

Planning the Technical Foundation

1. System Architecture Overview

Diagram:

Frontend (Next.js)

|

Sanity CMS -----> Product Data API

|

Third-Party API -----> Shipment Tracking API

|

Payment Gateway

Descriptions:

- **Frontend (Next.js):** The user interface for interacting with products, managing shopping carts, and completing purchases.
- **Sanity CMS:** The backend system that stores and manages data related to products, customers, and order information.
- **Product Data API:** Retrieves product listings and details from Sanity CMS for display on the frontend.
- **Shipment Tracking API:** Offers real-time tracking information for customer orders during shipping.
- **Payment Gateway:** A secure system that handles customer transactions and processes payments efficiently.

2. Key Workflows

User Workflows:

1. Product Browsing:

- The user visits on the homepage and explores the available products.

- Sanity API fetches product categories and product details. ○ Products showed dynamically on the frontend.

2. Cart Management:

- User can add products to the cart.
- Cart items and product 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
/product-list	GET	Fetches all Nike product details.	{ "id": 1, "name": "Nike Air pro", "price": 480, "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": "3 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' }
];
};
```

Customer Schema:

```
export default {
name: 'customer',
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: 'contactInfo', type: 'string', title: 'Payment Status' },
{ name: 'orderDate', type: 'datetime', title: 'Order Date' }
]
};
```

Payment Schema:

```
export default {
  name: 'payment',
  type: 'document',

  fields: [
    { name: 'customerId', type: 'string', title: 'Customer ID' },
    { name: 'orderId', type: 'string', title: 'Order ID' },
    { name: 'paymentStatus', type: 'string', title: 'Payment Status' },
    { name: 'price', type: 'number', title: 'Price' },
  ]
};
```