Hackathon 3 - Day 2: Planning the Technical Foundation

16/1/24

# Frontend (Next.js)

## Objective:

|  |  |
| --- | --- |
|  | **Build a fast and easy-to-use website for ordering daily products.**  Main Pages:  \* **Home:** Show featured products, categories (e.g., Men, Women, Kids)  \* **Product Listing:** Display products filtered by categories.  \* **Product Details**: Provide details like price, availability.  \* **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. |

## 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. |

## Backend (Sanity CMS)

|  |
| --- |
| **Objective:**  Manage product data, categories, and orders efficiently.  Tasks:  \* **Product Data:**  Add fields like name, price, stock, image, category.  \* **Categories**:  Group products into categories like  Men, Women, or Kids.  \* **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 |

## API Endpoints

|  |  |
| --- | --- |
|  | **Endpoints:**  \* **/products:**  \* Method: GET  \* Purpose: Get all available 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. |

## 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.  \* **APls:** Connect the frontend and backend to share data like product info categories, and orders. |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |