HACKATHON 3 - DAY 2: PLANNING THE TECHNICAL FOUNDATION

16/1/24

FRONTEND (NEXT.JS)

1. Objective:

Build a fast and easy-to-use website for ordering groceries and food products.

Main Pages:

* **Home:** Show featured products, categories (e.g., Fruits, Snacks, Beverages), and special offers.

* **Product Listing:** Display products filtered by categories.

* **Product Details**: Provide details like price, availability, and nutritional information.

* **Cart:** Let users review, add, or remove items before checkout.

* **Checkout**: Collect delivery address and payment details.

* **Order Confirmation**: Show order details and delivery time estimate.

2. Key Features:

* Use dynamic routing (e.g., / products/: id) for individual product pages.

* Fetch data from **Sanity CMS** using **GROQ queries**. Example:

[_type == "product"] {

_id, name, price, category-›name, images, stock

}

* Make the website mobile-friendly for easy ordering on the go.

3. Backend (Sanity CMS)

Objective:
Manage food product data, categories, and orders efficiently.
Tasks:
* **Product Data:**
Add fields like name, price, stock, image, category, and expiry date.

* **Categories**:
Group products into categories like
Vegetables, Frozen Foods, or Beverages.

* **Orders:**
Store customer information, ordered products, and payment status.

4. Interaction Between Frontend and Backend

How it Works:

* Fetching Data:

The frontend uses **GROQ queries** to get product and category data from **Sanity.**

Example:

**[_type == "category"] {*

_id,

name, description

* Submitting Orders:

When a customer places an order, their details are sent to sanity using a POST request

4. API Endpoints

Endpoints:

* ***/products:***

* Method: GET

* Purpose: Get all available food products.

* Response: Product name, price, stock, and category.

***** ***/categories:***

* Method: GET

* Purpose: Get all product categories.

* Response: Category name and description.

***** ***/orders:***

* Method: POST

* Purpose: Save customer orders.

* Payload: Customer name, address, ordered items, and payment status.

5. System Architecture

Overview:

* **Frontend (Next.js):** Displays products, handles user interactions, and communicates with the backend.

* **Backend (Sanity CMS):** Stores product data, categories, and orders. Updates the frontend in real-time.

* **APIs:** Connect the frontend and backend to share data like product info categories, and orders.