Marketplace Technical Foundation Document

---

1. Project Overview

- **Goal:** Build an e-commerce marketplace for athletes, fitness enthusiasts, and fashion-conscious individuals interested in athleisure and stylish sportswear.
- **Platform:** A responsive and pixel-perfect e-commerce website with product browsing, search functionality, and easy checkout.
- **Target Audience:** Athletes, fitness enthusiasts, and individuals with an active lifestyle.

---

2. Tech Stack

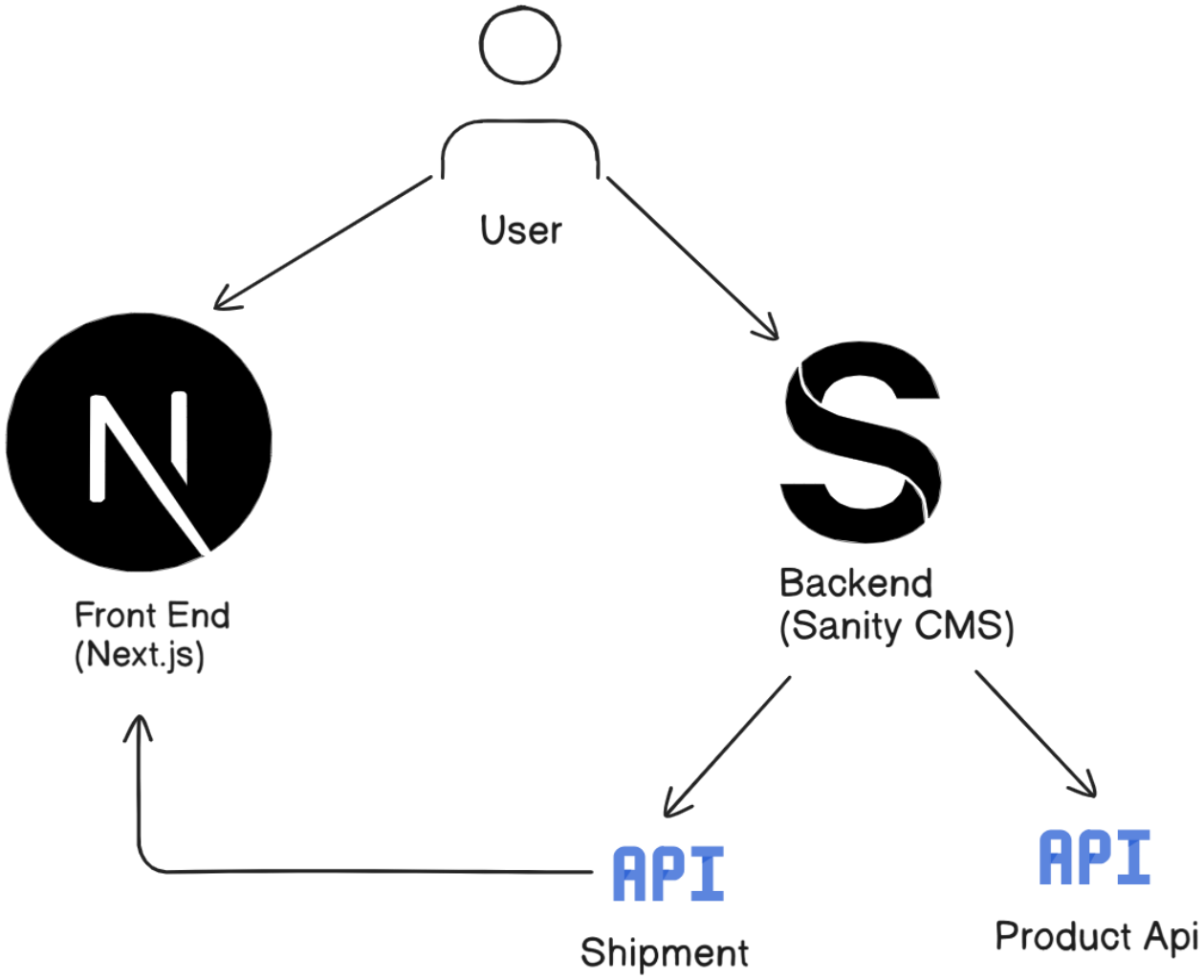
- **Frontend:**
  - **Next.js:** Framework for building the React-based frontend, offering server-side rendering (SSR) and static site generation (SSG) for better performance and SEO.
  - **Tailwind CSS:** Utility-first CSS framework for fast and responsive styling.
  - **ShadCN Select:** Customizable select component for category filtering in the product listing page.
  - **JavaScript (TypeScript):** To maintain type safety, code consistency, and scalability in the project.
- **Backend:**
  - **Sanity (Headless CMS):** A flexible and powerful backend for managing content such as products, images, categories, and customer orders.
  - **Custom API (Next.js API routes):** For handling user authentication, order placement, and managing product data.
- **Database:**
  - Managed by Sanity for storing product details, categories, user accounts, and orders.
- **Hosting:**

- **Vercel (for Next.js app):** To deploy the frontend with ease and benefit from serverless functions.
- **Sanity Studio:** For managing content (e.g., products, categories, etc.) directly from the Sanity CMS interface.

---

3. Features

1. **Product Listing Page:**
   - Display a grid of products with images, names, prices, and categories.
   - Filters:
     - Category filter using a **ShadCN Select** component.
     - Price range filter (if applicable).
   - Search bar for product search functionality.
2. **Product Detail Page:**
   - Each product will have a dedicated page showing more details like product description, size options, color choices, and add-to-cart functionality.
3. **Shopping Cart:**
   - Users can add products to the cart, view the cart, and proceed to checkout.
   - Cart will hold product details, quantity, and total price.
4. **Checkout Process:**
   - A multi-step process for entering user details, shipping address, payment, and order confirmation.
5. **User Authentication:**
   - Users can sign up, log in, and view their order history.
   - Session management for handling user login state.
6. **Admin Panel (Sanity Studio):**
   - Ability to add, update, and delete products.
   - Manage categories and view user orders.
   - Product inventory and order management.



4. Project Architecture

1. **Frontend:**
    - **Pages:**
        - `/` (Home): Display featured products, categories, and promotions.
        - `/product/[id]`: Product detail page.
        - `/cart`: Shopping cart page.
        - `/checkout`: Checkout form.
        - `/profile`: User account details and order history.
    - **Components:**
        - **Navbar**: Navigation bar with links to home, categories, cart, and profile.

        - **Footer**: Footer with essential information like contact, social links, etc.
        - **ProductCard**: Display individual product details.
        - **FilterBar**: Display filters like category and price range.
        - **SearchBar**: Product search bar.
    - **State Management:**
        - **React Context API** or **Redux** for managing the cart and user authentication states.
2. **Backend:**
    - **Next.js API Routes:**
        - Handle POST requests for order placement.
        - Handle GET requests for product data (from Sanity).
        - Handle user authentication (if not using a third-party auth system).
        - Optionally, handle payment processing (integrate with Stripe or PayPal).
3. **Sanity CMS Setup:**
    - Create document schemas for products, categories, and orders.
    - Use Sanity's flexible content studio to manage all product and order information.

---

5. User Flow

1. **Visitor Flow:**
    - Users land on the homepage with product listings.
    - They can filter or search for products based on categories and price.
    - They can view product details, add to the cart, and checkout as guests or sign in.
2. **Authenticated User Flow:**
    - Users can sign up and log in to their accounts.
    - After signing in, they can view their cart, past orders, and update profile details.
3. **Admin Flow:**
    - Admins can manage products, categories, and view user orders in the Sanity Studio.

---

6. Security & Performance Considerations

1. **Authentication & Authorization:**
    - Implement secure user authentication using Clerk.
    - Session management to ensure secure login/logout.
2. **Security Best Practices:**
    - Use environment variables for sensitive API keys and secrets.
    - Regular vulnerability testing and security updates.

Conclusion

This technical foundation provides the necessary structure to start building your e-commerce marketplace with Next.js and Sanity. By focusing on performance, security, and a smooth user experience, you'll be able to create an efficient and scalable platform that resonates with your target audience.