Day 2: Planning the Technical Foundation

Marketplace Technical Foundation - Nike App

1. System Architecture Overview

Diagram:

```
[Frontend (Next.js)]

     |

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

     |

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

     |

[Payment Gateway]
```

Component Descriptions:

- **Frontend (Next.js):** User interface for browsing products, managing carts, and placing orders.
- **Sanity CMS:** Backend to store and manage product data, customer information, and order details.
- **Product Data API:** Fetches product listings from Sanity CMS for the frontend.
- **Shipment Tracking API:** Provides real-time updates on order shipments.
- **Payment Gateway:** Securely processes customer payments.

---

2. Key Workflows

User Workflows:

1. **Product Browsing:**
    - User visits the homepage and browses Nike products.
    - Sanity API fetches product categories and details.
    - Products are displayed dynamically on the frontend.
2. **Cart Management:**
    - User adds products to the cart.
    - Cart items and quantities are stored temporarily on the frontend.
3. **Order Placement:**

    - User proceeds to checkout.
    - Order details are sent to Sanity CMS via API.
    - Payment is processed through the payment gateway.
    - Confirmation is displayed to the user.

4. **Shipment Tracking:**
    - Order status is fetched via the shipment tracking API.
    - Real-time updates are displayed on the user’s dashboard.

---

3. API Endpoints

Endpoint	Method	Purpose	Response Example
/products	GET	Fetches all Nike product details.	{ "id": 1, "name": "Nike Air Max", "price": 150, "stock": 25, "image": "url" }
/orders	POST	Creates a new order.	{ "orderId": 123, "status": "Order Placed", "paymentStatus": "Paid" }
/shipment	GET	Fetches shipment status for an order.	{ "shipmentId": 456, "orderId": 123, "status": "In Transit", "ETA": "2 days" }

---

4. Sanity Schema Examples

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: 'category', type: 'string', title: 'Category' },
    { name: 'image', type: 'image', title: 'Product Image' }
  ]
};
```

Order Schema:

```
export default {
  name: 'order',
  type: 'document',
  fields: [
    { name: 'customerId', type: 'string', title: 'Customer ID' },
    { name: 'products', type: 'array', of: [{ type: 'reference', to: [{ type: 'product' }] }], title: 'Products' },
    { name: 'paymentStatus', type: 'string', title: 'Payment Status' },
    { name: 'orderDate', type: 'datetime', title: 'Order Date' }
  ]
};
```