📅 Day 2 - Planning the 🛠️ Technical Foundation for Q-Commerce 🍔 Website

📋 Task Overview

On Day 2, the primary focus was to transition from 💼 business-oriented planning to 🖥️ technical preparation for building the Q-commerce website. The goal was to create a high-level technical plan, including 🗺️ system architecture, 🔄 workflows, and 🌐 API requirements, which will act as the blueprint for further development. Below is the detailed documentation of the work completed.

🎯 Goals Achieved on Day 2

1️⃣. Defining Technical Requirements

The technical requirements were derived from the 🎯 business goals defined earlier. The key areas focused on were:

🖥️ **Frontend Requirements:**

- Developed a user-friendly 🧑‍💻 interface for browsing fast food 🍟 products.
- Ensured 📱 responsiveness for mobile 📱 and desktop 🖥️ users.
- Created essential pages:
  - 🏡 Home
  - 🛒 Product Listing
  - 📃 Product Details
  - 🛍️ Cart
  - 💳 Checkout
  - ✅ Order Confirmation

💾 **Backend Requirements:**

- Utilized **Sanity CMS** to manage:

  - 🍔 Product data
  - 👤 Customer details
  - 📝 Order records
- Designed schemas in Sanity CMS to align with the business goals.

🌐 **Third-Party API Integrations:**

- Integrated APIs for:
  - 🚚 Shipment tracking
  - 💳 Payment gateways

2️⃣. System Architecture Design

A 🖼️ system architecture diagram was created to visualize how different components interact within the system. Below is the high-level structure:

🏗️ **Architecture:**

- **Frontend**: Built with Next.js ⚛️
- **CMS**: Sanity CMS to store and manage 🗂️ product and order data
- **APIs**:
  - 🚚 Shipment Tracking API
  - 💳 Payment Gateway API

🔄 **Workflow Example:**

1. A user visits the marketplace frontend to browse products 🛒.
2. Frontend requests product listings from the **Sanity CMS API**.
3. When an order is placed:
   - 📝 Order details are sent to Sanity CMS via an API request.
   - 🚚 Shipment details are fetched using the Shipment Tracking API.
   - 💳 Payment is processed through a secure Payment Gateway API.

3️⃣. API Requirements Planning

Detailed API requirements were outlined for smooth integration between the frontend and backend. Below are the key endpoints defined:

🌐 **API Endpoints:**

1. 📦 **Fetch Products:**

○ Endpoint: /products

○ Method: GET

○ Description: Retrieves all product details 🛍️.

Example Response:

```
 {
 "id": 1,
 "name": "🍔 Burger",
 "price": 200,
 "stock": 50
}
```

○

1. 🛍️ **Place Order:**

○ Endpoint: /orders

○ Method: POST

○ Description: Records a new order in Sanity CMS.

Payload Example:

```
 {
 "customerName": "👤 John Doe",
 "products": [
 { "id": 1, "quantity": 2 }

 ],

 "total": 400,

 "paymentStatus": "✅ Paid"

}
```

○

1. 🚚 **Track Shipment:**

- Endpoint: /shipment
- Method: GET
- Description: Fetches the current status of an order.

Example Response:

```
 {

 "orderId": 123,

 "status": "📦 In Transit",

 "ETA": "⏳ 15 mins"

}
```

4️⃣. Sanity CMS Schema Design

Sanity CMS schemas were created to handle key data entities for the Q-commerce 🍔 website. Below is an example schema for products:

```
export default {

 name: 'product',

 type: 'document',

 fields: [

 { name: 'name', type: 'string', title: '🍔 Product Name' },
 { name: 'price', type: 'number', title: '💲 Price' },
 { name: 'stock', type: 'number', title: '📦 Stock Level' },
 { name: 'image', type: 'image', title: '🖼️ Product Image' }
 ]
};
```

5️⃣. 🤝 **Collaboration and Refinement**

To ensure the quality and accuracy of the technical plan:

- Conducted a peer review session to refine the 🖼️ system architecture and 🌐 API endpoints.

- Validated the schema design against the 🎯 business goals.

🏁 Final Outcome for Day 2

By the end of Day 2, the following deliverables were completed:

1. A detailed 🛠️ technical plan aligned with the 🎯 business goals.
2. High-level 🖼️ system architecture diagram showing component interactions.
3. Well-documented 🌐 API requirements for product, order, and shipment management.
4. **Sanity CMS** schemas for managing core data entities.

This foundation ensures a seamless transition to Day 3, where 🌐 API integration and 📤 data migration will be the primary focus.

📤 Submission Details

- Document Title: **Day 2 - Planning the 🛠️ Technical Foundation for Q-Commerce 🍔 Website**

- Prepared by: **Riaz Hussain**

- Date: **15 January 2025**