**Day 4 Activity Summary**

On Day 4, I focused on creating dynamic frontend components using Sanity CMS and APIs, emphasizing modular design and state management techniques for eﬃcient data handling. I also worked on implementing responsive design and following UX/UI best practices, preparing myself for real-world client projects with professional workﬂows.

**New Arrivals**

For the "New Arrivals" section, I utilized Groq queries to fetch real-time product data from Sanity CMS. This enabled me to display newly added items with their images, descriptions, and prices in a clean, user-friendly format, ensuring a seamless browsing experience.

**Top Selling**

For the "Top Selling" section, I utilized Groq queries to dynamically retrieve best-selling products from Sanity CMS. This allowed me to present popular items with their images, descriptions, and prices in an engaging and organized layout, enhancing the overall shopping experience.

**Top Brand:**

I created a custom Top Brands component to showcase renowned brands dynamically fetched from Sanity CMS. By using Groq queries, I displayed brand logos, descriptions, and their featured products in a visually appealing layout. This component ensures seamless updates and adds a professional touch to the website's branding section.

**Newsletter:**

I updated the Newsletter Subscription component to make it more interactive and functional. By incorporating an intuitive design and integrating it with the backend, users can now easily subscribe for updates. This enhancement ensures eﬃcient data capture and aligns with modern UX practices, making the component both engaging and impactful.

**Updating On Sale Product:**

For the On Sale section, I implemented a dynamic component that displays all products fetched from Sanity CMS. Using Groq queries, I ensured that the products, along with their sale prices, descriptions, and images, are displayed in an organized and engaging format. This approach allows easy updates and provides a user-friendly shopping experience.

**Update Product Card Details:**

I updated the Product Card Details functionality to provide a seamless and interactive shopping experience. Now, when users click on a product, they are directed to its detailed product page. The page dynamically fetches and displays product-speciﬁc information, including images, descriptions, and pricing, directly from Sanity. This enhancement ensures users can explore product details eﬀortlessly and make informed decisions.

**Update Proudct Details Tabs**

I updated the Product Card Details functionality to provide a seamless and interactive shopping experience. Now, when users click on a product, they are directed to its detailed product page. The page dynamically fetches and displays product-speciﬁc information, including images,

descriptions, and pricing. This enhancement ensures users can explore product details eﬀortlessly and make informed decisions.

**Update Add to Cart**

I improved the "Add to Cart" functionality to streamline the shopping experience. Now, when a user clicks on a product, it is automatically added to the cart. The cart is dynamically updated with the selected product, including details such as quantity and pricing. This enhancement ensures a smoother and more eﬃcient process for users to manage their selections and proceed with their purchase.

**Groq Querry to fetch data**

To fetch data from Sanity using a GROQ query, you would typically write a query in JavaScript (using Sanity's client), or in a framework like Next.js or Gatsby that integrates with Sanity. Here's an example of a GROQ query to fetch data from a Sanity

.