Hackathon 3: Day 2

Technical Requirements

Next js

1. For user friendly interface clear navigation, responsive design and interactive features.

2. Tailwind for responsive designs

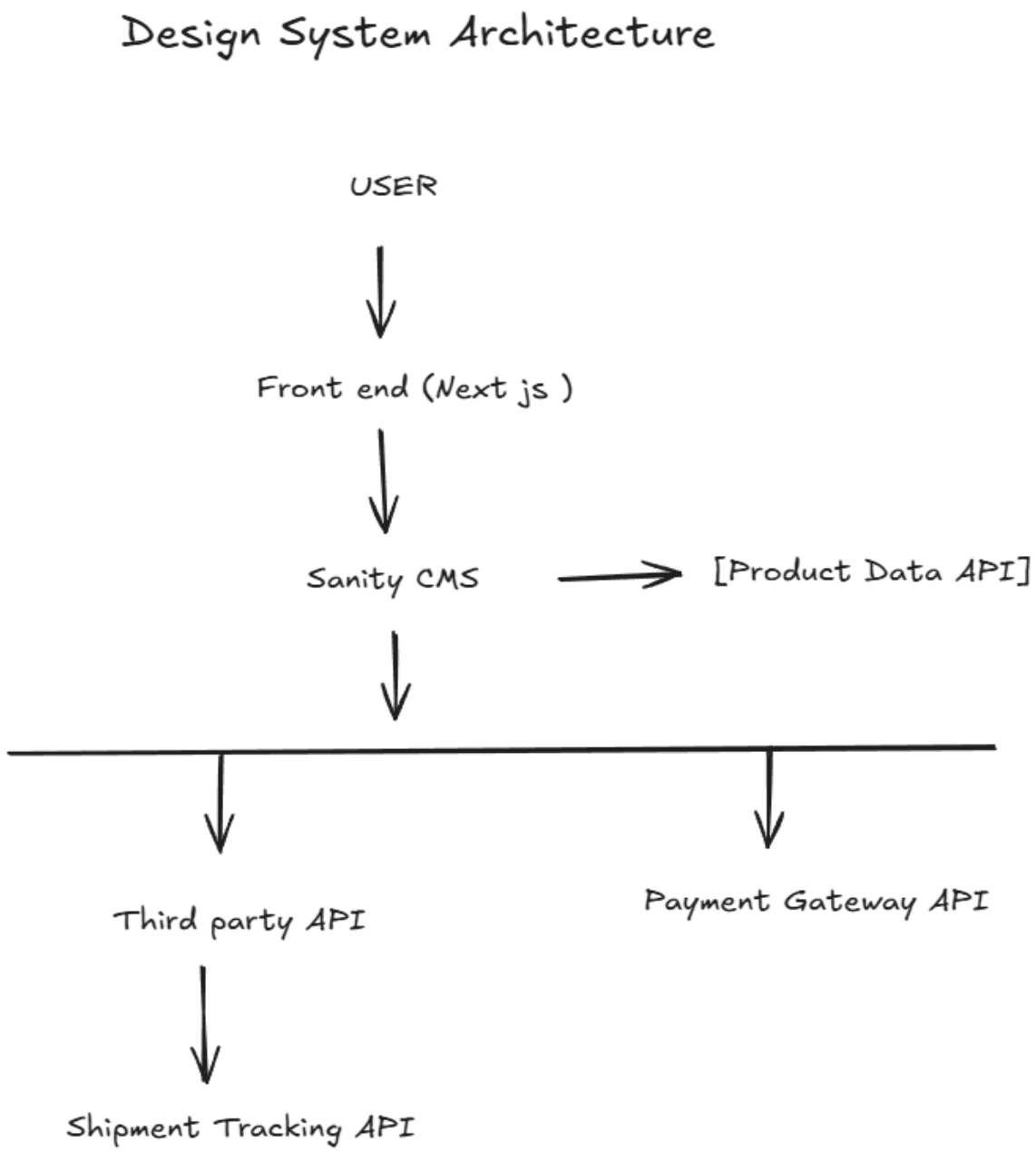
- **Home Page**: A welcoming page with featured dishes or categories.
- **Menu Page**: Displays all available dishes or meals.
- **Shop Page**: Showcases food-related products for purchase.
- **Shop Details Page**: Provides detailed information about a specific product.
- **Cart Page**: Lets users review and update selected items for purchase.
- **Checkout Page**: Collects shipping and payment details to complete orders.
- **Login Page**: Allows users to sign in to their accounts.
- **Signup Page**: Enables new users to create an account.

Sanity CMS

For this project, **Sanity CMS** will be used as a headless content management system to store and manage order details, payment statuses, and updates throughout the order lifecycle. It allows for flexible content structures with custom schemas for orders, customer details, and food items. Real-time updates will be fetched from Sanity to dynamically display order statuses on the front-end, ensuring a seamless user experience.

Third-Party API Integration:

- **Shipment Tracking**: Show users real-time order delivery status.
- **Payment Gateways**: Allow users to securely pay using credit cards, digital wallets, etc.
- **Additional Services**: For example, sending order confirmation emails or handling taxes.



Key Workflows:

1.User Places the Order:

- User clicks "Place Order" => Order details (items, quantity, price, address) sent to Sanity CMS => Order confirmation displayed on the front-end.

- **2.Payment Processing**:

User enters payment details => Payment gateway (e.g., Stripe/PayPal) processes payment => Confirmation sent back to user and saved in Sanity.

3.Order Confirmation:

Order confirmed in Sanity => Notification sent to the user (via email/SMS) => Order status updated to "Confirmed" on the front-end.

3.Order Preparation:

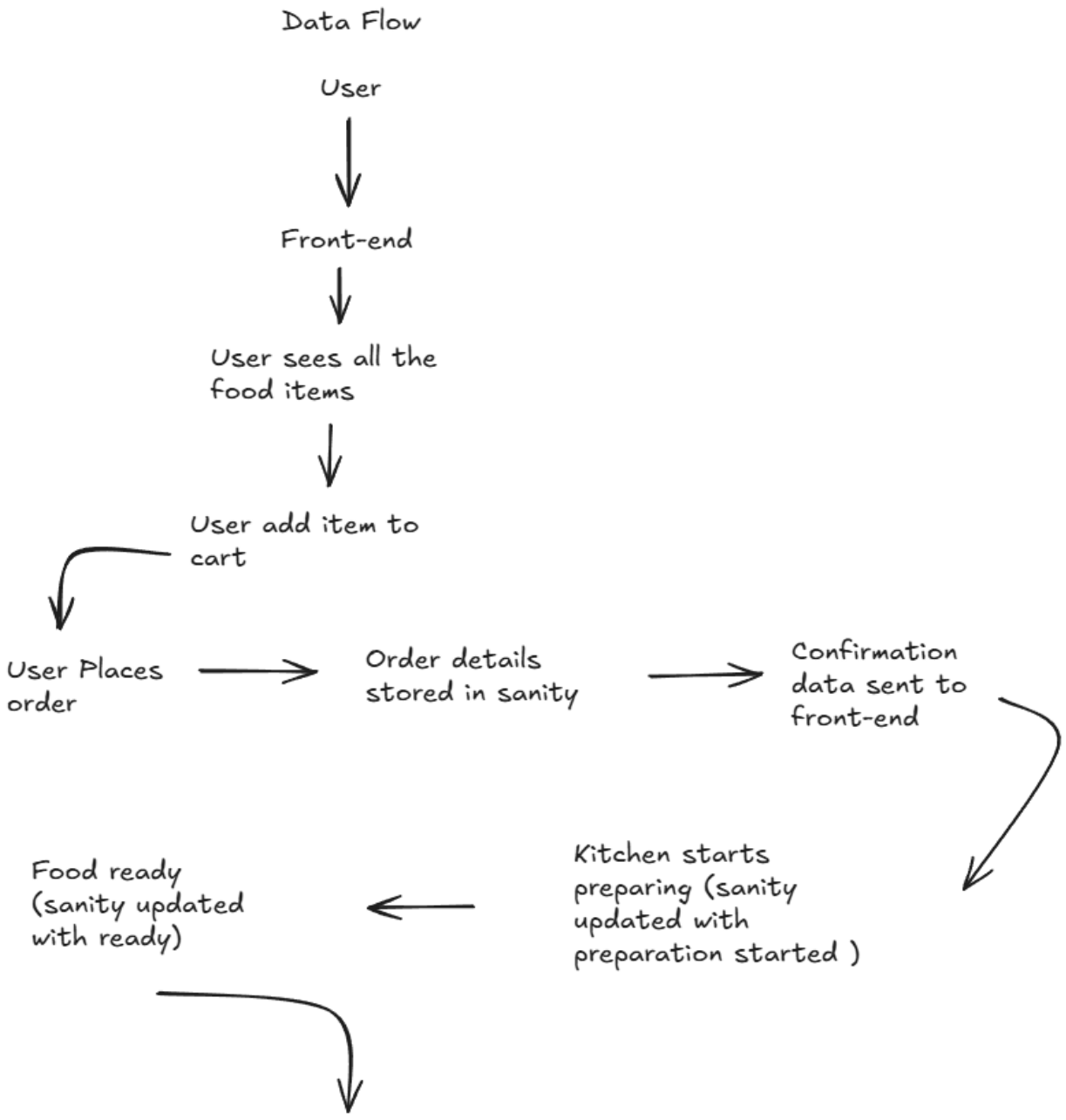
Order sent to the kitchen => Kitchen starts preparing the food => Preparation status updated in Sanity => User notified.

4.Food Ready for Delivery:

Food marked as "Ready" => Delivery partner notified => Tracking info (from shipment API) updated in real-time and displayed to the user.

5.Delivery and Completion:

Delivery partner picks up the food => Tracking API updates the user => Food delivered => Order status marked as "Delivered" in Sanity => User notified


Data Flow
User
Front-end
User sees all the food items
User add item to cart
User Places order
Order details stored in sanity
Confirmation data sent to front-end
Kitchen starts preparing (sanity updated with preparation started)
Food ready (sanity updated with ready)
Delivery partner picks up the food, and the status is updated in real-time via the shipment API.

API Requirements

	A	B	C
1	Api Endpoints	Method	Description
2	api/fooditems	Get	Fetch a list of food items available for browsing.
3	api/user	Post	Register a new user Account
4	api/order	Post	place an order from cart
5	api/confirm	Post	Saves order details,and send mail to user
6	api/delivery	Get	Fetch real time update on order
7	api/reviews	Get	Fetches all reviews for a specific food item
8			
9			

Schema Draft

```
export default {
  name: 'foodItem',
type: 'document',
  fields: [
    { name: 'name', type: 'string', title: 'Food Name' },
    { name: 'description', type: 'text', title: 'Description' },
    { name: 'price', type: 'number', title: 'Price' },
    { name: 'category', type: 'string', title: 'Category' },
    { name: 'availability', type: 'boolean', title: 'Availability' },
    { name: 'imageUrl', type: 'string', title: 'Image URL' },
```

```
  { name: 'createdAt', type: 'datetime', title: 'Created At' }
 ]
};


export default {
 name: 'cart',
 type: 'document',
 fields: [
  { name: 'userId', type: 'reference', to: [{ type: 'user' }], title: 'User' },
  { name: 'createdAt', type: 'datetime', title: 'Created At' },
  { name: 'updatedAt', type: 'datetime', title: 'Updated At' }
 ]
};



export default {
 name: 'order',
 type: 'document',
 fields: [
  { name: 'userId', type: 'reference', to: [{ type: 'user' }], title: 'User' },
  { name: 'cartId', type: 'reference', to: [{ type: 'cart' }], title: 'Cart' },
  { name: 'totalAmount', type: 'number', title: 'Total Amount' },
  { name: 'status', type: 'string', title: 'Status' },
  { name: 'deliveryAddress', type: 'string', title: 'Delivery Address' },
  { name: 'createdAt', type: 'datetime', title: 'Created At' }
 ]
};

export default {
 name: 'orderItem',
 type: 'document',
```

```
  fields: [
    { name: 'orderId', type: 'reference', to: [{ type: 'order' }], title: 'Order' },
    { name: 'itemId', type: 'reference', to: [{ type: 'foodItem' }], title: 'Food Item' },
    { name: 'quantity', type: 'number', title: 'Quantity' },
    { name: 'price', type: 'number', title: 'Price' }
  ]
};

export default {
  name: 'payment',
  type: 'document',
  fields: [
    { name: 'orderId', type: 'reference', to: [{ type: 'order' }], title: 'Order' },
    { name: 'paymentMethod', type: 'string', title: 'Payment Method' },
    { name: 'paymentStatus', type: 'string', title: 'Payment Status' },
    { name: 'amount', type: 'number', title: 'Amount' },
    { name: 'createdAt', type: 'datetime', title: 'Created At' }
  ]
};
export default {
name: 'shipment',
type: 'document',
  fields:
 { name: 'orderId', type: 'reference', to: [{ type: 'order' }], title: 'Order' },
    { name: 'status', type: 'string', title: 'Shipment Status' },
    { name: 'ETA', type: 'datetime', title: 'Estimated Time of Arrival' },
    { name: 'updatedAt', type: 'datetime', title: 'Last Updated' }
  ]
};

export default {
  name: 'review',
```

```
  type: 'document'
  fields: [
   {name: 'reviewId', type: 'string', title: 'Review ID' },
   {name: 'userId', type: 'reference', to: [{ type: 'user' }], title: 'User' },
   {name: 'foodItemId', type: 'reference', to: [{ type: 'foodItem' }], title: 'Food
Item' },
   {name: 'rating', type: 'number', title: 'Rating'},
   {name: 'comment', type: 'text', title: 'Comment' },
   {name: 'createdAt', type: 'datetime', title: 'Created At' }
  ]
};
```