Planning the Technical Foundation for Nike eCommerce

This workflow outlines the complete technical foundation for a Nike eCommerce website. The system is divided into three main components: Frontend, Backend, and Third-Party APIs. Each section details the essential processes, including product management, user interaction, and order processing, to ensure seamless functionality and an optimal user experience.

1. Frontend

The frontend is the user interface where customers interact with the e-commerce platform. Built using React or Next.js, it involves:

Key Components:

Folder Structure:

Pages: Contains routes like Home, Product Details, Cart, and Checkout.

Components: Reusable UI elements like Navbar, Footer, Product Card, etc.

Public: Static assets like images, fonts, etc.

Main Pages and Features:

Home Page: Displays products dynamically fetched from Sanity CMS.

Product Details Page: Displays specific product details such as name, price, description, and stock availability.

Cart Page:

Adds selected products.

Shows the quantity and total price.

Checkout Page:

Collects customer details.

Handles member checkout for registered users.

Displays order summary and delivery details.

2. Sanity CMS

Sanity is used as a headless CMS to manage and store backend data.

Key Schemas:

Products: Stores product details such as title, price, image, category, stock, and description.

Orders: Manages customer orders, including items, total price, and order status.

Customers: Stores customer details like name, email, and address.

Categories: Helps group products into categories for easy navigation.

Role in Workflow:

Acts as the central database for managing content like products, orders, and customer data.

Provides real-time updates to the frontend via APIs.

3. Third-Party APIs

Third-party APIs handle functionalities that are not part of the core CMS or frontend logic.

Key APIs:

Payment Gateway: Processes customer payments securely.

Shipment Tracking API: Tracks the shipment status of orders.

Role in Workflow:

Payment Gateway: Verifies and processes payments, ensuring smooth transactions.

Shipment Tracking API: Provides real-time shipment updates, which can be displayed to customers.

4. Roles in Workflow

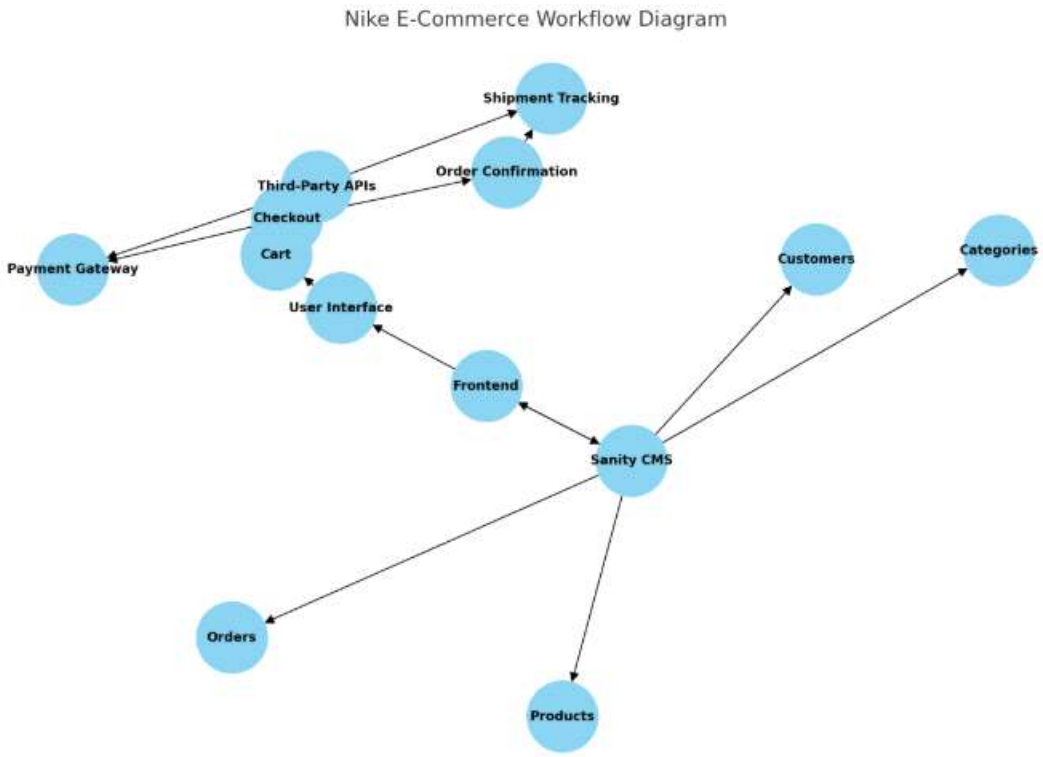
Here's how the components work together:

Frontend: Fetches product and customer data from Sanity CMS to display and manage interactions.

Sanity CMS: Acts as the database for storing and retrieving all e-commerce data.

Third-Party APIs: Completes transactions and logistics by processing payments and tracking shipments.

5. Workflow Diagram



6. Technical Roadmap

1: Project Setup

Initialize the Next.js project.

Install dependencies: sanity, axios, and required UI libraries.

Set up Sanity CMS schemas for Products, Categories, and Orders.

2: Frontend Development

Develop the Home, Product Details, Cart, and Checkout pages.

Fetch data from Sanity CMS using GROQ queries or APIs.

Implement cart functionality using React Context or Redux.

3: Integrate Third-Party APIs

Add payment gateway integration (e.g., Stripe).

Integrate shipment tracking API for order updates.

Week 4: Testing and Deployment

Test API endpoints, payment functionality, and frontend performance.

Deploy the app using Vercel or Netlify.

7. Key Outputs:

Frontend: Fully responsive and functional UI for an e-commerce platform.

CMS: Sanity CMS with well-defined schemas for products and orders.

API Integration: Payment gateway and shipment tracking.

Deployment: Live e-commerce website.