

# HACKATHONE 3

## DAY 2 PLANNING THE TECHNICAL FOUNDATION

### 1. Purpose:

This e-commerce website focuses on selling electronics, especially kitchen gadgets, catering to customers seeking innovative and practical tools for their homes.

### 2. Target Audience:

Primary:

Homeowners, chefs, and cooking enthusiasts.

Age group: 25–55 years.

Individuals looking for modern solutions to simplify cooking and household tasks.

Secondary:

Gift buyers and online shoppers interested in trendy gadgets.

### 3. Market Need (Aim to Solve):

Problem:

Limited access to high-quality and unique kitchen gadgets in local marketplaces.

Inconvenience in finding reliable, affordable, and efficient products online.

Solution:

Offering a wide range of gadgets with detailed descriptions, competitive pricing, and seamless shopping.

Providing customer reviews, fast shipping, and a user-friendly experience to build trust and convenience.

#### **Sanity CMS as Backend:**

The planning focuses on creating a dynamic business infrastructure, addressing data schemas for managing products, customers, orders, shipments, and payments. By leveraging Sanity CMS for content management and Next.js for frontend development, ensures a seamless flow of data between the backend and frontend systems.

UI/UX Suggestions

## Frontened

### A. Homepage Design

**Hero Section:**

Display a banner showcasing the latest kitchen gadgets with an engaging headline like "Revolutionize Your Kitchen with Smart Gadgets!"

Add a prominent CTA (Call to Action) button like "Shop Now."

**Featured Products Section:**

Showcase best-sellers and new arrivals with high-quality images, short descriptions, and price tags.

Include a "Quick View" option for easy product exploration.

**Search bar & Categories:**

Add a sticky search bar for quick product searches.

Highlight product categories: Blenders, Coffee Machines, Air Fryers, etc.

**Trust Signals:**

Include customer testimonials, ratings, and trust badges (e.g., "Fast Shipping," "Secure Payments").

**B. Product Page Design**

High-quality product images with zoom functionality.

Detailed descriptions including features, materials, dimensions, and usage tips.

Add "Frequently Bought Together" recommendations to boost sales.

Display reviews with photos/videos from real customers.

**C. Checkout Flow****Cart Summary:**

Display a clear order summary with subtotal, tax, and shipping cost.

**Guest Checkout:**

Allow quick checkout without requiring account creation.

**Multiple Payment Options:**

Support credit cards, mobile wallets, and cash-on-delivery (if applicable).

**D. Mobile Optimization**

Ensure the design is responsive with thumb-friendly buttons and streamlined navigation.

Optimize image sizes for faster loading on mobile devices.

## **E. Additional Features**

**Wishlist:** Allow users to save products for future purchases.

**Blog Section:** Share tips like "How to Use Air Fryers for Healthy Cooking."

## **2. Tech Stack Insights**

### **A. Frontend**

#### **Framework:**

Use Next.js for its SEO-friendly structure and server-side rendering.

#### **Styling:**

Use Tailwind CSS for fast, component-based styling.

### **B. Backend**

#### **Sanity CMS:**

Use Sanity for dynamic product management and blog content.

Add Structured Content Models for scalability (e.g., categories, reviews).

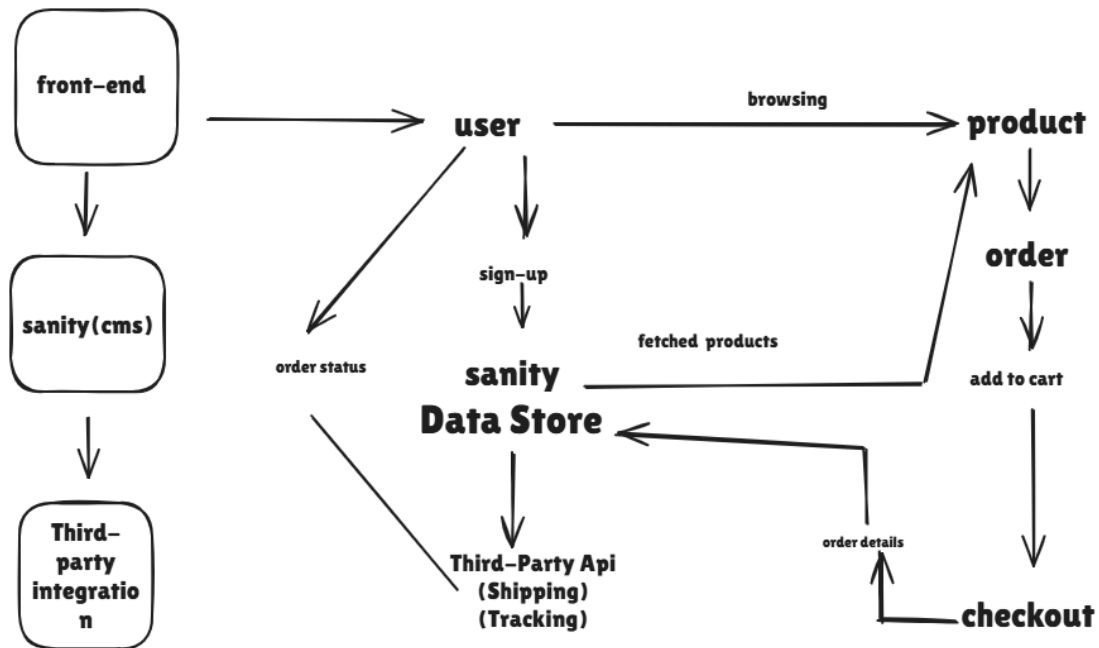
#### **API Integration:**

#### **Hosting**

#### **Vercel:**

Optimized for Next.js with fast deployments and CDN.

## **System Architecture**



## API Endpoints

Endpoints	Method	Description
/Products	Get	Fetches all products
/Products/id	Get	Fetches a specific product
Orders	Post	Creates a new order
Orders/id	Get	Fetches a specific order
/shipment/tracking/id	Get	Tracks shipment status

## API Requirements

### 1. User Management

Register User (POST/api/users/register):

- Registers a new user in the system.
- Response: Frontend sends data → API processes request → Sanity CMS stores user info →

Response sent back to Frontend.

Login User (POST/api/users/login):

- Authenticates user credentials and provides a token for secure access.
- Response: Frontend sends login data → API validates credentials → Authentication service returns token → Token sent back to Frontend.

## **2. Product Management**

Fetch All Products (GET/api/products):

- Retrieves the complete list of products for browsing.
- Response: Frontend requests product list → API fetches from Sanity CMS → Product data sent back to Frontend.

Fetch Product by ID (GET/api/products/:id):

- Gets details of a specific product by its unique ID.
- Response: Frontend requests product details → API retrieves data from Sanity CMS → Product details sent back to Frontend.

## **3. Cart Management**

Add to Cart (POST/api/cart):

- Adds a selected product to the user's shopping cart.
- Response: Frontend sends cart data → API updates cart in Sanity CMS → Confirmation sent back to Frontend.

Get Cart Details (GET/api/cart/:userId):

- Fetches the list of items in the cart for a specific user.
- Response: Frontend requests cart details → API fetches cart data from Sanity CMS → Cart

details sent back to Frontend.

#### 4. Order Management

Place Order (POST/api/orders):

- Submits an order for processing after cart review.
- Response: Frontend sends order data → API processes the order in Sanity CMS → Order confirmation sent back to Frontend.

Fetch Order Details (GET/api/orders/:orderId):

- Retrieves details of a placed order using the order ID.
- Response: Frontend requests order details → API retrieves data from Sanity CMS → Order details sent back to Frontend.

#### 5. Shipment Management

Track Shipment (GET/api/shipment/:shipmentId):

- Provides shipment tracking information for an order.
- Response: Frontend sends shipment ID → API retrieves tracking data from Sanity CMS →

Tracking details sent back to Frontend.

#### 6. Payment Management

Process Payment (POST/api/payment):

## SCHEMAS

#### PRODUCT:

```
name: 'product',  
type: 'document',  
title: 'Product',  
fields: [  
  { name: 'productId', type: 'number', title: 'Product ID' },  
  { name: 'productName', type: 'string', title: 'Product Name' },
```

```
    { name: 'slug', type: 'slug', title: 'Slug', options: { source: 'productName', maxLength: 96 } },
    { name: 'price', type: 'number', title: 'Price' },
    { name: 'stock', type: 'number', title: 'Stock' },
    { name: 'images', type: 'array', title: 'Images', of: [{ type: 'image' }] },
    { name: 'category', type: 'string', title: 'Category' },
  ],
};
```

## **USER**

```
name: 'customer',
type: 'document',
title: 'Customer',
fields: [
  { name: 'customerId', type: 'number', title: 'Customer ID' },
  { name: 'customerName', type: 'string', title: 'Customer Name' },
  { name: 'customerEmail', type: 'string', title: 'Customer Email' },
  { name: 'phoneNumber', type: 'string', title: 'Phone Number' },
  { name: 'address', type: 'string', title: 'Address' },
],
};
```

## **ORDER**

```
name: 'order',
type: 'document',
title: 'Order',
fields: [
  { name: 'orderId', type: 'number', title: 'Order ID' },
  { name: 'customerId', type: 'number', title: 'Customer ID' },
  { name: 'orderDate', type: 'datetime', title: 'Order Date' },
```

```
{ name: 'totalAmount', type: 'number', title: 'Total Amount' },  
  
{  
  name: 'orderStatus',  
  type: 'string',  
  title: 'Order Status',  
  options: { list: [{ title: 'Pending', value: 'pending' }, { title: 'Received', value: 'received' }] }  
},  
],  
};
```

## **DELIVERY ZONE**

```
name: 'deliveryZone',  
type: 'document',  
title: 'Delivery Zone',  
fields: [  
  { name: 'zoneId', type: 'number', title: 'Zone ID' },  
  {  
    name: 'zoneName',  
    type: 'array',  
    title: 'Zone Name',  
    of: [{ type: 'string' }]  
  },  
  { name: 'coverageArea', type: 'string', title: 'Coverage Area' },  
],
```