

API Requirements

1. Product Data APIs

The product-related API endpoints allow the frontend to retrieve information about the menu items and other related details. This ensures customers can browse available items, view descriptions, prices, and images.

Endpoint Name: `/products`

- **Method:** GET
- **Description:** Fetch all available products (menu items) from the Sanity CMS.
- **Payload:** No payload is required for this GET request.
- **Response Example:**

```
[
  {
    "id": 1,
    "name": "Spaghetti Carbonara",
    "price": 15.99,
    "stock": 50,
    "image": "https://example.com/images/spaghetti-carbonara.jpg",
    "description": "Classic Italian pasta with creamy sauce, pancetta, and parmesan."
  },
  {
    "id": 2,
    "name": "Margherita Pizza",
    "price": 12.99,
    "stock": 30,
    "image": "https://example.com/images/margherita-pizza.jpg",
    "description": "Traditional pizza with tomato sauce, mozzarella, and fresh basil."
  }
]
```

This endpoint will provide the frontend with a list of available menu items, including their names, prices, stock levels, and descriptions.

Endpoint Name: `/product/{id}`

- **Method:** GET
- **Description:** Fetch detailed information for a single product by its ID from the Sanity CMS.
- **Payload:** No payload is required.
- **Response Example:**

```
{
  "id": 1,
  "name": "Spaghetti Carbonara",
  "price": 15.99,
  "stock": 50,
  "image": "https://example.com/images/spaghetti-carbonara.jpg",
  "description": "Classic Italian pasta with creamy sauce, pancetta, and parmesan."
}
```

This API endpoint provides more detailed information for a specific product, including its description, price, and image.

2. Order Management APIs

The order management APIs are responsible for processing customer orders. They allow the system to record orders, update order statuses, and track payments.

Endpoint Name: `/orders`

- **Method:** POST
- **Description:** Create a new order in the Sanity CMS and initiate the checkout process.

- **Payload:**

```
{
  "customerId": 123,
  "items": [
    {
      "productId": 1,
      "quantity": 2
    },
    {
      "productId": 2,
      "quantity": 1
    }
  ],
  "totalPrice": 44.97,
  "paymentStatus": "Pending",
  "shippingAddress": {
    "street": "123 Main St",
    "city": "Cityville",
    "state": "Stateville",
    "zip": "12345",
    "country": "Country"
  },
  "contact": {
    "phone": "+123456789",
    "email": "customer@example.com"
  }
}
```

- **Response Example:**

```
{
  "orderId": 101,
  "status": "Success",
  "message": "Order has been successfully created."
}
```

This endpoint is used for creating a new order in the system. The payload contains details such as the customer ID, items ordered, total price, and shipping information. Once the order is successfully created, the system responds with a confirmation.

Endpoint Name: `/order/{id}`

- **Method:** GET
- **Description:** Retrieve the details of an existing order by order ID.
- **Payload:** No payload is required.
- **Response Example:**

```
{
  "orderId": 101,
  "customerId": 123,
  "items": [
    {
      "productId": 1,
      "quantity": 2,
      "price": 15.99
    },
    {
      "productId": 2,
      "quantity": 1,
      "price": 12.99
    }
  ],
  "totalPrice": 44.97,
  "shippingAddress": {
    "street": "123 Main St",
    "city": "Cityville",
    "state": "Stateville",
    "zip": "12345",
    "country": "Country"
  },
  "paymentStatus": "Pending",
}
```

```
    "orderStatus": "Processing"
  }
```

This API will return all the details for a specific order, including the items, total price, shipping details, and payment status.

3. Shipment Tracking APIs

Shipment tracking APIs help customers track the real-time status of their orders, including delivery updates and expected delivery times.

Endpoint Name: `/shipment/{orderId}`

- **Method:** GET
- **Description:** Track the shipment status of a specific order via third-party delivery services.
- **Payload:** No payload is required.
- **Response Example:**

```
{
  "orderId": 101,
  "status": "In Transit",
  "shipmentId": "XYZ123",
  "eta": "15 mins",
  "currentLocation": "Cityville Distribution Center",
  "expectedDeliveryTime": "2025-01-15T17:30:00Z"
}
```

This endpoint will allow customers to track the current status of their order, including the shipment ID, current location, expected delivery time, and the real-time status (e.g., "In Transit," "Delivered").

4. Payment Integration APIs

Payment APIs will securely process transactions and update the payment status once the payment is confirmed.

Endpoint Name: `/payment`

- **Method:** POST
- **Description:** Process a payment for an order using a secure third-party payment gateway.
- **Payload:**

```
{
  "orderId": 101,
  "paymentMethod": "CreditCard",
  "paymentDetails": {
    "cardNumber": "4111111111111111",
    "expiryDate": "12/25",
    "cvv": "123"
  },
  "amount": 44.97
}
```

- **Response Example:**

```
{
  "paymentId": 5001,
  "status": "Success",
  "message": "Payment has been successfully processed."
}
```

This API endpoint is responsible for processing the payment for an order. It sends the payment details to the third-party payment gateway for secure processing and returns the payment status once the transaction is completed.

5. Notification APIs

Notification APIs will send order updates and confirmations to customers via email or SMS.

Endpoint Name: `/send-notification`

- **Method:** POST
- **Description:** Send a notification to the customer about their order status (confirmation, shipment, or delivery update).
- **Payload:**

```
{
  "customerId": 123,
  "notificationType": "Order Confirmation",
  "message": "Your order has been successfully placed and is being processed.",
  "contact": {
    "email": "customer@example.com",
    "phone": "+123456789"
  }
}
```

- **Response Example:**

```
{
  "status": "Success",
  "message": "Notification has been sent successfully."
}
```

This API will send notifications to the customer through email or SMS. It allows various types of notifications, such as order confirmation, shipping updates, or delivery notifications.

These API endpoints are designed to provide the essential functionalities required for the restaurant's website, including product browsing, order management, shipment tracking, payment processing, and customer notifications. By implementing these APIs, the restaurant can create a seamless and efficient system for both customers and staff, ensuring a smooth experience from browsing the menu to receiving orders. Each API is designed to interact with the backend (Sanity CMS), third-party services, and the frontend, facilitating real-time updates and transaction processing.