Hackathon

Day 2

PLANING THE TECHNICAL FOUNDATION

FRONT-END (UI)

Overview:

Our e-commerce platform harnesses the power of Next.js, a state-of-the-art front-end framework renowned for its unparalleled speed and performance. By integrating advanced libraries such as Tailwind CSS, , and implementing industry-leading responsive design techniques, we deliver a sophisticated, high-performing user interface that sets a new standard for seamless and engaging online experiences.

KEY FEATURES OF THE USER INTERFACE

1

Intuitive and user-friendly browsing experience for all product categories.

2

Fully responsive layouts optimized for both mobile and desktop devices.

3

Clean and aesthetically pleasing design tailored to showcase furniture collections.

ESSENTIAL PAGES

1. Homepage: Highlight featured furniture collections and trending designs.
2. Product Listing: Easy navigation with filtering and sorting options.
3. Product Details: Detailed product descriptions, images, and specifications.
4. Shopping Cart: Simple and efficient cart management.
   Order Confirmation: Confirmation and tracking information for completed purchases.

SANITY CMS: THE BACKBONE OF OUR DYNAMIC E-COMMERCE PLATFORM

Sanity CMS offers a powerful, real-time content management system designed for modern e-commerce needs. It streamlines product data, customer records, and order management while serving as the store's primary database. Key features include real-time stock updates, seamless order processing, and instant content synchronization. With customizable schemas, Sanity adapts perfectly to unique business requirements, ensuring flexibility and efficiency.

ENHANCED API INTEGRATIONS

Shipping Management

Optimize shipment tracking with real-time updates, enabling precise monitoring across diverse zones to ensure timely deliveries.

Secure Payment Solutions

Facilitate seamless and secure transactions with comprehensive support for multiple payment methods, including digital wallets (Stripe, EasyPaisa, JazzCash) and Cash on Delivery.

Smart Notification Systems

Leverage cutting-edge APIs like Twilio and Firebase to provide instant SMS and email updates, keeping customers informed at every stage of their order journey.

KEY API SPECIFICATIONS

Product Management API

Endpoint: /products
Method: GET
Purpose: Retrieve a list of all available products in the Sanity CMS database.

Example:

```
[{"productId": 101,"productName":
"Laptop","productPrice": 1200},
{"productId": 102,"productName":
"Smartphone","productPrice":  800}]
```

Order Management API

Endpoint: /products
Method: GET
Purpose: Retrieve a list of all available products in the Sanity CMS database.

Example:

```
[{"productId": 101,"productName":
"Laptop","productPrice": 1200},
{"productId": 102,"productName":
"Smartphone","productPrice":  800}]
```

ADVANCED SHIPMENT TRACKING API

Endpoint: /shipment

HTTP Method: GET

Purpose: Efficiently monitor the real-time status of orders via a cutting-edge third-party shipment tracking system, delivering precise updates to enhance customer satisfaction and logistics management.

Example:

```
{"shipment_id": 832,"order_id": 725,"status": "In
Transit","expected_delivery_date":   "17-01-2025"}
```

SCHEMAS

```
import { createClient } from '@sanity/client';

// Sanity Client Configuration
const client = createClient({
projectId: process.env.FURNITURE_PROJECT_ID,
dataset: process.env.FURNITURE_DATASET,
useCdn: true,
apiVersion:  '2023-01-01',
});

// Product Schema
const productSchema = {
name: 'product',
type: 'document',
fields: [
{ name: 'name', type: 'string', title: 'Product Name' }, // Example: Sofa, Table
{ name: 'price', type: 'number', title: 'Price' }, // Example: 299.99
{ name: 'category', type: 'string', title: 'Category' }, // Example: Furniture, Decor
],
};

// Export Schemas
export const schemas = [productSchema];

// Fetch Function Example
export async function fetchProducts() {
try {
const products = await client.fetch('*[_type == "product"]');
console.log('Fetched products:', products);
return products;
} catch (error) {
console.error('Error fetching products:', error);
}
}
```