**Day 4 Action Plan & Checklist**

**What We Do Now: Required for Submission**

We will focus on the core functionality, which is required for submission. The goal is to complete the essential components and submit a fully functional product. After submission, we can enhance and improve the features.

**1. Core Components for Submission (Day 4)**:

* **Product Listing Component**: Dynamically render product data in a grid layout, showing name, price, image, and stock status.
* **Product Detail Component**: Create individual product detail pages with dynamic routing (using Next.js), displaying product descriptions, price, and available sizes/colors.
* **Category Component**: Display categories fetched dynamically and enable filtering of products based on selected categories.
* **Search Bar**: Implement search functionality to filter products by name or tags.
* **Cart Component**: Display added items with quantity and total price, using state management to track cart items.
* **Pagination Component**: Break down large product lists into manageable pages with previous/next or numbered pagination.
* **Footer and Header Components**: Build consistent navigation and branding, ensuring responsiveness and accessibility.

**2. Submission Requirements:**

* **Functional Deliverables**:
  + Screenshots or screen recordings of the following:
    - Product listing page with dynamic data.
    - Product detail pages with routing and data rendering.
    - Working category filters, search bar, and pagination.
  + Code snippets for key components (ProductCard, ProductList, SearchBar).
  + Scripts or logic for API integration and dynamic routing.
* **Documentation**:
  + A technical report including:
    - Steps taken to build and integrate components.
    - Challenges and solutions.
    - Best practices followed.
  + Submit the document in PDF or Markdown format.
* **Repository Submission**:
  + Upload all files to your designated GitHub repository with a well-structured folder hierarchy.

**What We Work on Later: Post-Submission Enhancements**

These features are not required for submission right now but will improve the project. We will work on them after the core components are completed and submitted.

**1. Advanced Features for Improvement**:

* **Wishlist Component**: Allow users to save products for future reference using local storage or global state management.
* **Checkout Flow Component**: Create a multi-step form for checkout (billing, shipping, and payment details).
* **User Profile Component**: Display user details such as name, email, saved addresses, and order history.
* **Reviews and Ratings Component**: Allow users to submit and view reviews for products.
* **Filter Panel Component**: Provide advanced filtering options (price range, brand selection, in-stock only).
* **Related Products Component**: Suggest similar products on product detail pages.
* **Notifications Component**: Show real-time alerts (add to cart, errors, successful purchases).
* **Analytics Dashboard Component**: Display KPIs such as sales, user traffic, and popular products.
* **Multi-Language Support Component**: Implement a language switcher and dynamically fetch translations.
* **Order Tracking Component**: Provide users with real-time order status updates.
* **Subscription Management Component**: Allow users to manage subscriptions and renewal dates.
* **Admin Dashboard Component**: Interface for managing products, orders, users, and categories.

**2. Non-Essential Features for Later**:

* **Product Comparison Component**: Allow users to compare products side-by-side.
* **Discount and Promotion Component**: Manage discounts, coupons, and promotional campaigns.
* **Social Media Sharing Component**: Enable users to share products on social media.
* **Bulk Upload Component**: Allow admin users to upload large amounts of product data via CSV or Excel files.
* **AI Recommendations Component**: Recommend products based on user preferences or purchase history.
* **Gift Card and Voucher Component**: Track gift card balances and usage.
* **Customer Feedback Component**: Collect and display user feedback.
* **Advanced Search Component**: Implement faceted search to refine results with multiple filters.

**Steps for Implementation:**

1. **Core Functionality**:
   * Set up your Next.js project with Sanity CMS or API for data fetching.
   * Build core components: Product Listing, Product Detail, Category, Search Bar, Cart, Footer, and Header.
   * Test functionality and dynamic routing.
2. **Testing and Integration**:
   * Confirm the integration with Sanity CMS/API for dynamic data fetching.
   * Ensure all components work as expected, with responsive design.
3. **Documentation**:
   * Document your process, challenges, and best practices.
   * Provide clear explanations and screenshots for the key components.
4. **Submission**:
   * Prepare the final code and documentation for submission by uploading to GitHub.
   * Make sure everything is structured properly.

### Conclusion: We will focus on building and submitting the core components first. Once everything is set up, we will work on improving the project by adding advanced features post-submission. Let's work steadily to meet the submission requirements first. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ ****Immediate Focus: Files Needed for Submission and Core Functionality****

#### 1. ****Components:****

* These components need to be functional for dynamic data handling:
  + **src/app/components/layout/Header.tsx**
  + **src/app/components/layout/Footer.tsx**
  + **src/app/components/sections/HeroSection.tsx**
  + **src/app/components/sections/FeaturedProducts.tsx**
  + **src/app/components/sections/OurProducts.tsx**
  + **src/app/components/sections/ProductCard.tsx**
  + **src/app/components/sections/TopCategories.tsx**
  + **src/app/components/sections/ExploreNewStyles.tsx**
  + **src/app/components/sections/BrandingCards.tsx**
  + **src/app/components/shared/Button.tsx**
  + **src/app/components/shared/Input.tsx**

#### 2. ****Context:****

* For Cart and state management:
  + **src/context/CartContext.tsx**

#### 3. ****Sanity Integration:****

* **src/sanity/env.ts**
* **src/sanity/lib/client.ts**
* **src/sanity/lib/image.ts**
* **src/sanity/lib/live.ts**
* **src/sanity/schemaTypes/product.ts**
* **src/sanity/schemaTypes/category.ts**
* **src/sanity/structure.ts**

#### 4. ****Pages for Dynamic Routing:****

* **src/app/product/[id]/page.tsx**
* **src/app/products/page.tsx**
* **src/app/cart/page.tsx**
* **src/app/about/page.tsx**
* **src/app/contact/page.tsx**
* **src/app/faq/page.tsx**
* **src/app/terms/page.tsx**
* **src/app/page.tsx** (main entry page)

#### 5. ****API Integration for Categories and Products:****

* **src/app/api/categories/route.ts**
* **src/app/api/products/route.ts**

#### 6. ****Globals and Layout Styling:****

* **src/styles/globals.css**
* **src/app/layout.tsx**
* **tailwind.config.ts**

#### 7. ****Utilities/Helper Files:****

* **src/libs/fetcher.ts**
* **src/libs/hooks/useFetch.ts**

### ****Later Improvements (To Be Addressed After Initial Submission)****

#### 1. ****Testing and Error Handling:****

* Ensure error handling for API fetches and state management.

#### 2. ****Pagination & Advanced Search:****

* **Pagination Component** in **ProductList** and **ProductsPage**.
* **Search Bar Component** for filtering products dynamically.

#### 3. ****Additional Features for User Engagement:****

* **Wishlist Component**
* **Cart Flow Integration with Notifications**
* **Reviews and Ratings Component**
* **User Profile Component**

#### 4. ****Sanity Features (Advanced):****

* **Implementing Real-Time Changes** in **Sanity** for instant updates.
* **Adding Admin Functionality** for managing categories and products.

#### 5. ****Documentation Refinements:****

* Prepare a **professional submission document** summarizing work, challenges faced, solutions provided, and any features built beyond the basic requirements.
* Code comments for clarity and best practices.

#### 6. ****Optional Advanced Components:****

* **Multi-Language Support**
* **Discount & Promotions**
* **AI Product Recommendations**
* **Social Media Sharing**

### ****Files to Avoid Direct Modifications (for Now)****

* **📁build**
* **📁cache**
* **📁node\_modules**
* **📁public** (unless adding images, not core components)
* **📁scripts**
* **📁static**

#### ****Next Steps:****

1. **Start with the core components first.**
2. **Ensure API routes work with Sanity.**
3. **Work on dynamic rendering for product listing and individual product pages.**
4. **Submit the core components with screenshots and a functional demo (if possible).**

Let me know if you need further clarifications or adjustments! This list should help streamline the workflow. 💙

------------------------------------------------------------------------------------------------------------------------------------  
  
  
Thank you for sharing the folder structure! Based on the action plan you provided, I’ll organize everything accordingly. I’ll list the necessary core components for submission and future improvements, ensuring that the structure reflects the immediate needs and the features to work on later.

### Current Focus (For Submission):

#### 1. ****Core Components for Submission (Day 4):****

* **Product Listing Component**:
  + File Path: src/app/productListingPage.tsx (Page for displaying products dynamically).
  + Needed to render the product data in a grid layout.
* **Product Detail Component**:
  + File Path: src/app/product/[id]/page.tsx (Page for individual product details).
  + This handles the dynamic routing and renders product details, description, price, and available sizes/colors.
* **Category Component**:
  + File Path: src/app/api/categories/route.ts (API route to fetch categories dynamically).
  + Allows filtering of products by categories.
* **Search Bar**:
  + File Path: src/components/shared/Input.tsx (Assuming this component is used for input functionality).
  + This component will be used to implement search functionality for filtering products by name or tags.
* **Cart Component**:
  + File Path: src/app/cart/page.tsx (Page that shows added items to the cart).
  + This will manage the cart items using state and display the quantity and total price.
* **Pagination Component**:
  + We can create a pagination component under src/components/shared/Button.tsx, assuming that pagination for the product list will be controlled via buttons.
* **Footer and Header Components**:
  + File Paths:
    - src/components/layout/Header.tsx (Header component).
    - src/components/layout/Footer.tsx (Footer component).
  + These components will ensure consistent navigation and branding across all pages and will need to be responsive.

### Future Work (Post-Submission Enhancements):

#### 2. ****Advanced Features for Improvement (Post-Submission):****

* **Wishlist Component**: Store user’s favorite products for future reference.
* **Checkout Flow Component**: Multi-step form for checkout with billing, shipping, and payment details.
* **User Profile Component**: Display user details, addresses, and order history.
* **Reviews and Ratings Component**: Allow users to leave reviews and ratings for products.
* **Filter Panel Component**: Advanced filtering options like price range, brand selection, and in-stock only.
* **Related Products Component**: Show similar products on the product details page.
* **Notifications Component**: Real-time alerts for cart additions, errors, and purchases.
* **Analytics Dashboard Component**: Display key performance indicators (KPIs) like sales, user traffic, and popular products.
* **Multi-Language Support Component**: Enable language switching and dynamic fetching of translations.
* **Order Tracking Component**: Track real-time status updates of user orders.
* **Subscription Management Component**: Manage subscriptions and renewal dates.
* **Admin Dashboard Component**: Admin interface for managing products, users, and orders.

#### 3. ****Non-Essential Features (Post-Submission):****

* **Product Comparison Component**: Allows users to compare products.
* **Discount and Promotion Component**: Manage discounts, coupons, and promotions.
* **Social Media Sharing Component**: Enable users to share products.
* **Bulk Upload Component**: Admin feature to upload large datasets of product information.
* **AI Recommendations Component**: Suggest products based on user behavior.
* **Gift Card and Voucher Component**: Track gift card balances.
* **Customer Feedback Component**: Collect user feedback and display it.
* **Advanced Search Component**: Implement faceted search with multiple filters.

### Steps for Implementation:

1. **Core Functionality Setup**:
   * Start with setting up Sanity CMS (or API) for data fetching (Product, Category, etc.).
   * Create the core components like ProductListing, ProductDetail, Cart, Category, and SearchBar.
   * Test routing, dynamic data fetching, and basic functionality.
2. **Testing and Integration**:
   * Ensure proper integration with Sanity (or API) to fetch product and category data.
   * Ensure responsive design for each component.
3. **Documentation**:
   * Write the technical report covering:
     + The steps to build and integrate the components.
     + Challenges faced and solutions.
     + Best practices followed.
4. **Final Submission**:
   * Upload all completed files to GitHub with proper folder structure.
   * Include screenshots or recordings as required by the submission guidelines.
   * Make sure everything is well-documented in a PDF or Markdown file.