**Marketplace Builder Hackathon 2025**

**DAY -2 PLANNING THE TECHNICAL FOUNDATION**

### **Objective**

Your website, **"Comforty Shoes"**, will focus on selling comfortable shoes. It will emphasize comfort, style, and durability, making it easy for users to find the perfect pair for casual, professional, or athletic use.

### **Website Features and Requirements**

#### **Frontend Requirements:**

1. **User-Friendly Interface:**
   * Intuitive layout with categories like **Casual Shoes**, **Formal Shoes**, and **Athletic Shoes**.
   * Clean product displays, ensuring users can easily explore designs and features.
2. **Responsive Design:**
   * A mobile-first design ensures excellent performance on mobile and desktop devices.
   * Flexible grids, responsive images, and interactive elements optimized for all screen sizes.
3. **Essential Pages:**
   * **Home Page:** Highlight key categories and featured shoes with attractive banners.
   * **Product Listing Page:** Display shoes with filters for size, color and price.
   * **Product Details Page:** Show in-depth details like shoe material, comfort rating, available sizes, and high-quality images.
   * **Cart Page:** Allow users to view, update, or remove items before checkout.
   * **Checkout Page:** Offer a secure and straightforward way to finalize purchases.
   * **Order Confirmation Page:** Summarize the order with delivery and payment details.

#### **Backend Requirements with Sanity CMS**

1. **Product Management:**
   * Manage shoe details (Product name , Category, size , Price and style).
   * Seamlessly update the product catalog in Sanity CMS.
2. **Customer and Order Management:**
   * Track customer data (name, contact details, and order history).
   * Manage orders and shipping information dynamically.

### **System Architecture:**

1. **Frontend:**
   * Built with React and Next.js for dynamic rendering and a responsive UI.
   * Tailwind CSS for styling.
2. **Backend:**
   * Sanity CMS for managing content and product data.
   * Custom API endpoints to handle data fetching and updating.
3. **Tools & Libraries:**
   * **Clerk:** Authentication.
   * **Sanity CMS:** Content management.
   * **ShipEngine API:** Real-time shipping updates.
   * **Stripe:** Secure payment gateway integration.
   * **React Context API:** Manage cart functionality.

### **API Endpoints**

1. **/products (GET):**
   * Retrieve a list of all shoes available.
   * Include filters like size, color, and price for better usability.
2. **/product/{id} (GET):**
   * Fetch details of a specific shoe, including images, available sizes, material, and ratings.
3. **/orders (POST):**
   * Record an order after checkout with all necessary details (customer info, items, shipping address).
4. **/shipment/{orderId} (GET):**
   * Provide real-time tracking for orders, from dispatch to delivery.

### **Workflow Explanation**

1. **Clerk (Authentication):**
   * Simplifies the sign-up/login process with secure authentication.
2. **Sanity CMS:**
   * Stores and updates shoe inventory, including categories, product descriptions, and customer information.
3. **ShipEngine API:**
   * Generates shipping labels and tracks orders across delivery stages.
4. **Stripe (Payment Gateway):**
   * Ensures a secure checkout process with multiple payment methods.
5. **React Context API:**
   * Manages the cart state across components, enabling real-time updates.

### **Next Steps for Your Shoes Website**

1. **Set Up Frontend Pages:**
   * Create the essential pages mentioned above using Next.js and Tailwind CSS.
2. **Integrate Sanity CMS:**
   * Use schemas to manage shoe data, including categories, product details, and orders.
3. **Develop APIs:**
   * Implement custom APIs to fetch products, handle orders, and manage shipment statuses.
4. **Implement Clerk Authentication:**
   * Add secure login and registration features for customers.
5. **Set Up Payments with Stripe:**
   * Configure Stripe for secure transactions with options for multiple payment methods.
6. **Enable Shipping with ShipEngine:**
   * Use ShipEngine to handle real-time shipment tracking and rate calculations.
7. **Test & Optimize:**
   * Perform thorough testing for responsiveness, performance, and security before launch.