

System Architecture: Marketplace Technical Foundation

Overview:

The system is designed for an online marketplace where users can browse products, make orders, and track shipments. The architecture is composed of three primary components: Frontend, Backend, and Third-Party Integrations.

Frontend:

- Built using Next.js for dynamic rendering and responsive UI.
- Pages: Home, Product Listing, Product Details, Cart, Checkout, Order Confirmation.

Backend:

- Sanity CMS is used for managing product data, customer data, and orders.
- Sanity schemas are designed to align with business goals.

Third-Party APIs:

- APIs for payment processing and shipment tracking are integrated.

Key Workflows:

1. User Registration: User signs up and data is saved in Sanity CMS. A confirmation is sent to the user.
2. Product Browsing: Users browse the homepage, and the frontend fetches data from Sanity CMS to display products.
3. Order Placement: Users add items to the cart, proceed to checkout, and the order is saved in Sanity CMS.
4. Shipment Tracking: Shipment status is fetched using third-party APIs, and status updates are shown to users.

1. System Architecture

Overview:

- **Frontend:** Built using Next.js for dynamic and responsive interfaces.
- **Backend:** Sanity CMS for data management.
- **Third-Party APIs:** Integration for payment processing and shipment tracking.

Architecture Diagram:

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

Workflows:

1. User interacts with the frontend to browse products.
2. Frontend fetches data from Sanity CMS.
3. Orders are saved in Sanity CMS.
4. Shipment status is retrieved using third-party APIs.
5. Payments are processed securely.

2. Define Technical Requirements

Frontend Requirements:

- User-friendly interface for browsing products.
- Responsive design for mobile and desktop users.
- Essential pages: Home, Product Listing, Product Details, Cart, Checkout, and Order Confirmation.

Backend (Sanity CMS):

- Use Sanity CMS to manage product data, customer details, and order records.
- Focus on designing schemas in Sanity to align with the business goals.

Third-Party APIs:

- Integrate APIs for shipment tracking and payment gateways.
- Ensure APIs provide the necessary data for frontend functionality.

3. Plan API Requirements

API Documentation:

Endpoint	Method	Purpose	Response Example
<code>/products</code>	GET	Fetch all product details	<code>{ "id": 1, "name": "Product A", "price": 100 }</code>
<code>/orders</code>	POST	Save a new order	<code>{ "status": "Success", "orderId": 123 }</code>
<code>/shipment</code>	GET	Track shipment status	<code>{ "orderId": 123, "status": "In Transit" }</code>

Example Detailed Endpoint:

- **Endpoint:** `/express-delivery-status`
- **Method:** GET
- **Description:** Fetch real-time delivery updates for perishable items.
- **Response Example:** `{ "orderId": 123, "status": "In Transit", "ETA": "15 mins" }`

4. Sanity CMS Schemas

Example Schema for Products:

```
export default {
  name: 'product',
  type: 'document',
  fields: [
    { name: 'name', type: 'string', title: 'Product Name' },
    { name: 'price', type: 'number', title: 'Price' },
    { name: 'stock', type: 'number', title: 'Stock Level' },
    { name: 'description', type: 'text', title: 'Description' },
    { name: 'image', type: 'image', title: 'Product Image' }
  ]
};
```

5. Design System Architecture

Key Workflows:

1. **User Registration:**
 - User signs up -> Data saved in Sanity CMS -> Confirmation sent to the user.
 2. **Product Browsing:**
 - User visits homepage -> Sanity API fetches product data -> Data displayed on frontend.
 3. **Order Placement:**
 - User adds items to cart -> Proceeds to checkout -> Order saved in Sanity CMS.
 4. **Shipment Tracking:**
 - Shipment details fetched via API -> Status displayed to user.
-

6. Write Technical Documentation

Required Documents:

1. **System Architecture Document:**
 - Describes the overall design and interaction between components.
2. **API Specification Document:**
 - Details endpoints, methods, payloads, and responses.
3. **Workflow Diagram:**
 - Visualizes user interactions and data flows.
4. **Data Schema Design:**
 - Defines entities and relationships for databases or CMS.

Example Workflow Diagram:

1. User visits the site and browses products.
 2. Frontend requests product data from Sanity CMS.
 3. Order is created and saved in Sanity.
 4. Shipment tracking and payment APIs are integrated.
-

7. Collaborate and Refine

Steps for Collaboration:

1. Organize brainstorming sessions to exchange ideas.
 2. Share your technical plans with peers for feedback.
 3. Use GitHub for version control and clear commit tracking.
 4. Incorporate feedback and refine the technical foundation.
-