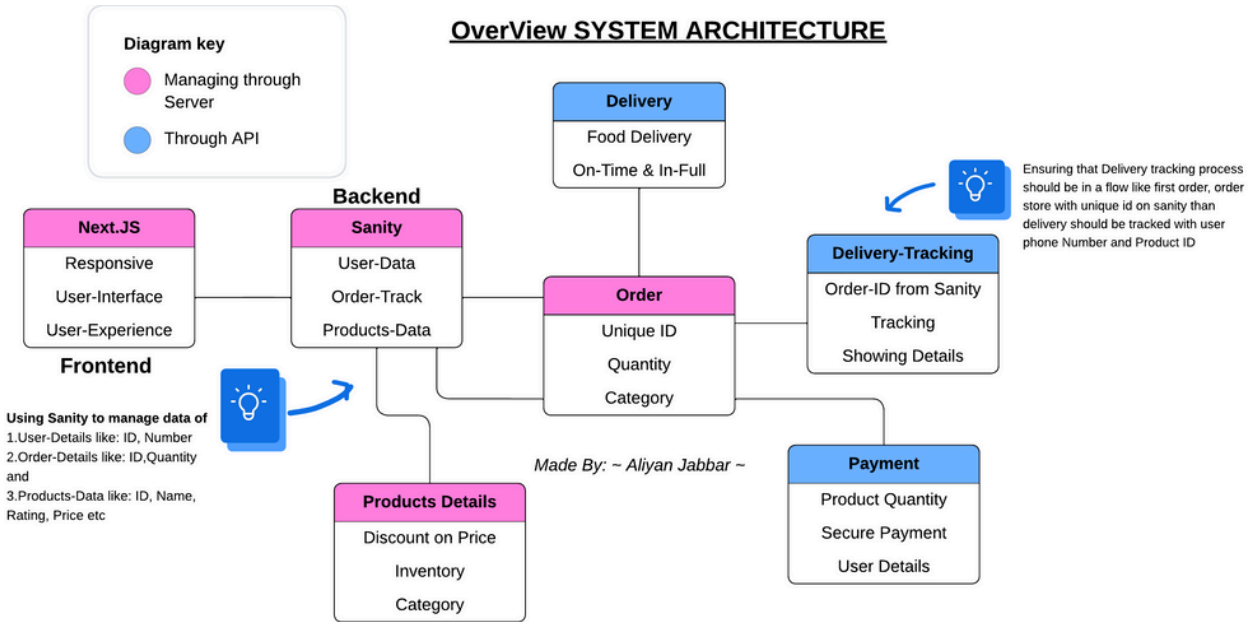
Technical Documentation: Q-Commerce System

Overview of System Architecture

The system architecture integrates Next.js for frontend operations, Sanity for backend data management, and APIs for order, delivery, and payment processing. The system ensures food delivery efficiency with real-time tracking, secure payment, and data-driven workflows.



Key Features

1. **Frontend (Next.js):**

- Responsive UI for customers to place orders.

- Focus on user experience and interface.

2. **Backend (Sanity):**

- Data Management:

- User Data: Stores customer details (ID, contact info).

- Order Data: Includes unique IDs, quantities, and categories.

- Product Data: Details such as name, price, rating, and inventory.

3. **Delivery Tracking:**

- Tracks orders via unique IDs.

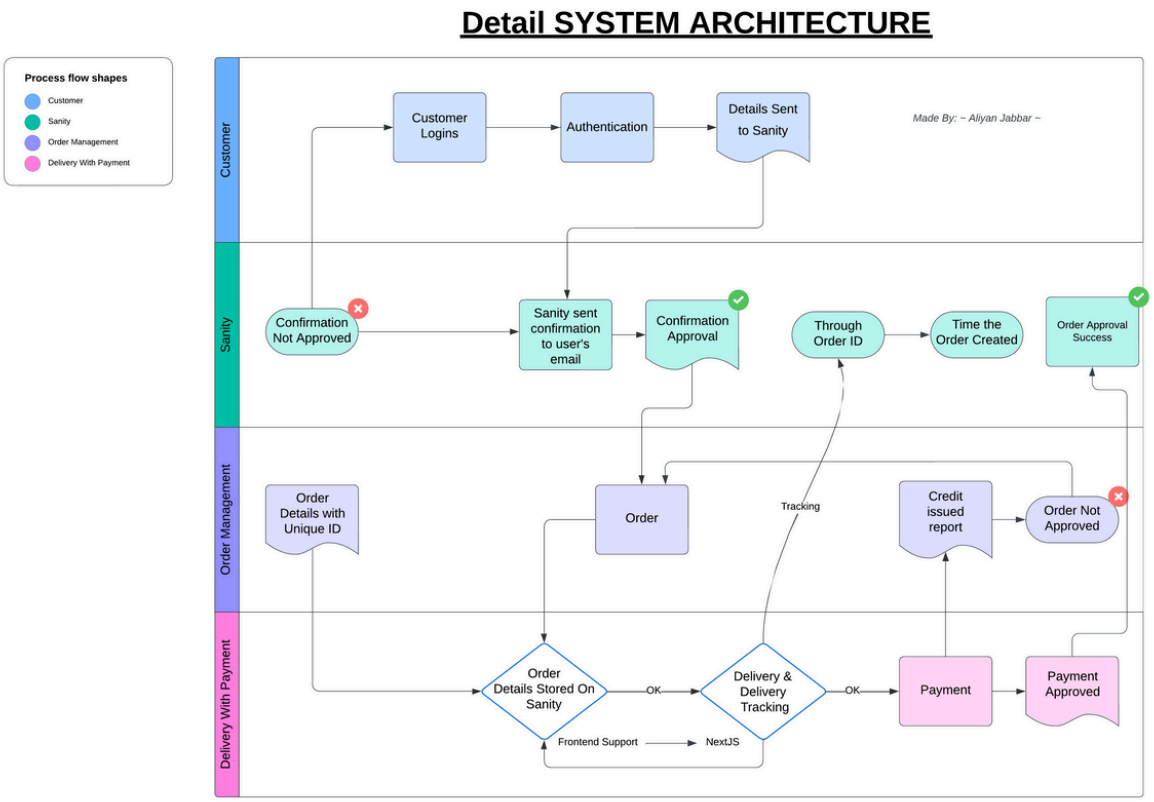
- Ensures seamless delivery updates based on order and product details.

4. **Payment System:**

- Manages secure payments.

- Links order details with payment status.

Detailed System Architecture



1. **User Authentication:**

- Customers log in and authenticate.

- Details are sent to Sanity for validation.

- Confirmation email sent to the customer upon approval.

2. **Order Creation:**

- Approved orders are assigned unique IDs.

- Order details are stored on Sanity.

3. **Delivery Tracking:**

- Links order and product details to track real-time progress.

- Customers receive delivery updates through APIs.

4. **Payment Processing:**

- Handles secure transactions.

- Sends confirmation on successful payment approval.

API Endpoints

1. **User Authentication**

- Endpoint: /authenticate

- Method: POST

- Description: Authenticate users and create sessions.

- Response Example: {

"userId": "12345",

"status": "Authenticated",

"token": "abcdef"

}

2. **Fetch Product Details**

- Endpoint: /products

- Method: GET

- Description: Retrieve all product details from Sanity.

- Response Example: {

"id": 1,

"name": "Product A",

"price": 100,

"stock": 25,

"rating": 4.5

}

3. **Create Order**

- Endpoint: /orders

- Method: POST

- Description: Create new orders and store them in Sanity.

- Response Example: {

"orderId": "67890",

"status": "Success",

"details": {

"quantity": 2,

"category": "Food"

}

}

4. **Track Delivery** - Endpoint: /delivery-tracking

- Method: GET

- Description: Fetch real-time delivery tracking details.

- Response Example: {

"orderId": "67890",

"status": "In Transit",

"ETA": "15 mins"

}

5. **Payment Processing**

- Endpoint: /payments

- Method: POST

- Description: Process secure payments for an order.

- Response Example: {

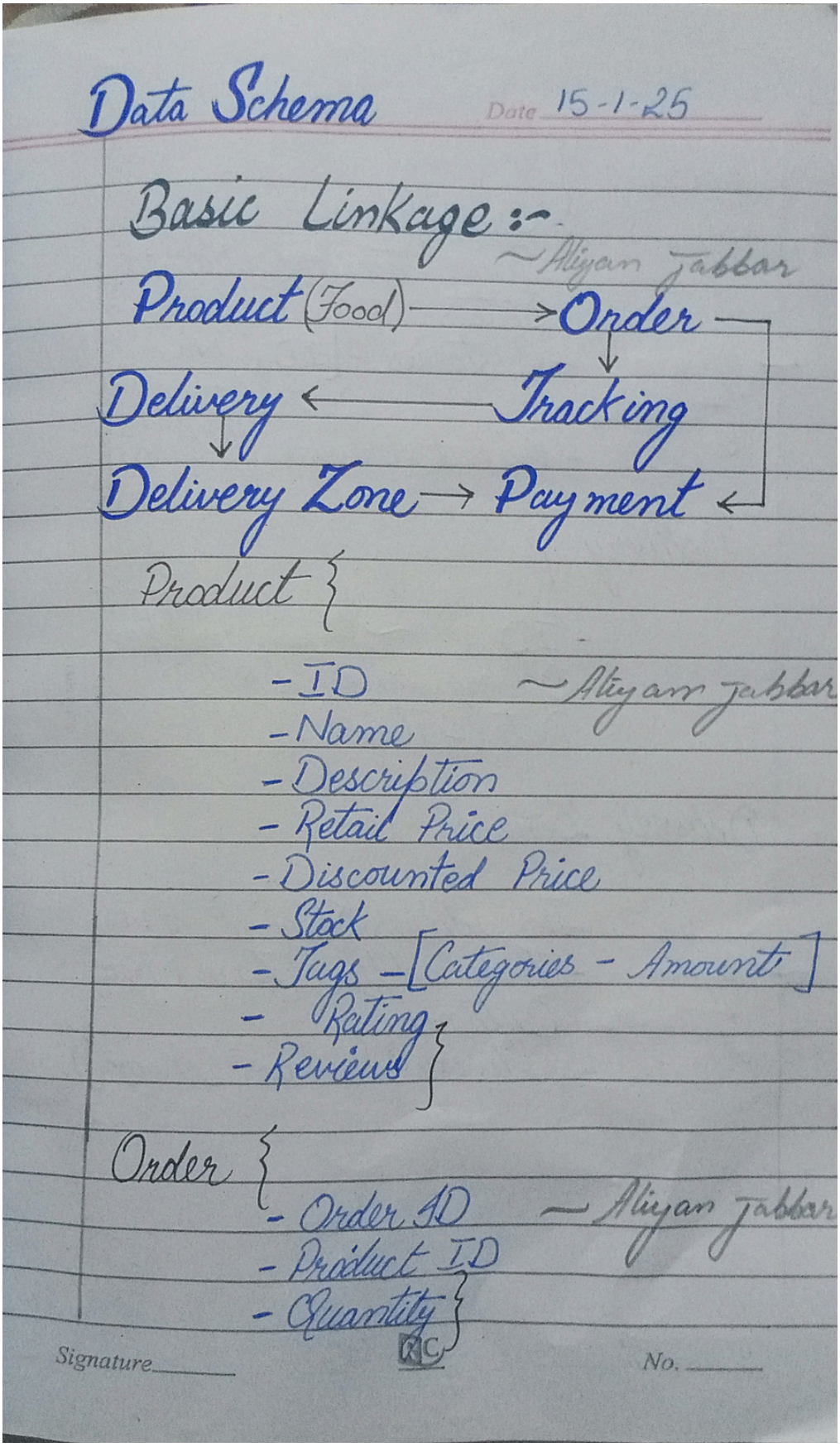
"paymentId": "abc123",

"status": "Success",

"orderId": "67890"

}

Sanity Schema



Date 15-1-25

Tracking {
- Order (Previous Schema)
- Time the food Ordered
- Status
- Customer - [ID, Name, Contact Info]
- Product (Previous Schema) }

~Aliyan Jabbar

Delivery {
- ID ~Aliyan Jabbar
- Delivery driver details
- Date - Time }

Delivery Zone {
- To address {Name, H.No}
- From address {Zone, Area}
- Distance
- Delivery {Previous Schema} }

~Aliyan Jabbar

Signature

Signature