**Hackathon Day 4**

**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.

**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 Static**

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.

**Payment Functionality Static**

I have also updated the payment option functionality. Now, when you click on the "Checkout" button, it will seamlessly redirect you to a page displaying various payment options. This enhancement ensures a more streamlined and eﬃcient checkout process, allowing users to easily select their preferred payment method and complete their purchase without any hassle.

**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 dataset:

**Product Detail Coding**

I have coded the product detail page, fetching product data from Sanity using a GROQ query. The details, like name, description, price, and image, are displayed dynamically based on the product's unique "slug." I also added an "Add to Cart" button for a seamless shopping experience.