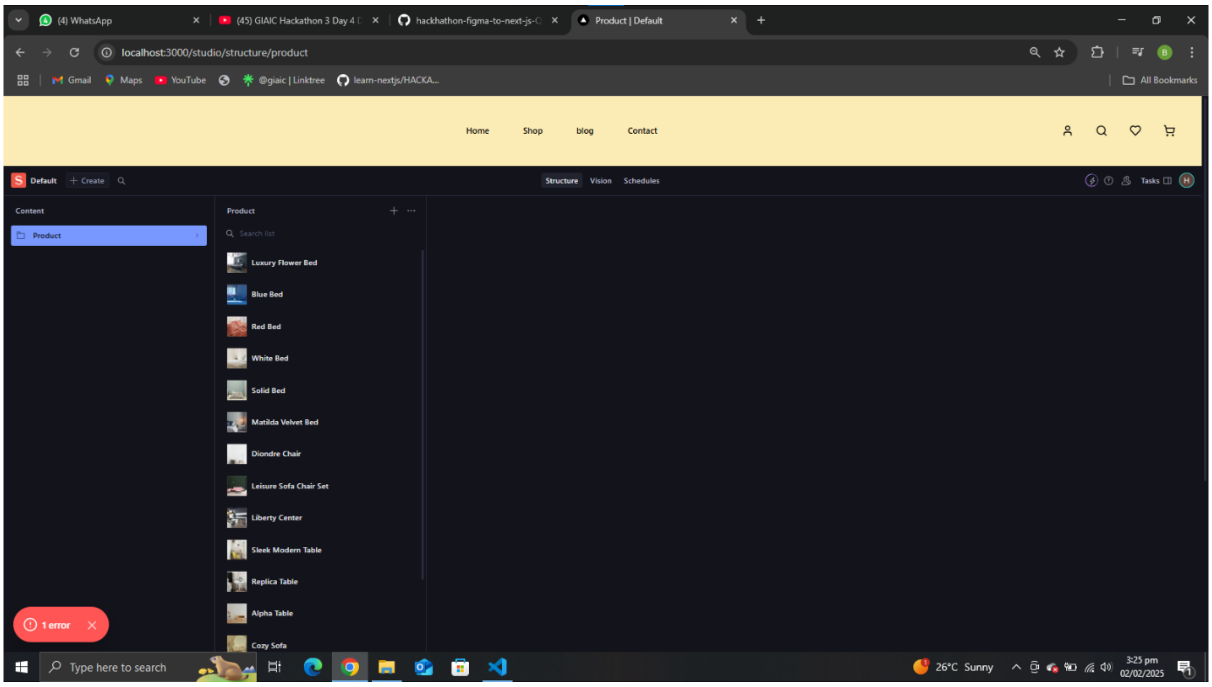
HACKTHOON DAY 3

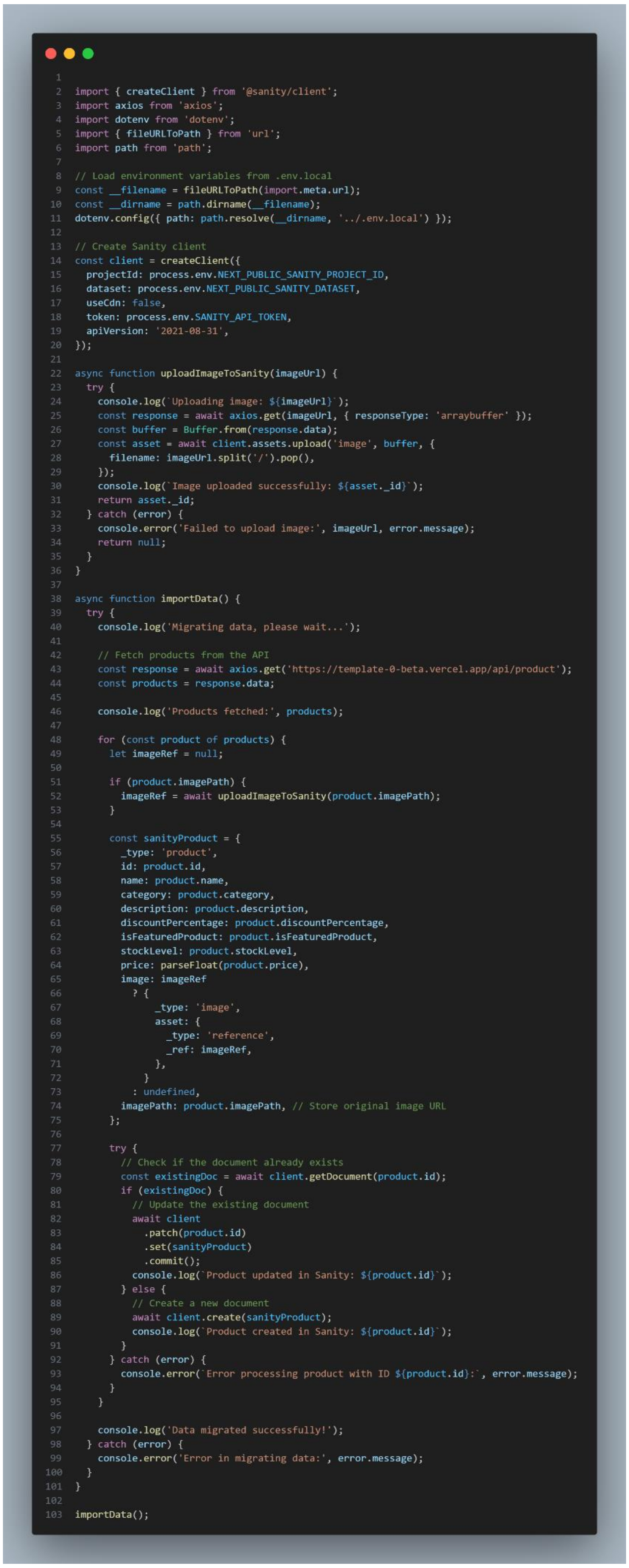
Task Api integration and Data fetching



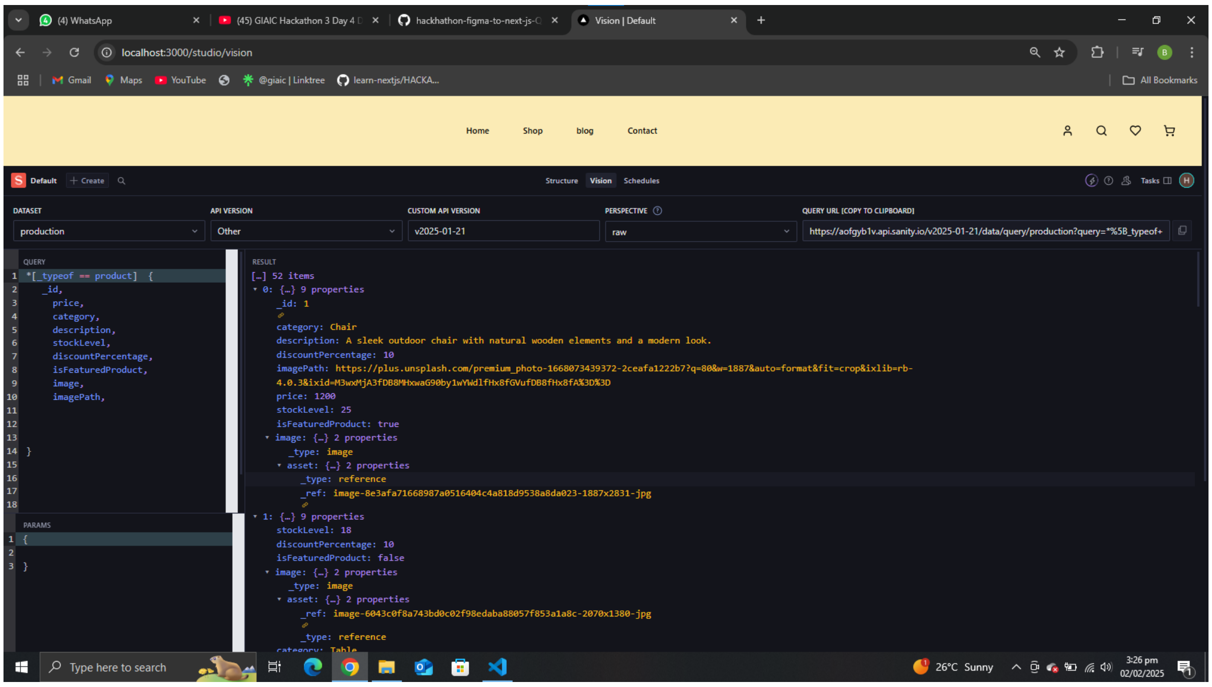
```
export default {
  name: 'product',
  title: 'Product',
  type: 'document',
  fields: [
    {
      name: 'id',
      title: 'ID',
      type: 'string',
    },
    {
      name: 'name',
      title: 'Name',
      type: 'string',
    },
    {
      name: 'image',
      title: 'Image',
      type: 'image',
    },
    {
      name: 'imagePath',
      title: 'Image Path',
      type: 'url',
    },
    {
      name: 'price',
      title: 'Price',
      type: 'number',
    },
    {
      name: 'description',
      title: 'Description',
      type: 'text',
    },
    {
      name: 'discountPercentage',
      title: 'Discount Percentage',
      type: 'number',
    },
    {
      name: 'isFeaturedProduct',
      title: 'Is Featured Product',
      type: 'boolean',
    },
    {
      name: 'stockLevel',
      title: 'Stock Level',
      type: 'number',
    },
    {
      name: 'category',
      title: 'Category',
      type: 'string',
    },
  ],
};
```



Luxury Flower Bed
Blue Bed
Red Bed
White Bed
Solid Bed
Matilda Velvet Bed
Diondre Chair
Leisure Sofa Chair Set
Liberty Center
Sleek Modern Table
Replica Table
Alpha Table
Cozy Sofa



```javascript

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',
});

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.message);
    return null;
  }
}

async function importData() {
  try {
    console.log('Migrating data, please wait...');

    // Fetch products from the API
    const response = await axios.get('https://template-0-beta.vercel.app/api/product');
    const products = response.data;

    console.log('Products fetched:', products);

    for (const product of products) {
      let imageRef = null;

      if (product.imagePath) {
        imageRef = await uploadImageToSanity(product.imagePath);
      }

      const sanityProduct = {
        _type: 'product',
        id: product.id,
        name: product.name,
        category: product.category,
        description: product.description,
        discountPercentage: product.discountPercentage,
        isFeaturedProduct: product.isFeaturedProduct,
        stockLevel: product.stockLevel,
        price: parseFloat(product.price),
        image: imageRef
          ? {
              _type: 'image',
              asset: {
                _type: 'reference',
                _ref: imageRef,
              },
            }
          : undefined,
        imagePath: product.imagePath, // Store original image URL
      };

      try {
        // Check if the document already exists
        const existingDoc = await client.getDocument(product.id);
        if (existingDoc) {
          // Update the existing document
          await client
            .patch(product.id)
            .set(sanityProduct)
            .commit();
          console.log(`Product updated in Sanity: ${product.id}`);
        } else {
          // Create a new document
          await client.create(sanityProduct);
          console.log(`Product created in Sanity: ${product.id}`);
        }
      } catch (error) {
        console.error(`Error processing product with ID ${product.id}:`, error.message);
      }
    }

    console.log('Data migrated successfully!');
  } catch (error) {
    console.error('Error in migrating data:', error.message);
  }
}

importData();
```



DATASET: production

API VERSION: Other

CUSTOM API VERSION: v2025-01-21

PERSPECTIVE: raw

QUERY URL [COPY TO CLIPBOARD]: https://aofgyb1v.api.sanity.io/v2025-01-21/data/query/production?query=*%5B_typeof+

QUERY

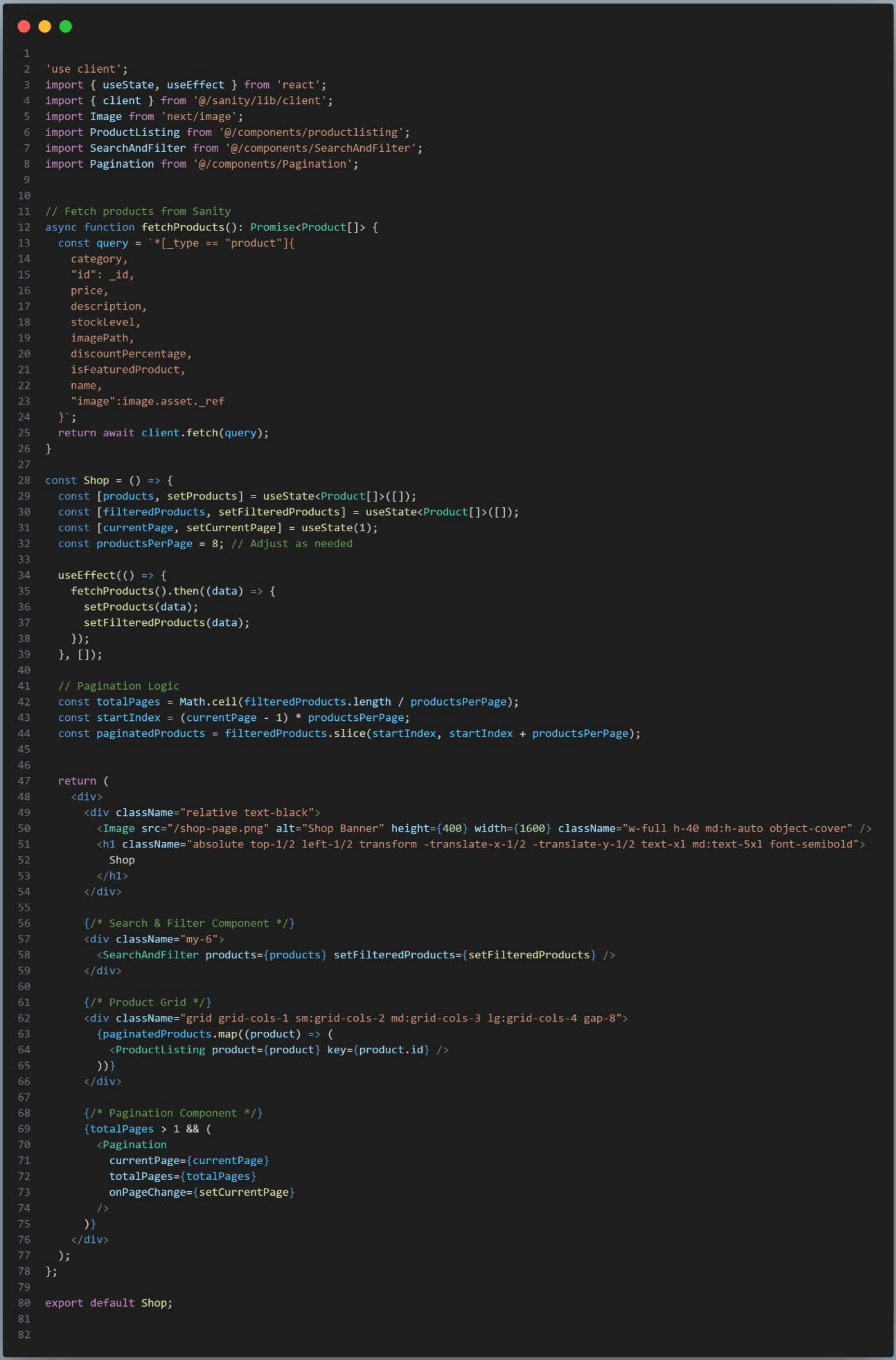
```
*[_typeof == product]  {
  _id,
    price,
    category,
    description,
    stockLevel,
    discountPercentage,
    isFeaturedProduct,
    image,
    imagePath,



}
```

PARAMS

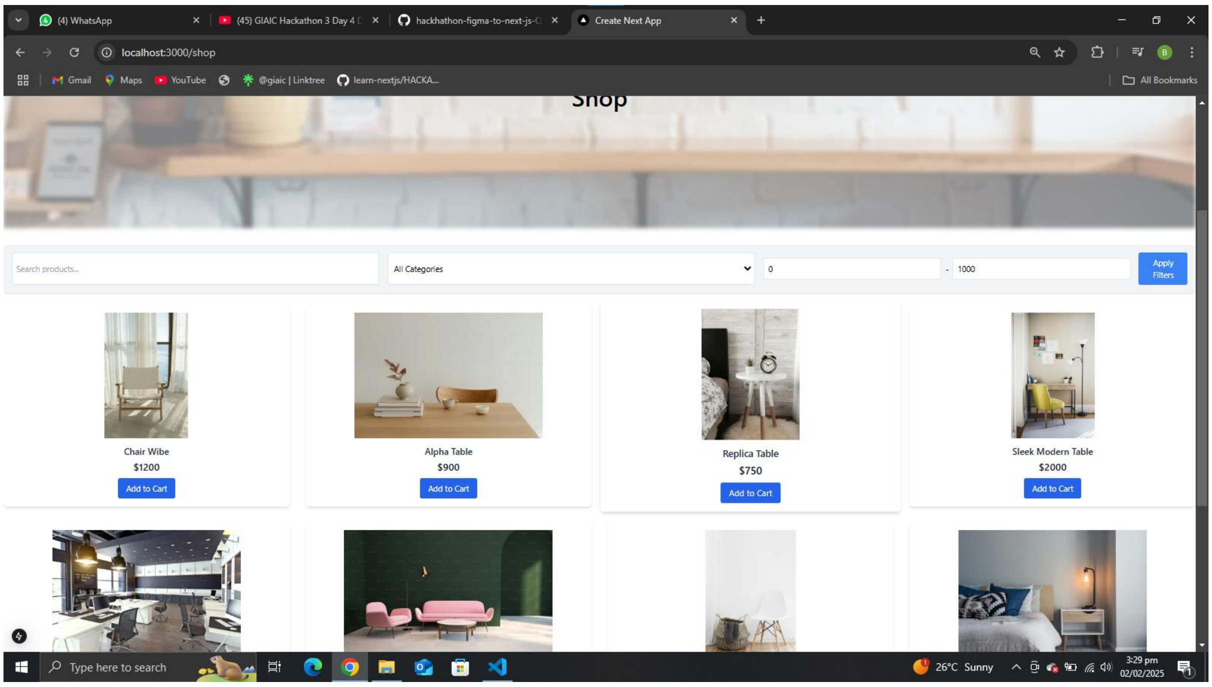
```
{

}
```

RESULT

```
[…] 52 items
▾ 0: {…} 9 properties
    _id: 1
    category: Chair
    description: A sleek outdoor chair with natural wooden elements and a modern look.
    discountPercentage: 10
    imagePath: https://plus.unsplash.com/premium_photo-1668073439372-2ceafa1222b7?q=80&w=1887&auto=format&fit=crop&ixlib=rb-
    4.0.3&ixid=M3wxMjA3fDB8MHxwaG90by1wYWdlfHx8fGVufDB8fHx8fA%3D%3D
    price: 1200
    stockLevel: 25
    isFeaturedProduct: true
  ▾ image: {…} 2 properties
      _type: image
    ▾ asset: {…} 2 properties
        _type: reference
        _ref: image-8e3afa71668987a0516404c4a818d9538a8da023-1887x2831-jpg
▾ 1: {…} 9 properties
    stockLevel: 18
    discountPercentage: 10
    isFeaturedProduct: false
  ▾ image: {…} 2 properties
      _type: image
    ▾ asset: {…} 2 properties
        _ref: image-6043c0f8a743bd0c02f98edaba88057f853a1a8c-2070x1380-jpg
        _type: reference
    category: Table
```



```
'use client';
import { useState, useEffect } from 'react';
import { client } from '@/sanity/lib/client';
import Image from 'next/image';
import ProductListing from '@/components/productlisting';
import SearchAndFilter from '@/components/SearchAndFilter';
import Pagination from '@/components/Pagination';


// Fetch products from Sanity
async function fetchProducts(): Promise<Product[]> {
  const query = `*[_type == "product"]{
    category,
    "id": _id,
    price,
    description,
    stockLevel,
    imagePath,
    discountPercentage,
    isFeaturedProduct,
    name,
    "image":image.asset._ref
  }`;
  return await client.fetch(query);
}

const Shop = () => {
  const [products, setProducts] = useState<Product[]>([]);
  const [filteredProducts, setFilteredProducts] = useState<Product[]>([]);
  const [currentPage, setCurrentPage] = useState(1);
  const productsPerPage = 8; // Adjust as needed

  useEffect(() => {
    fetchProducts().then((data) => {
      setProducts(data);
      setFilteredProducts(data);
    });
  }, []);

  // Pagination Logic
  const totalPages = Math.ceil(filteredProducts.length / productsPerPage);
  const startIndex = (currentPage - 1) * productsPerPage;
  const paginatedProducts = filteredProducts.slice(startIndex, startIndex + productsPerPage);


  return (
    <div>
      <div className="relative text-black">
        <Image src="/shop-page.png" alt="Shop Banner" height={400} width={1600} className="w-full h-40 md:h-auto object-cover" />
        <h1 className="absolute top-1/2 left-1/2 transform -translate-x-1/2 -translate-y-1/2 text-xl md:text-5xl font-semibold">
          Shop
        </h1>
      </div>

      {/* Search & Filter Component */}
      <div className="my-6">
        <SearchAndFilter products={products} setFilteredProducts={setFilteredProducts} />
      </div>

      {/* Product Grid */}
      <div className="grid grid-cols-1 sm:grid-cols-2 md:grid-cols-3 lg:grid-cols-4 gap-8">
        {paginatedProducts.map((product) => (
          <ProductListing product={product} key={product.id} />
        ))}
      </div>

      {/* Pagination Component */}
      {totalPages > 1 && (
        <Pagination
          currentPage={currentPage}
          totalPages={totalPages}
          onPageChange={setCurrentPage}
        />
      )}
    </div>
  );
};

export default Shop;
```



Shop

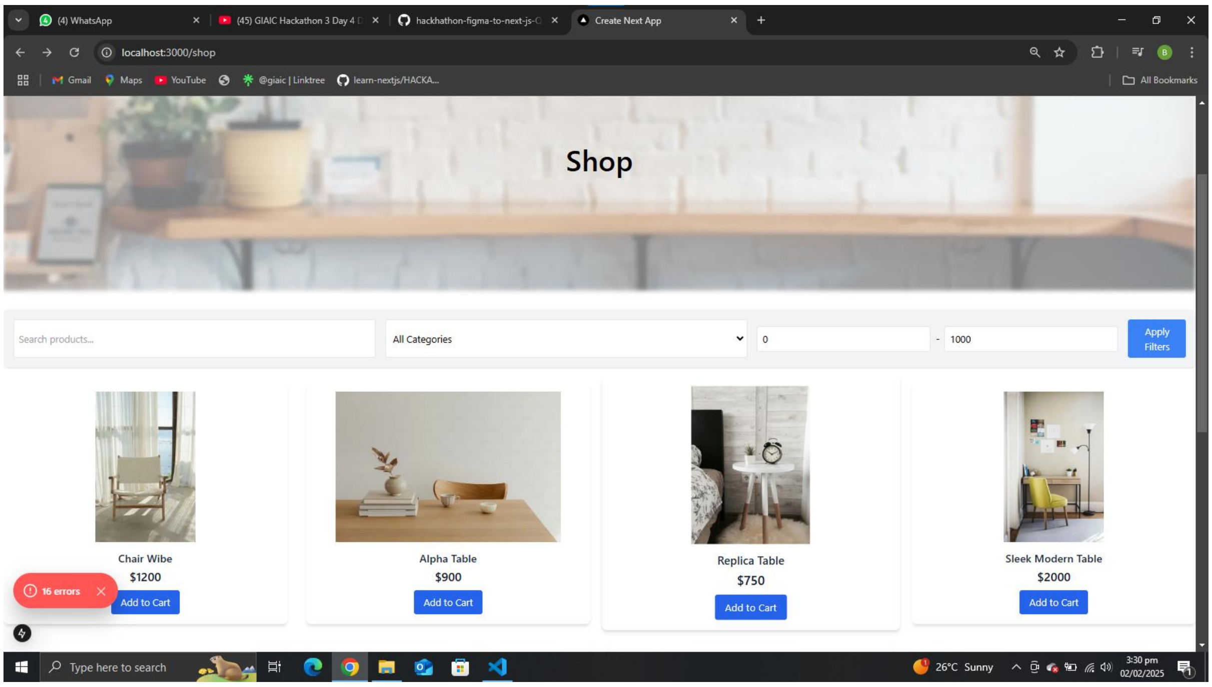
Search products... | All Categories | 0 - 1000 | Apply Filters

Chair Wibe
$1200
Add to Cart

Alpha Table
$900
Add to Cart

Replica Table
$750
Add to Cart

Sleek Modern Table
$2000
Add to Cart



Shop

Search products... | All Categories | 0 - 1000 | Apply Filters

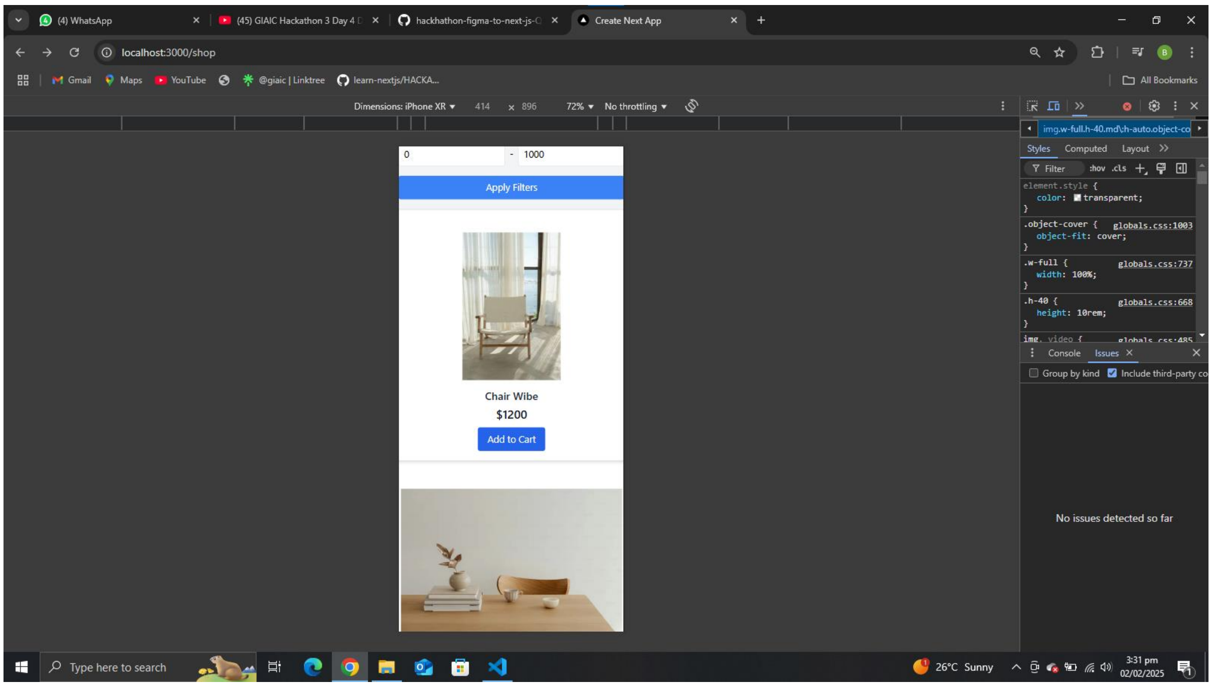
Chair Wibe
$1200
Add to Cart

Alpha Table
$900
Add to Cart

Replica Table
$750
Add to Cart

Sleek Modern Table
$2000
Add to Cart

16 errors



```
element.style {
  color: transparent;
}
.object-cover {                globals.css:1003
  object-fit: cover;
}
.w-full {                       globals.css:737
  width: 100%;
}
.h-40 {                         globals.css:668
  height: 10rem;
}
img, video {                    globals.css:485
```