Day 4: Filtering Data, Dynamic Pages, Cart, Checkout, and Order Placement

On Day 4, the focus was on implementing advanced product filtering, setting up dynamic pages for product details, integrating an add-to-cart functionality, enabling a smooth checkout experience, and allowing users to place orders.

1. Filtering Data

To enhance user experience, a filtering system was implemented to allow customers to find products based on different attributes such as category, price range, and brand. This was achieved using:

- **Sanity Queries**: Querying products dynamically from Sanity based on user-selected filters.
- **State Management**: Utilizing `useState` and `useEffect` in Next.js to manage selected filter states.
- **Real-Time Updates**: Ensuring that filter selections dynamically update the displayed products without requiring a full-page reload.

2. Dynamic Product Pages

Each product now has its own dynamic page where users can view detailed information before making a purchase.

- **Next.js Dynamic Routes**: Implemented `[slug].tsx` inside the `/products` directory to generate product pages dynamically.
- **Data Fetching**: Used `getStaticProps` and `getStaticPaths` to pre-fetch product details for optimized performance.
- **Detailed View**: Displaying high-quality images, descriptions, pricing, and availability.

3. Add to Cart Functionality

Users can now add products to their cart, where selected items persist until checkout.

- **Local Storage & State Management**: Storing cart items in local storage for persistence across sessions.
- **Cart UI**: Implemented a sidebar/cart drawer that dynamically updates when items are added or removed.
- **Quantity Management**: Users can increase/decrease product quantities or remove items from the cart.

4. Checkout Process

A seamless checkout process was integrated, allowing users to finalize their purchases effortlessly.

- **Clerk Authentication**: Ensured users are logged in before proceeding to checkout.

- **Stripe Integration**: Implemented a secure payment gateway using Stripe.
- **Shipping & Billing Forms**: Created a step-by-step form for users to enter shipping and billing details.
- **Order Summary**: Displaying a summary of selected items, total cost, and payment options before confirming the purchase.

5. Placing Orders

Once checkout is completed, the order is placed, and users receive a confirmation.

- **Order Storage in Sanity**: Successfully saving order details in Sanity for tracking.
- **Email Confirmation**: Sending confirmation emails to users upon successful order placement.
- **Order Tracking Page**: Created an `/orders` page where users can view their order history and status.

With these features implemented, the marketplace now provides a complete shopping experience, from product discovery to checkout and order fulfillment.

Some feature are not updated yet like ,filter stripe and shipengine but I will do it as I learned.