Day 2: Planning the Technical Foundation

1. Overview

This document outlines the technical foundation for the project, integrating the key components depicted in the flowchart. The system architecture comprises a frontend developed with Next.js, a backend powered by Sanity Headless CMS, and various APIs for seamless functionality, including a Product API, Third-Party API, and a Payments Gateway.

2. System Architecture

Frontend Development

- Developed using **Next.js** for a modern, scalable, and efficient web application.
- Manages the main user interface and routes for different pages.
- Fetches and displays data from APIs.

Backend and Content Management

- Uses **Sanity Headless CMS** to manage dynamic content.
- Includes schema types such as:
  - **Chef.ts** (Handles chef-related data)
  - **Food.ts** (Handles food-related data)
- Provides structured content to the frontend.

APIs and Integrations

- **Product API**: Supplies product-related data to the frontend.
- **Third-Party API**: Facilitates communication with external services, including payment processors.
- **Payments Gateway**: Manages secure transactions for e-commerce functionality.

3. Page Structure

The application consists of the following main pages:

- **Home**: The landing page displaying an overview of available products and services.
- **About**: Information about the business and its mission.
- **Contact**: A form for users to reach out for inquiries or support.
- **Menu**: A detailed list of available food items with descriptions and prices.
- **Shop**: E-commerce functionality for purchasing products.
- **Pages**: Additional structured pages for content.

- **Blog**: Articles and updates relevant to the business.

4. Data Flow and Interaction

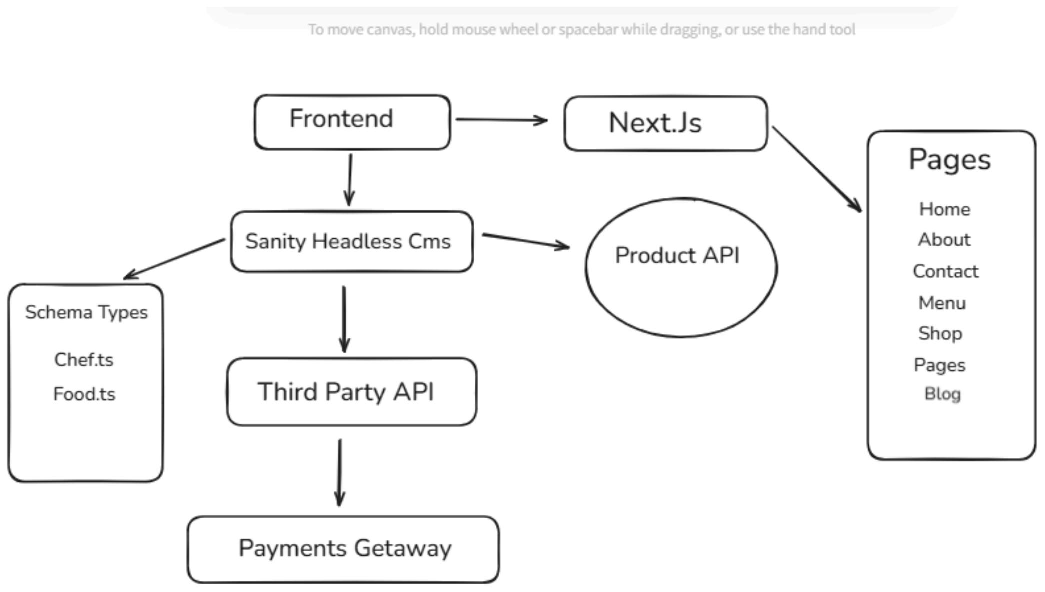
1. The **Frontend (Next.js)** fetches content from **Sanity Headless CMS** and **Product API**.
2. The **Sanity Headless CMS** structures and provides the content using defined schema types.
3. The **Third-Party API** enables additional functionality such as external integrations.
4. The **Payments Gateway** ensures secure transactions for purchasing items.

5. Security and Performance Considerations

- Implement authentication and authorization for sensitive operations.
- Optimize API calls to reduce load times and improve user experience.
- Ensure secure data handling and encryption for transactions.
- Enable caching strategies to enhance performance.

6. Conclusion

This document establishes the foundation for the project's technical implementation. The integration of **Next.js, Sanity CMS, APIs, and a secure Payments Gateway** ensures a structured, efficient, and scalable solution. Future enhancements can be made based on business needs and user feedback.


To move canvas, hold mouse wheel or spacebar while dragging, or use the hand tool
Frontend
Next.Js
Pages
Home
About
Contact
Menu
Shop
Pages
Blog
Sanity Headless Cms
Product API
Schema Types
Chef.ts
Food.ts
Third Party API
Payments Getaway