Marketplace Technical Foundation - Online Furniture Marketplace

---

1. Design System Architecture

System architecture refers to the structured design of a system's components and their interactions. It outlines how different parts of the system work together to achieve the desired functionality.

In the context of an **e-commerce platform**, the architecture consists of the following key components:

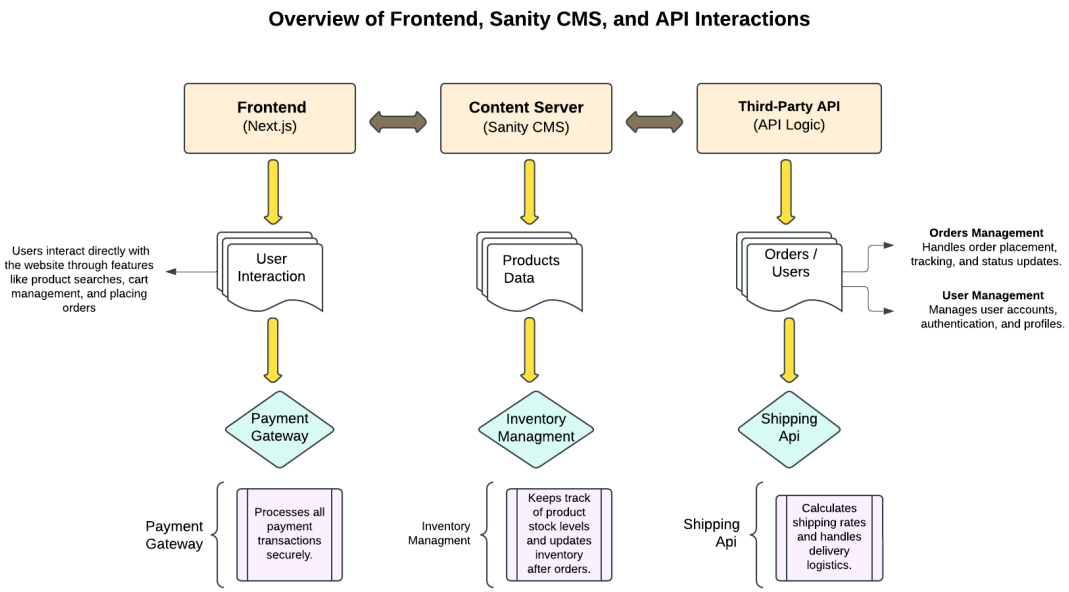
1. **Frontend (Next.js):**
    - The user interface where customers interact with the website (e.g., browsing products, adding items to the cart).
2. **Backend (API):**
    - The logic layer that processes user requests, handles business logic, and communicates with the database and external services.
3. **Database (Sanity CMS):**
    - Stores dynamic content like product details, user information, and order data.
4. **Third-Party APIs:**
    - External services like **payment gateways (e.g., Stripe)** for processing payments and **shipping APIs** for tracking orders.

How It Works (Interactions)

Frontend to Backend:
The frontend sends API requests (e.g., fetch product details, submit orders).

Backend to Database:
The backend retrieves or updates data in the CMS (e.g., product information, order history).

Backend to Third-Party APIs:
Processes payments, manages shipping, and integrates external functionalities.



2. Key Workflows

Frontend (Next.js):

- User signs up → Sanity CMS (Data storage).
- User browses products → Products Displayed (Displays the products fetched from Sanity CMS).
- After the user makes a purchase, order details are saved to Sanity CMS.

Sanity CMS:

- The Sanity CMS stores user data (sign-up) and product data.
- Product data is fetched from Sanity CMS for the frontend.
- Order details are saved to Sanity CMS after the user confirms the purchase.

Third-Party APIs:

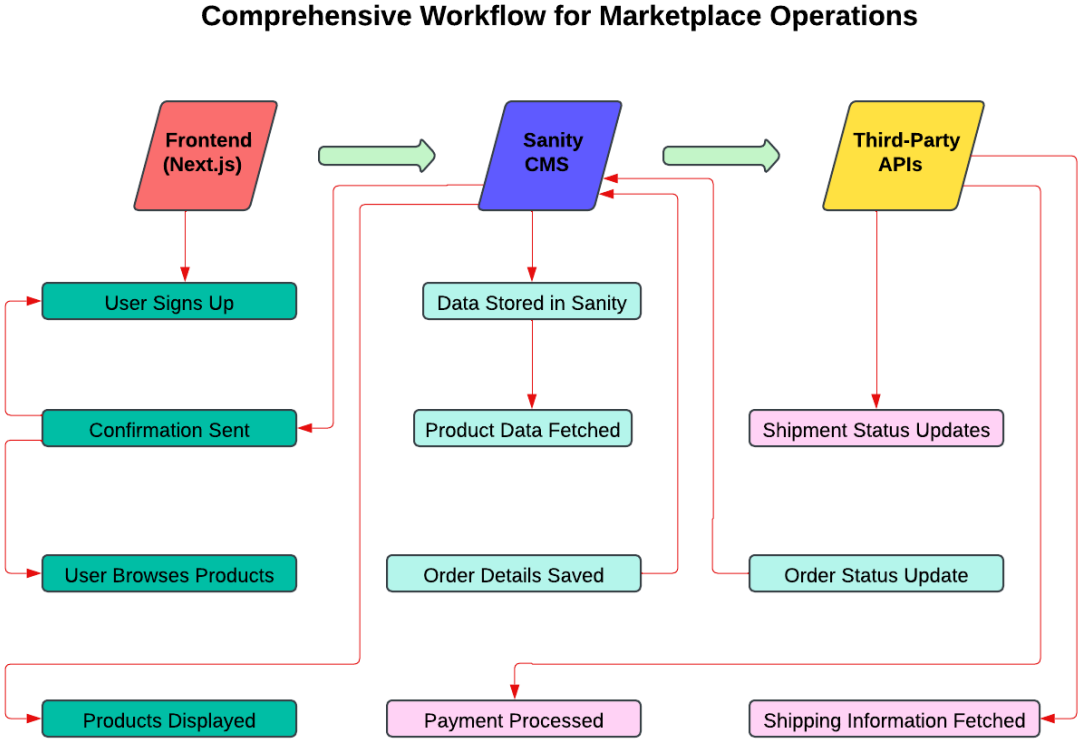
- Shipment Status Updates and Shipping Information are fetched from a third-party API.
- Payment processing and order status updates are handled through third-party APIs as well.

Confirmation:

- After sign-up, a confirmation is sent to the user.
- After payment is processed, order status updates (via a third-party API).

Flow:

1. User signs up → Data is stored in Sanity CMS.
2. User browses products → Product data is fetched from Sanity CMS and displayed on the frontend.
3. After browsing, the user places an order:
   - Order details are saved in Sanity CMS.
   - Payment is processed via a third-party API.
   - Shipping information is fetched from a third-party API.
   - Shipment status updates are fetched from a third-party API.
4. User receives confirmation after sign-up and after payment.
5. The order status is updated via a third-party API after processing.



3. Category-Specific Instructions

General eCommerce Workflows for Furniture

1. **Product Browsing**
   - Users can filter furniture by category (e.g., sofas, beds, dining tables).

2. **Cart Management**
   - Users can add, remove, or update items in their cart.

3. **Order Placement**
   - Includes checkout, payment, and order confirmation workflows.

4. API Endpoints

Endpoint	Method	Purpose	Response Example
`/products`	GET	Fetch all available products from Sanity	{"id": 1, "name": "Product A", "price": 100, "stock": 50, "image": "url"}

Endpoint	Method	Purpose	Response Example
/orders	POST	Save order details to Sanity (includes product and customer info)	{"orderId": 123, "customerId": 456, "products": [...], "total": 500, "status": "Pending"}
/customers	POST	Save customer details to Sanity	{"customerId": 456, "name": "John Doe", "email":"john@example.com", "phone": "1234567890"}
/payment	POST	Process payment for an order	{"orderId": 123, "amount": 500, "paymentStatus": "Success"}
/shipment	GET	Fetch shipment status via third-party API	{"shipmentId": 321, "orderId": 123, "status": "In Transit", "expectedDeliveryDate": "2025-01-20"}

---

5. Sanity Schema

Product Schema

```
const ProductSchema = {
    name: "product",
    title: "Product",
    type: "document",
    fields: [
      {
        name: "name",
        title: "Product Name",
        type: "string",
      },
      {
        name: "slug",
        title: "Slug",
        type: "slug",
        options: {
          source: "name",
          maxLength: 96,
        },
      },
      {
        name: "price",
        title: "Price",
        type: "number",
      },
      {
        name: "description",
        title: "Description",
        type: "text",
      },
      {
```

```
      name: "dimensions",
      title: "Dimensions",
      type: "object",
      fields: [
        {
          name: "height",
          title: "Height",
          type: "string",
        },
        {
          name: "width",
          title: "Width",
          type: "string",
        },
        {
          name: "depth",
          title: "Depth",
          type: "string",
        },
      ],
    },
    {
      name: 'image',
      title: 'Image',
      type: 'image',
    },
    {
      name: "inStock",
      title: "In Stock",
      type: "boolean",
      initialValue: true,
    },
  ],
};
```

```
export default ProductSchema;
```

Order Schema

```
const OrderSchema  = {
    name: 'order',
    type: 'document',
    title: 'Order',
    fields: [
      {
        name: 'orderId',
        type: 'string',
        title: 'Order ID',
      },
      {
        name: 'customer',
        type: 'reference',
        title: 'Customer',
        to: [{ type: 'customer' }],
      },
      {
        name: 'products',
        type: 'array',
        title: 'Ordered Products',
        of: [{ type: 'reference', to: [{ type: 'product' }] }],
      },
      {
        name: 'status',
        type: 'string',
        title: 'Order Status',
        options: {
          list: ['Pending', 'Shipped', 'Delivered', 'Cancelled'],
```

```
        },
      },
      {
        name: 'totalAmount',
        type: 'number',
        title: 'Total Amount',
      },
    ],
  };
  export default OrderSchema;
```

Customer Schema

```
const CustomerSchema = {
    name: 'customer',
    type: 'document',
    title: 'Customer',
    fields: [
      {
        name: 'name',
        type: 'string',
        title: 'Customer Name',
      },
      {
        name: 'email',
        type: 'string',
        title: 'Email Address',
      },
      {
        name: 'phone',
        type: 'string',
        title: 'Phone Number',
      },
```

```
    {
      name: 'address',
      type: 'string',
      title: 'Shipping Address',
    },
  ],
};
export default CustomerSchema;
```
