Marketplace Technical Foundation - Food Ordering Website

1. Define Technical Requirements:

Frontend Requirements:

- User-friendly interface for browsing food items and chefs.
- Responsive design for mobile and desktop users.
- Essential pages:
  - Home
  - Food Listing
  - Food Details
  - Cart
  - Checkout
  - Order Confirmation
  - User Profile

Sanity CMS as Backend:

Sanity CMS will manage:

- Food Items (name, price, description, category, image, etc.).
- Chefs (name, experience, specialization, availability).
- Orders (user details, ordered items, status, total price).
- Customers (name, email, address, order history).

Third-Party APIs:

- Payment Gateway API (Stripe, PayPal) for secure transactions.
- Shipment Tracking API to track food deliveries.
- Authentication API (Clerk/Auth0) for login & signup.

2. Design System Architecture:

```
[Frontend (Next.js)]
    |
[Sanity CMS] ---------> [Food Data API]
    |
[Third-Party API] -----> [Shipment Tracking API]
    |
[Payment Gateway]
```

Workflow Example: Ordering Food:

1. User browses the food menu on the frontend.

2. Frontend requests food data from Sanity CMS via API.

3. User selects food, adds it to the cart, and proceeds to checkout.

4. Order details are sent to Sanity CMS, where it is recorded.

5. Payment Gateway API processes the transaction.

6. Shipment Tracking API provides real-time order status.

7. Order is updated in Sanity CMS and displayed to the user.

3. Plan API Requirements:

Endpoint	Method	Purpose	Response Example
/foods	GET	Fetch all available food items	{ "id": 1, "name": "Pizza", "price": 10 }
/Chefs	GET	Fetch list of chefs	{ "id": 2, "name": "John Doe", "specialty": "Italian" }
/Orders	POST	Place a new order	{ "orderId": 123, "status": "Processing" }
/orders/:id	GET	Get order details by ID	{ "orderId": 123, "items": [...], "status": "Delivered" }
/shipment/:id	GET	Get real-time tracking of food delivery	{ "orderId": 123, "ETA": "30 mins" }

4. Sanity Schema Design:

Food Schema

```
export default {
  name: 'food',
  type: 'document',
  title: 'Food Item',
  fields: [
    { name: 'name', type: 'string', title: 'Food Name' },
    { name: 'price', type: 'number', title: 'Price' },
    { name: 'description', type: 'text', title: 'Description' },
    { name: 'category', type: 'string', title: 'Category' },
    { name: 'image', type: 'image', title: 'Image' },
    { name: 'chef', type: 'reference', to: [{ type: 'chef' }], title: 'Prepared By' }
  ]
};
```

Chef Schema

```
export default {
 name: 'chef',
 type: 'document',
 title: 'Chef',
 fields: [
  { name: 'name', type: 'string', title: 'Name' },
  { name: 'experience', type: 'number', title: 'Years of Experience' },
  { name: 'specialty', type: 'string', title: 'Specialty Cuisine' },
  { name: 'image', type: 'image', title: 'Chef Image' }
 ]
};
```

Order Schema

```
export default {
 name: 'order',
 type: 'document',
 title: 'Order',
 fields: [
  { name: 'customer', type: 'reference', to: [{ type: 'customer' }], title: 'Customer' },
  { name: 'items', type: 'array', of: [{ type: 'reference', to: [{ type: 'food' }] }], title: 'Ordered Items' },
  { name: 'totalPrice', type: 'number', title: 'Total Price' },
  { name: 'status', type: 'string', title: 'Order Status', options: { list: ['Pending', 'Processing', 'Delivered'] } }
 ]
};
```

Customer Schema

```
export default {
 name: 'customer',
 type: 'document',
 title: 'Customer',
 fields: [
  { name: 'name', type: 'string', title: 'Full Name' },
  { name: 'email', type: 'string', title: 'Email Address' },
  { name: 'address', type: 'string', title: 'Delivery Address' },
  { name: 'orderHistory', type: 'array', of: [{ type: 'reference', to: [{ type: 'order' }] }], title: 'Order History' }
 ]
};
```