

Technical Foundation Planning

Frontend Requirements

User Interface

- Design an intuitive and user-friendly interface to ensure seamless product browsing.
- Incorporate smooth navigation and interactive elements for an engaging user experience.

Responsive Design

- Ensure the platform is fully functional and optimized for both mobile and desktop devices.

Essential Pages

- **Home Page:** Highlight featured products or categories for easy access.
 - **About Page:** Provide detailed information about the business.
 - **Product Listing Page:** Display all available products with filtering and sorting options.
 - **Product Details Page:** Show product images, descriptions, prices, and include an "Add to Cart" button.
 - **Sign-Up/Login Page:** Allow users to create an account or log in. Only logged-in users can add products to the cart or place orders.
 - **Cart Page:** Display selected products with options to update quantities or remove items (for logged-in users).
 - **Checkout Page:** Collect customer payment and shipping details to complete the purchase.
-

Backend Requirements

Sanity CMS for Data Management

- **Product Data:** Manage product details like name, description, price, category, and images.
- **Customer Information:** Store and manage customer profiles, including name, contact details, address, and email.

- **Order Records:** Track order statuses, payment confirmations, and shipping details.

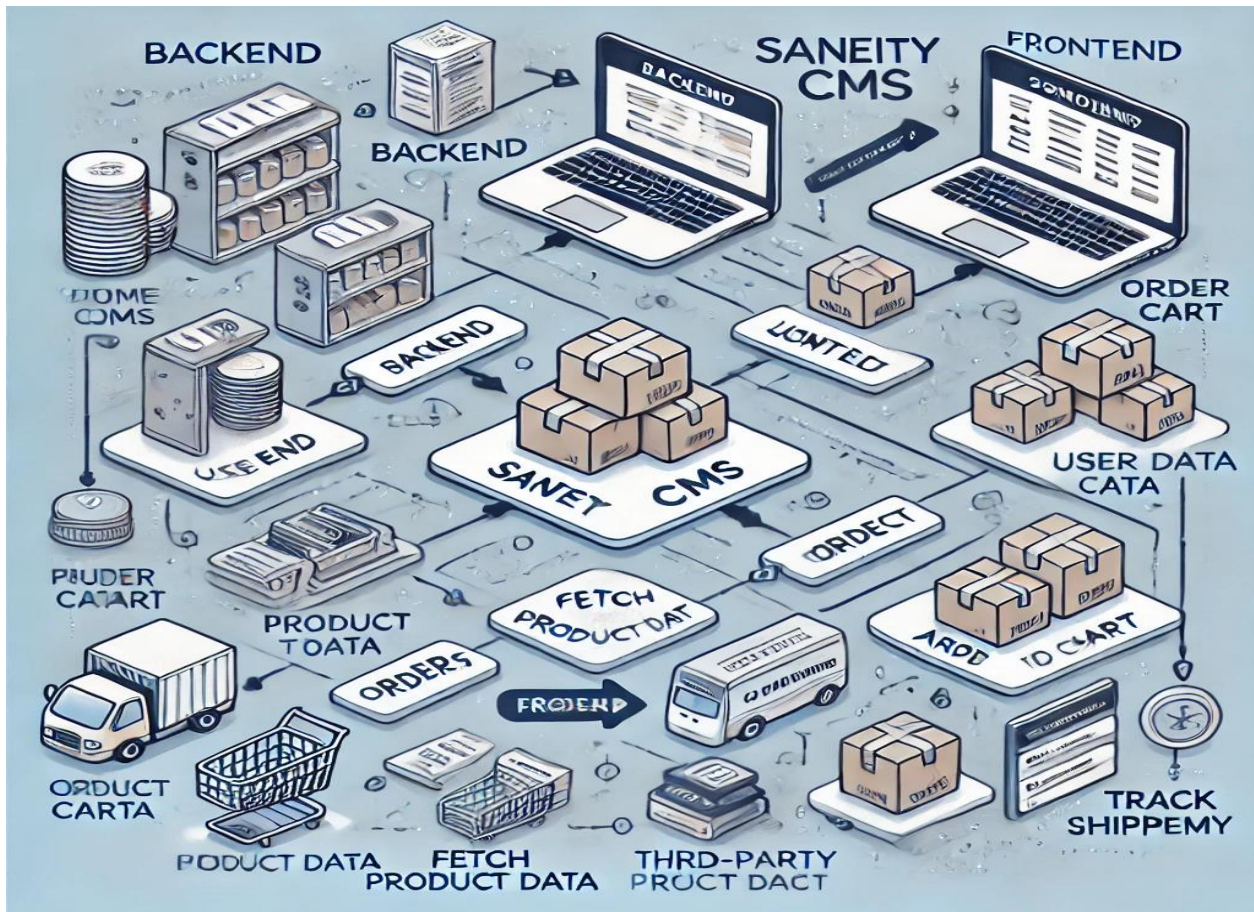
Schema Design

- Create schemas in Sanity to organize data for business needs, including products, orders, customers, and delivery zones.
-

Third-Party API Integration

- **Shipment Tracking API:** Allow customers to track the status and progress of their orders effortlessly.
- **Payment Gateway API:** Integrate secure and reliable payment gateways (e.g., Stripe, Easypaisa, JazzCash) for processing customer transactions

System Architecture Diagram



Workflows

i. User Registration

The user fills out the registration form, and the information is sent to Sanity CMS. A new user account is created, and a confirmation message or email is sent to the user. This ensures a smooth registration process and efficient account management.

ii. Product Browsing

The user navigates through the marketplace, and the frontend makes an API call to Sanity CMS to retrieve product data, including names, descriptions, prices, stock levels, and images. The data is dynamically displayed on the website, enabling users to explore products and make well-informed purchase decisions.

iii. Order Placement

The user adds items to their cart, and at checkout, the order details (product info, total cost, and user info) are sent to Sanity CMS through an API call. The order is recorded in Sanity CMS with a unique order ID, and the user is redirected to the payment gateway to complete the transaction. After successful payment, a confirmation is sent to the user, and the order status is updated in Sanity CMS. This ensures seamless order processing and secure payment handling.

iv. Shipment Tracking

The user checks their order status on the website, and the frontend fetches real-time tracking information from a Third-Party Shipment API. Shipment details, such as the current location and estimated delivery time, are displayed to the user, and updates are stored in Sanity CMS for record-keeping. This keeps users informed about their shipment progress and delivery timelines.

API Endpoints

1. Get All Products

- **Endpoint:** /products
- **Method:** GET
- **Description:** "Fetch a comprehensive list of products with details like name, price, stock, and category from Sanity CMS."

- **Sample Response:**

```
[
  {
    "id": 1,
    "name": "Dinning Chair",
    "price": 150,
    "stock": 100,
    "category": "Chairs",
    "image": "dinning_chair.jpg"
  },
  {
    "id": 2,
    "name": "Wooden Table",
    "price": 300,
    "stock": 200,
    "category": "Tables",
    "image": "wooden_table.jpg"
  }
]
```

2. Create a New Order

- **Endpoint:** /orders
- **Method:** POST
- **Description:** Submit a new order to Sanity CMS. This includes customer details, product information, and payment status.
- **Sample Payload:**

```
{
  "customerId": "123456789",
  "customerName": "Ali",
  "address": "Karachi, Pakistan",
  "products": [
    {
      "productId": 1,
      "quantity": 5,
      "price": 150
    },
    {
      "productId": 2,
      "quantity": 9,
      "price": 300
    }
  ],
  "totalAmount": 450,
  "paymentStatus": "Pending"
}
```