

Marketplace Technical Foundation - Restaurant Platform

1. System Architecture Overview

Diagram:

The system architecture is designed to support a seamless user experience for a restaurant website and online food delivery platform. Below are the main components and their interactions:

- **Frontend (Next.js):** Handles user interface, interactions, and API calls.
- **Sanity CMS:** Manages backend data for products, orders, and customers.
- **Product Data API:** Fetches product details from the CMS.
- **Order API:** Saves and manages order details.
- **Payment Gateway:** Processes secure online payments.
- **Shipment API:** Tracks delivery status and updates.

Data Flow:

1. A user browses the menu on the frontend.
2. The frontend fetches product details via the Product Data API (connected to Sanity CMS).
3. Orders are placed, and details are saved in the CMS via the Order API.
4. Payments are processed through the Payment Gateway.
5. Delivery status is updated and displayed to users via the Shipment API.

2. Key Workflows

User Registration:

1. User signs up on the platform.
2. Details are saved in Sanity CMS.
3. A confirmation email is sent to the user.

Product Browsing:

1. User navigates to the menu.
2. Product categories and details are fetched from the Product Data API.
3. Products are dynamically displayed on the frontend.

Order Placement:

1. User adds items to the cart.
2. User proceeds to checkout and submits order details.
3. Order is saved in Sanity CMS via the Order API.
4. Payment details are processed, and confirmation is sent to the user.

Shipment Tracking:

1. Order status is updated via the Shipment API.
2. Users can track the real-time status of their order on the platform.

3. API Endpoints

Products API:

- **Endpoint:** /products
- **Method:** GET
- **Description:** Fetch all available products.
- **Response Example:**
- {
- "id": 1,
- "name": "BBQ Platter",
- "price": 15.99,
- "stock": 10
- }

Order API:

- **Endpoint:** /orders
- **Method:** POST
- **Description:** Create a new order.
- **Payload Example:**
- {
- "customerId": 123,
- "items": [- {"id": 1, "quantity": 2}
-],
- "total": 31.98
- }
- **Response Example:**
- {
- "orderId": 456,
- "status": "Pending"
- }

Shipment API:

- **Endpoint:** /shipment
- **Method:** GET
- **Description:** Track the delivery status of an order.
- **Response Example:**
 - {
 - "orderId": 456,
 - "status": "In Transit",
 - "ETA": "30 mins"
 - }

Delivery Zones API:

- **Endpoint:** /delivery-zones
- **Method:** GET
- **Description:** Fetch all delivery zones.
- **Response Example:**
 - {
 - "zoneId": 789,
 - "name": "Downtown",
 - "coverage": ["ZIP123", "ZIP124"]
 - }

4. Sanity Schema Example

Product Schema:

```
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: 'category', type: 'string', title: 'Category' },
    { name: 'tags', type: 'array', of: [{type: 'string'}], title: 'Tags' }
  ]
};
```

Order Schema:

```
export default {
  name: 'order',
  type: 'document',
  fields: [
    { name: 'customerId', type: 'string', title: 'Customer ID' },
    { name: 'items', type: 'array', of: [
      { type: 'object', fields: [
        { name: 'productId', type: 'string', title: 'Product ID' },

```

```
        { name: 'quantity', type: 'number', title: 'Quantity' }
      ]}
    ], title: 'Items' },
    { name: 'total', type: 'number', title: 'Total Amount' },
    { name: 'status', type: 'string', title: 'Order Status' }
  ]
};
```
