# Marketplace Technical Foundation - [****General E-Commerce.****]

## System Architecture Overview

### Architecture Diagram

[Frontend (Next.js)]

|

[Sanity CMS] ---------> [Product Data API]

|

[Third-Party APIs] -----> [Shipment Tracking API]

-----> [Payment Gateway]

### Component Roles

* **Front-end (Next.js):** Responsible for rendering the UI, handling user interactions, and communicating with APIs.
* **Sanity CMS:** Acts as the database for managing product data, customer information, and order details.
* **Third-Party APIs:** Provides functionalities like payment processing and shipment tracking.

## Key Workflows

### 1. User Registration

1. User fills out the registration form on the frontend.
2. Frontend sends data to the Sanity CMS API.
3. Sanity CMS stores the user details and sends a confirmation back to the frontend.

### 2. Product Browsing

1. User navigates to the product listing page.
2. Frontend fetches product data from the Sanity CMS API.
3. Products are displayed dynamically on the frontend.

### 3. Order Placement

1. User adds items to the cart and proceeds to checkout.
2. Frontend sends order details to the Sanity CMS API for storage.
3. Payment gateway processes the payment.
4. Sanity CMS updates the order status to "confirmed."

### 4. Shipment Tracking

1. User views their order status on the frontend.
2. Frontend requests shipment details from the third-party Shipment Tracking API.
3. Tracking information is displayed on the user interface.

## API Requirements

### General Endpoints

| **Endpoint** | **Method** | **Description** | **Request Payload** | **Response Example** |
| --- | --- | --- | --- | --- |
| /products | GET | Fetch all available products. | N/A | { "id": 1, "name": "Product A", "price": 100, "stock": 20 } |
| /orders | POST | Create a new order in Sanity. | { "customerId": 123, "cart": [...] } | { "orderId": 456, "status": "confirmed" } |
| /shipment | GET | Fetch shipment tracking details. | { "orderId": 456 } | { "status": "In Transit", "ETA": "15 mins" } |
| /user/register | POST | Register a new user. | { "name": "John Doe", "email": "..." } | { "userId": 789, "status": "success" } |

### Payment Gateway Endpoints

| **Endpoint** | **Method** | **Description** | **Request Payload** | **Response Example** |
| --- | --- | --- | --- | --- |
| /payment/process | POST | Process a payment. | { "orderId": 456, "amount": 100 } | { "transactionId": 789, "status": "success" } |

## Sanity Schemas

### Product Schema

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' },

],

};

### Order Schema

export default {

name: 'order',

type: 'document',

fields: [

{ name: 'customerId', type: 'reference', to: [{ type: 'customer' }], title: 'Customer ID' },

{ name: 'items', type: 'array', of: [{ type: 'product' }], title: 'Items' },

{ name: 'totalAmount', type: 'number', title: 'Total Amount' },

{ name: 'status', type: 'string', title: 'Order Status' },

],

};

## Technical Roadmap

### Milestones

1. **Frontend Development:**
   * Set up Next.js project.
   * Create essential pages (Home, Product Details, Cart, Checkout, Order Confirmation).
2. **Backend Integration:**
   * Configure Sanity CMS schemas.
   * Integrate Sanity API with frontend.
3. **Third-Party API Integration:**
   * Implement payment gateway.
   * Add shipment tracking functionality.
4. **Testing & Deployment:**
   * Conduct end-to-end testing.
   * Deploy the application on Vercel.

### Deliverables

* Fully functional ecommerce marketplace.
* Complete API documentation.
* Polished user interface with responsive design.