

## DAY:3 API INTEGRATION REPORT OF FURNITURE:

# Next.js + Sanity E-commerce Project Documentation

## Table of Contents:

1. Project Setup
2. Environment Configuration
3. Sanity Studio Setup
4. Schema Configuration
5. Client-Side Implementation
6. Deployment Steps

## 1. Project Setup

### Initial Setup

```
# Create a new Next.js project with TypeScriptnpx create-next-app@latest test6 test6# Answer the setup questions: ✓ Would you like to use TypeScript? Yes ✓ Would you like to use ESLint? Yes ✓ Would you like to use Tailwind CSS? Yes ✓ Would you like to use `src/` directory? Yes ✓ Would you like to use App Router? Yes ✓ Would you like to customize the default import alias? No
```

### Install Required Dependencies

bash

Copy

```
npm install next-sanity @sanity/image-url @sanity/vision sanity @portabletext/react
```

## 2. Environment Configuration

Create a `.env.local` file in your root directory:

env

```
NEXT_PUBLIC_SANITY_PROJECT_ID="your_project_id"NEXT_PUBLIC_SANITY_DATASET="production"
```

API Url : <https://template6-six.vercel.app/api/products>

□ Schema: <https://github.com/developer-hammad>

[rehman/template6/blob/main/src/sanity/schemaTypes/product.ts](https://github.com/developer-hammad/blob/main/src/sanity/schemaTypes/product.ts)

BY AMEEN ALAM

4

□ Migration: <https://github.com/developer-hammad>

[rehman/template6/blob/main/importData.js](https://github.com/developer-hammad/blob/main/importData.js)

### 3. Sanity Studio Setup

Initialize Sanity

```
# In your project rootnpm create sanity@latest -- --template clean --create-project "my Project Name" --dataset production
```

Configure Sanity Studio

Create or update `sanity.config.ts`:

typescript

```
import { defineConfig } from 'sanity';import { deskTool } from 'sanity/desk';import { schemas } from './sanity/schemas';export default defineConfig({ name: 'default', title: 'Your Project Name', projectId: process.env.NEXT_PUBLIC_SANITY_PROJECT_ID!, dataset: process.env.NEXT_PUBLIC_SANITY_DATASET!, basePath: '/studio', plugins: [deskTool()], schema: { types: schemas } });
```

### 4. Schema Configuration

Product Schema (`sanity/schemas/product.ts`)

```
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: "dicountPercentage",          type:
"number",          title: "Discount Percentage",      },
{      name: "isNew",          type: "boolean",
title: "New Badge",      }      ]});

```

## 5. Client-Side Implementation

Setup Sanity Client (`sanity/lib/client.ts`)

```

import { createClient } from 'next-sanity'; export const client
= createClient({ projectId:
process.env.NEXT_PUBLIC_SANITY_PROJECT_ID, dataset:
process.env.NEXT_PUBLIC_SANITY_DATASET, apiVersion: "2023-10-
10", useCdn: true,});

```

Setup Image Builder (`sanity/lib/image.ts`)

```

import imageUrlBuilder from '@sanity/image-url'; import
{ client } from './client'; const builder =
imageUrlBuilder(client); export function urlForImage(source:
any) { return builder.image(source); }

```

Product Type Definition (`types/product.ts`)

```

export type Product = { _id: string; title: string;
description: string; productImage: { asset: { _ref:
string; _type: 'reference'; }; }; price: number;
tags?: string[]; dicountPercentage?: number; isNew?:
boolean; };

```

Query Setup (`lib/queries.ts`)

```

export const productsQuery = `*[_type == "product"] { _id,
title, description, productImage, price, tags,
dicountPercentage, isNew}`;

```

## 6. Deployment Steps

1. Build my project:

```
npm run build
```

1. Deploy in Vercel:

```
# Install Vercel CLI if not installednpm i -g vercel#  
Deployvercel
```

1. Set up environment variables in Vercel:

1. Go to your project settings in Vercel
2. Add the environment variables from your `.env.local`

## Additional Notes

### Project Structure

Copy

```
your-project/├── src/│   ├── app/│   │   ├── layout.tsx│   │   └── page.tsx│   ├── components/│   │   ├── ProductCard.tsx│   │   └── client.ts│   ├── lib/│   │   ├── schemas/│   │   │   └── product.ts│   │   └── product.ts│   ├── public/│   │   └── .env.local│   └── sanity.config.ts└── package.json
```

### Running the Project

Development:

bash

Copy

```
# Run Next.js development servernpm run dev# Run Sanity  
Studionpm run studio
```

Production:

bash

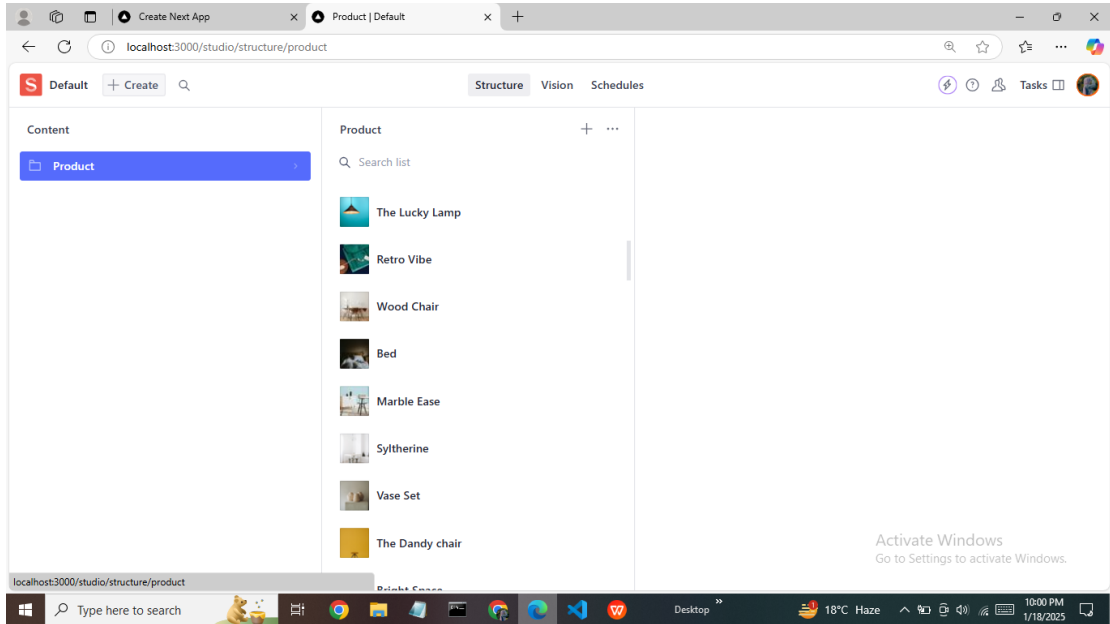
Copy

```
npm run buildnpm start
```

## THEN PRODUCT.TS (SCHEMA)

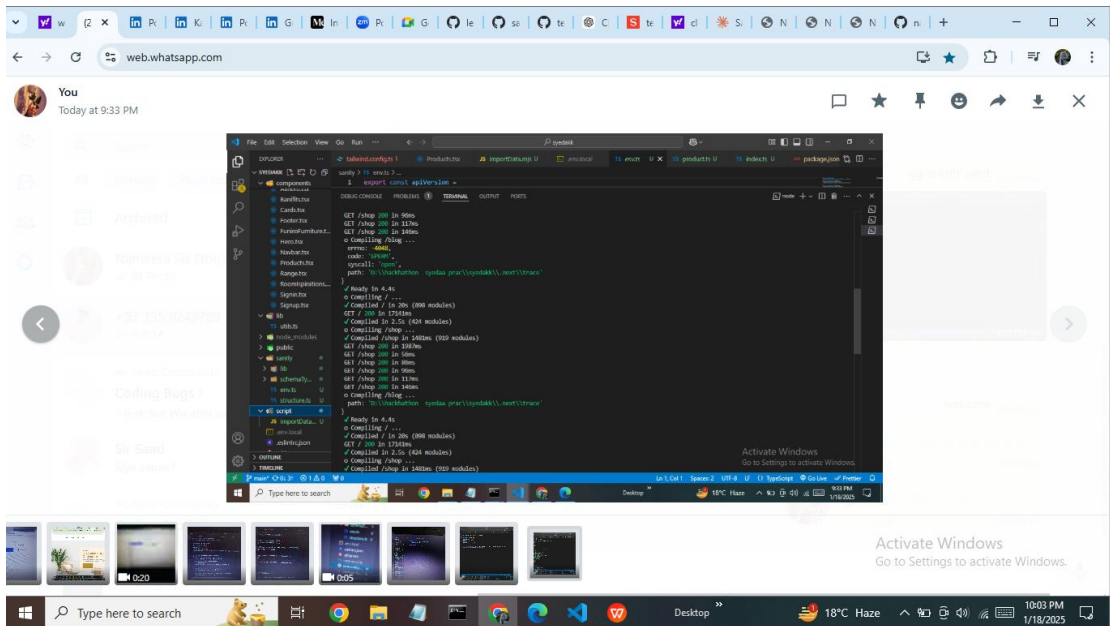
```
import { defineType } from "sanity"

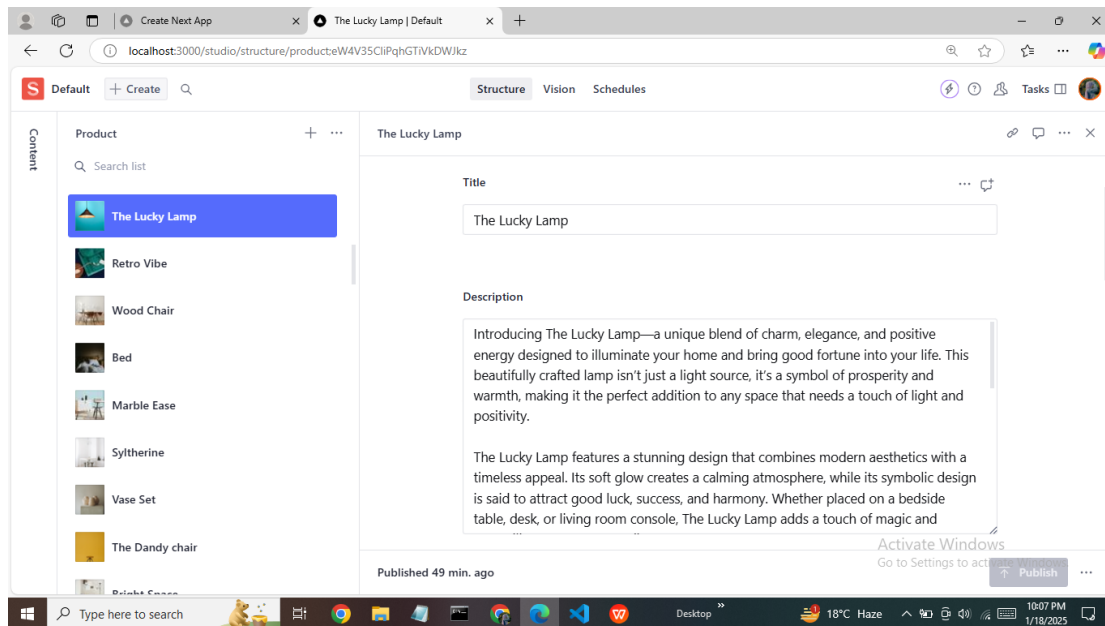
export const product = defineType({
  name: "product",
  title: "Product",
  type: "document",
  fields: [
    {
      name: "title",
      title: "Title",
      validation: (rule) => rule.required(),
      type: "string"
    },
    {
      name: "description",
      type: "text",
      validation: (rule) => 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: "dicountPercentage",
      type: "number",
      title: "Discount Percentage",
    },
    {
      name: "isNew",
      type: "boolean",
      title: "New Badge",
    }
  ]
})
```

$$\left. \begin{array}{l} \text{[} \\ \text{)} \end{array} \right\} \text{ )}$$


## FETCHING SHOW IN TERMINAL AS WELL:

--	--	--





PICTURES SHOW IN FRONTEND:

