

Technical documents

System Architecture Overview

The platform is designed to facilitate real-time food ordering, delivery tracking, and inventory updates. It interacts

Frontend: Built using Next.js for a seamless user experience.

Backend: A Node.js server handling APIs, integrated with Sanity CMS for content management and inventory co

Third-party APIs: Integrated for payment gateways and delivery tracking services.

Components Roles

Frontend: Manages user interaction and renders product details fetched via APIs.

Backend: Processes requests, updates inventory, and handles payment and delivery status.

Sanity CMS: Central hub for product and inventory management.

Third-party APIs: Ensures secure transactions and real-time tracking.

Workflows

Workflow: Placing an Order

User selects products and adds them to the cart.

API /cart/add updates the cart database.

User proceeds to checkout, confirming the order and payment.

API /orders/create creates a new order entry in the database and triggers payment processing.

Order confirmation screen is displayed.

Workflow: Real-time Delivery Tracking

API /express-delivery-status fetches the current delivery status from the third-party provider.

Updates are pushed to the frontend in real-time via WebSocket or polling.

Flow: Real-time Delivery Tracking

Express-delivery-status fetches the current delivery status from the third-party provider.

Data is pushed to the frontend in real-time via WebSocket or polling.

API Schema

```
default {  
  product,  
  document",  
}
```

```
name: "name", type: "string", title: "Name"},
```

```
image: "image", type: "string", title: "Image"},
```

```
name: "description", type: "text", title: "Description"},
```

```
name: "price", type: "number", title: "Price"}
```