Market Place Technial Foundation

Nike

Hackathon Day 2: Planning The Technical Foundation

1. Tech Stack Selection

Framework: Next.js 15 with TypeScript for server-side rendering and dynamic routing.

Styling: Tailwind CSS for responsive and pixel-perfect design.

CMS: Sanity CMS for managing product inventory, descriptions, and promotions.

Database: MongoDB or Firebase for storing user data, orders, and analytics.

APIs: REST or GraphQL for communication between the frontend and backend.

Payment Integration: Stripe for secure payment handling.

Hosting: Vercel for seamless deployment.

2. Core Functionalities

Frontend
Dynamic product pages with filters (size, color, price).

User-friendly navigation (Home, Products, Sales, About Us, Contact).

Product recommendations using AI-powered APIs (e.g., Algolia).

Wishlist and cart management with local storage or database integration.

Backend
User authentication with JWT or OAuth (Google, Facebook).

Real-time inventory management synced with Sanity CMS.

APIs for product search, reviews, and order tracking.

Shipment integration (e.g., EasyPost, Shippo) for live tracking.

3. Database Architecture

Collections:

Users: User profiles, wishlist, order history.

Products: Details (name, price, SKU, description, inventory).

Orders: Order details (userID, productIDs, shipment status).

Reviews: Product reviews and ratings.

Optimize queries using indexing and caching for faster performance.

4. Design Approach

Minimalist UI using Nike's branding colors (black, white, red).

Mobile-first design for responsive layout.

Accessibility (WCAG standards) for broader user inclusion.

5. Key Features

Product Reviews: Allow customers to leave ratings and comments.

Dynamic Promotions: Time-sensitive discounts managed through Sanity CMS.

Real-Time Search: Implement Algolia for lightning-fast product search.

Shipment Tracking: Let users track orders in real time.

Performance Optimization: Use lazy loading, image compression, and code splitting.

6. Development Workflow

1. Project Setup:

Initialize with create-next-app.

Set up Tailwind CSS and configure Sanity CMS.

2. Frontend Implementation:

Build reusable components (Header, Footer, Product Card).

Create dynamic routes for product pages.

3. Backend Integration:

Set up API routes in Next.js.

Configure Sanity and Stripe.

4. Testing:

Use Jest or Cypress for end-to-end testing.

5. Deployment:

Deploy on Vercel with environment variables for sensitive keys.

System Architecture

```
        Frontend (Next.js + Tailwind CSS)
                        |
          Fetch product data, user info, etc.
                        |
        Sanity CMS (Content Management)
                        |
   Store product details, inventory, orders, etc.
                        |
   --------------------------------------------------
            |                                 |
Product Data API                  Shipment Tracking API
 (Fetch inventory, reviews, etc.)   (Live tracking via
                  EasyPost/Shippo)
                        |
                Third-Party APIs
  (Stripe for payments, Algolia for search, etc.)
```