

## "Marketplace Technical Foundation - [Nike]"

[illegible]

I have created the following pages in my E-commerce website using Next.js and Tailwind CSS:

Home Page: Displays key components ( e.g Hero Section, Gear Up Section, Best Of Air max etc.).

### Products Page: Fetch all products dynamically from Sanity CMS.

Product Detail Page: Dynamic route to display individual product details.

Cart Page: List all added products, fetch data from Sanity (POST method).

Checkout Page: Form where users submit their details for an order.

## Backend

Use Sanity CMS to manage:

## Products Data

## Order Data

### Customer Details

## Create schemas for products, orders, and customers in Sanity CMS.

<<<<<<<<<<<<<<<<<<<<System Architecture Blueprint>>>>>>>>>>>>>>>>>>>

[Frontend (Next.js + Tailwind)]

1

↓

[Sanity CMS (Database)] ↔ [ShipEngine (Shipment API)] ↔ [Payment Gateway]

[illegible]

## User Browses Products

Frontend fetches product data from Sanity CMS using a GET API request.

The product data is displayed dynamically on the Products page.

## User Places Order

Cart data is stored in Sanity CMS using a POST API request.

User details are submitted through the checkout form and stored in Sanity.

## Shipment Tracking & Payment

After order submission, shipment data is retrieved using ShipEngine's API.

Payment details are processed via the payment gateway and stored securely.

[illegible]

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

```
export default {
  name: 'order',
  type: 'document',
  fields: [
    { name: 'customerName', type: 'string', title: 'Customer Name' },
    { name: 'products', type: 'array', of: [{ type: 'reference', to: [{ type: 'product' }] }], title: 'Products' },
    { name: 'totalPrice', type: 'number', title: 'Total Price' },
    { name: 'status', type: 'string', title: 'Order Status' },
    { name: 'shipmentTracking', type: 'string', title: 'Shipment Tracking ID' },
  ],
};
```

```
export default {  
  name: 'customer',  
  type: 'document',  
  fields: [  
    { name: 'name', type: 'string', title: 'Full Name' },  
    { name: 'email', type: 'string', title: 'Email Address' },  
    { name: 'address', type: 'string', title: 'Address' },  
    { name: 'phone', type: 'string', title: 'Phone Number' },  
  ],  
};
```

```
export default {  
  name: 'checkout',  
  type: 'document',  
  fields: [  
    { name: 'customer', type: 'reference', to: [{ type: 'customer' }], title: 'Customer' },  
    { name: 'order', type: 'reference', to: [{ type: 'order' }], title: 'Order' },  
    { name: 'paymentMethod', type: 'string', title: 'Payment Method' },  
    { name: 'shippingAddress', type: 'text', title: 'Shipping Address' },  
    { name: 'billingAddress', type: 'text', title: 'Billing Address' },  
    { name: 'paymentStatus', type: 'string', title: 'Payment Status' },  
    { name: 'orderDate', type: 'datetime', title: 'Order Date' },  
    { name: 'shippingStatus', type: 'string', title: 'Shipping Status' },  
  ],  
};
```

```
<<<<<<<<<<<<<<<<<< system architecture diagram:>>>>>>>>>>>>>>>>>
```

## User Views Products



[Sanity API] → Fetch Product Data → Display Products on Frontend



## User Adds Items to Cart



[POST Request to Sanity] → Save Order in Sanity



## User Tracks Shipment



### [ShipEngine API] → Get Shipment Tracking Status



User Pays



[Payment Gateway] → Process Payment Transaction



## Order Confirmed

## Key Workflows

1. **User views products** → Sanity API provides product data → Displayed on the frontend.
2. **User adds items to cart** → POST request to Sanity saves cart data.
3. **User places an order** → Order data stored in Sanity.
4. **User tracks shipment** → ShipEngine provides tracking status.
5. **User pays** → Payment Gateway processes the transaction securely.