

Day 4 – Dynamic Frontend Components

Shaista Jawed (00461445)

Report

Overview: This report outlines all the work accomplished in the project so far, highlighting key tasks, solutions to challenges, and the overall progress made.

Project Components

1. Sanity Integration:

- **Objective:** Use Sanity CMS as the backend to fetch and manage product data.
- **Tasks Accomplished:**
 - Created queries (postPathsQuery and postQuery) to fetch product paths and details.
 - Implemented sanityFetch and integrated it with the application for seamless product data fetching.
 - Successfully integrated product data into the pages and components.

2. Next.js and App Directory Setup:

- Utilized the app directory for routing and layout management.
- Implemented generateStaticParams for static generation of product pages.
- Ensured the routing is dynamic, allowing each product to have its own dedicated page.

3. Cart Functionality:

- **Objective:** Create a shopping cart system that works across all pages.
- **Implementation Details:**
 - Designed a CartContext using React Context API to manage cart state globally.
 - Added functions in the CartContext to handle adding, removing, and updating cart items.

- Utilized TypeScript to enforce type safety for cart items.
 - **Dynamic Quantity Display:**
 - Displayed the total number of items in the cart dynamically on the Navbar.
 - 4. **UI Components:**
 - Created and customized the following reusable components using Tailwind CSS:
 - **Navbar:** Integrated the shopping cart button with a dynamic item count.
 - **ProductDiscription Component:**
 - Added an “Add to Cart” button for individual product pages.
 - Implemented client-side logic for adding products to the cart.
 - **ShopContent:**
 - Displayed all products fetched from Sanity CMS.
 - Included a pagination system and a search bar for filtering products.
 - 5. **Shopping Cart Modal:**
 - Implemented a modal for the shopping cart that can be toggled from the Navbar.
 - The modal displays all cart items, including their names, quantities, and prices.
 - Added functionality to update and remove items directly from the modal.
 - 6. **Client-Side and Server-Side Code Separation:**
 - Ensured proper separation of server-side (generateStaticParams) and client-side (useCart) logic to avoid runtime conflicts.
 - Moved client-specific logic (e.g., cart actions) to client-side components.
-

Challenges and Solutions

1. **Challenge:** Conflicts between server-side and client-side logic.
 - **Solution:** Separated logic into server-side rendering (PostPage) and client-side components (ProductDiscription).

2. **Challenge:** Ensuring cart state is globally accessible.
 - **Solution:** Designed a CartContext to handle global state management.
 3. **Challenge:** Dynamic Navbar quantity update.
 - **Solution:** Used useCart hook to fetch the cart state and dynamically display the item count.
 4. **Challenge:** Styling inconsistencies.
 - **Solution:** Leveraged Tailwind CSS for responsive and consistent styling.
-

Key Accomplishments

1. Integrated Sanity CMS for product management and dynamic routing.
 2. Designed a shopping cart system with dynamic updates and a modal interface.
 3. Built reusable UI components with Tailwind CSS.
 4. Ensured proper TypeScript integration for type safety.
 5. Successfully managed and resolved client-server rendering conflicts.
-

Next Steps

1. **Testing:**
 - Perform rigorous testing to ensure the cart and other functionalities work seamlessly across pages and devices.
2. **Optimization:**
 - Optimize Sanity queries for faster data fetching.
 - Improve cart modal animations and transitions.
3. **Feature Expansion:**
 - Add functionality for applying discounts or coupon codes in the cart.
 - Enable user authentication and save cart state to the database.
4. **Deployment:**
 - Prepare the project for deployment on a production server.



Bold Nest

★★★★★ 10 reviews

\$1199.55

Availability: [in stock](#)

Welcome to BoldNest—where fearless design meets comfort and creativity. Crafted for those who embrace individuality and bold expressions, BoldNest is more than just a piece of furniture; it's a statement. With its striking design, exceptional comfort, and modern aesthetics, BoldNest is perfect for anyone looking to add a unique touch to their home or office. The BoldNest collection combines daring colors, dynamic shapes, and high-quality materials to create an environment where style and comfort coexist in perfect harmony. Whether it's a chair, sofa, or accent piece, each item is designed to stand out while offering a welcoming space for relaxation and enjoyment. Its sturdy construction and thoughtful design ensure that BoldNest not only makes a bold visual impact but also provides lasting durability. Perfect for those who are looking to break away from the ordinary and make their home a reflection of their bold personality, BoldNest is an ideal choice for creating an energetic and unique atmosphere. Key features: Bold and unique design that adds personality to any space. High-quality materials for comfort, durability, and style. Available in a variety of striking colors and patterns perfect for modern homes and offices that embrace creativity. Versatile design that complements contemporary and eclectic decor. Transform your home into a space that reflects your bold, unique style with BoldNest—where standout design and comfort meet.



Select Options

[Description](#) [Additional Information](#) [Reviews](#)



The Catalyzer

Blue bottle crucifix vinyl post-ironic four dollar toast vegan taxidermy. Gastropub indigo juice poutine, ramps microdosing banh mi pug VHS try-hard ugh iceland kickstarter tumblr live-edge tilde.

Blue bottle crucifix vinyl post-ironic four dollar toast vegan taxidermy. Gastropub indigo juice poutine, ramps microdosing.

Neptune

Blue bottle crucifix vinyl post-ironic four dollar toast vegan taxidermy. Gastropub indigo juice poutine, ramps microdosing banh mi pug VHS try-hard ugh iceland kickstarter tumblr live-edge tilde.

Blue bottle crucifix vinyl post-ironic four dollar toast vegan taxidermy. Gastropub indigo juice poutine, ramps microdosing live-edge.

Bandage

Company Info

[About Us](#)
[Career](#)
[We are hiring](#)
[Blog](#)

Legal

[About Us](#)
[Career](#)
[We are hiring](#)
[Blog](#)

Features

[About Us](#)
[Career](#)
[We are hiring](#)
[Blog](#)

Resources

[About Us](#)
[Career](#)
[We are hiring](#)
[Blog](#)



```

// components/Search.tsx
import React, { useState, useEffect } from "react";

interface SearchProps {
  onSearch: (query: string) => void;
}

const Search: React.FC<SearchProps> = ({ onSearch }) => {
  const [query, setQuery] = useState("");

  useEffect(() => {
    const handler = setTimeout(() => {
      onSearch(query);
    }, 300);

    return () => {
      clearTimeout(handler);
    };
  }, [query, onSearch]);

  return (
    <div className="relative">
      <input
        type="text"
        className="w-full px-4 py-2 border rounded-lg shadow-sm focus:outline-none focus:ring-2 focus:ring-blue-500 focus:border-transparent"
        placeholder="Search..."
        value={query}
        onChange={(e) => setQuery(e.target.value)}
      />
      <div className="absolute right-3 top-2">
        <svg
          xmlns="http://www.w3.org/2000/svg"
          className="w-5 h-5 text-gray-500"
          fill="none"
          viewBox="0 0 24 24"
          stroke="currentColor"
        >
          <path
            strokeLinecap="round"
            strokeLinejoin="round"
            strokeWidth={2}
            d="M21 21l-4.35-4.35m0 0a7.5 7.5 0 10-10.61 0 7.5 7.5 0 010.61 0z"
          />
        </svg>
      </div>
    </div>
  );
};

export default Search;

```

```
"use client";
```

```
import { SanityDocument } from "@sanity/client";  
import { PortableText } from "@portabletext/react";  
import { client } from "@sanity/lib/client";  
import imageUrlBuilder from "@sanity/image-url";
```

```
import Image from "next/image";
```

```
const builder = imageUrlBuilder(client);
```

```
const Post = ({ post }: { post: SanityDocument }) => {  
  return (  
    <main className="">  
      <h1>{post.title}</h1>  
      <Image src={builder.image(post.image_url).height(300).width(300).alt(post.title) />  
      {post?.body ? <PortableText value={post.body} /> : null}  
    </main>  
  );  
};
```

```
export default Post;
```