day3-techinical-foundation\note

Task Overview: I have manually entered data into a Sanity CMS and have created three schemas:

Product Schema Order Schema Category Schema Sales Schema After defining the schemas, you successfully fetched the data and displayed it in the browser.

Detailed Steps Explanation:

Manual Data Entry in Sanity CMS:

I manually input data for the schemas directly into Sanity Studio, a headless CMS that allows easy management of structured content. Schema Creation:

Product Schema: Defines the structure for storing product-related information like name, price, description, images, and any other product attributes. Order Schema: Tracks information about customer orders, including order ID, customer details, items purchased, total cost, and order status. Category Schema: Organizes products into categories for better accessibility and navigation, such as "Furniture," Sales Schema: Captures details about sales transactions, including sales ID, product(s) sold, quantity, total revenue, and date of sale. Data Fetching:

Once the schemas were set up and data was entered, you wrote logic to fetch the data from Sanity CMS and display it in the browser. This process involved querying the Sanity backend using Sanity's GROQ (Graph-Relational Object Queries) or a client-side library like @sanity/client. Browser Display:

The fetched data is rendered in the browser, likely using a frontend framework such as React, Next.js, Proper UI/UX was implemented to ensure the data is visually structured and user-friendly. Suggestions for Enhancement:

Implement search and filter functionality on the frontend to allow users to navigate through products or orders more efficiently. Consider adding relationships between schemas (e.g., linking products to categories or sales to orders) to better manage interconnected data. Use Sanity’s real-time updates feature to ensure the browser reflects any changes made to the data in Sanity CMS immediately. Optimize I fetching logic with pagination or limit queries to improve performance, especially for large datasets.