Day 04: Building Dynamic Frontend Components for Avion Marketplace

Overview

On Day 04 of the hackathon, we focused on enhancing the dynamic functionality of the frontend components for Avion, our marketplace. The key objectives were to improve the user experience by dynamically rendering product details, adding interactivity with the cart, and implementing a search functionality for seamless navigation. Below is a detailed account of the progress, challenges faced, and solutions implemented.

Objectives

1. Display all products dynamically fetched from Sanity CMS.
2. Navigate users to a detailed product page upon clicking a product.
3. Enhance the product detail page with a visually appealing layout and interactive features.
4. Implement "Add to Cart" functionality with dynamic alerts and quantity management.
5. Provide a search functionality to filter and navigate products.

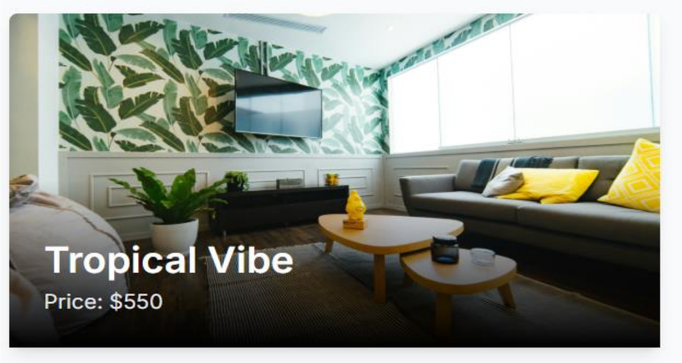
Process & Flow

1. Displaying Products on the Landing Page

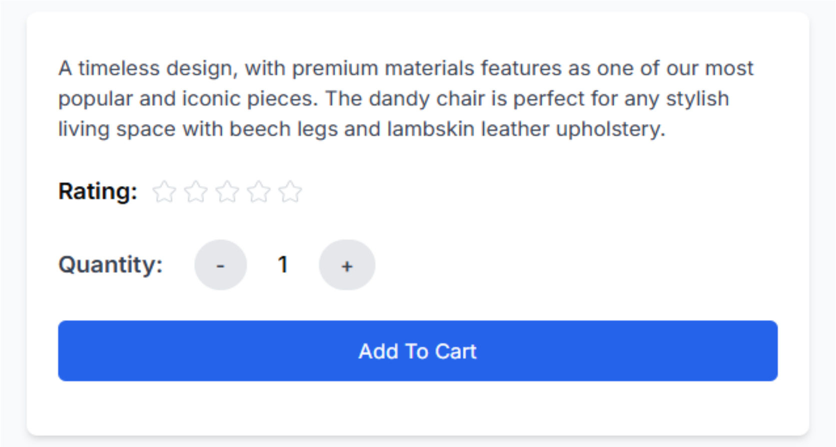
- Products were fetched from the Sanity CMS API.
- Data was dynamically mapped to create a responsive grid layout showing product images, names, and prices.

2. Product Detail Page

- Design:
  - Left Side: Displays the product image with an overlay containing the product name and price for visual appeal.



- Right Side: Includes the product description, discount percentage, quantity selector, and an "Add to Cart" button.

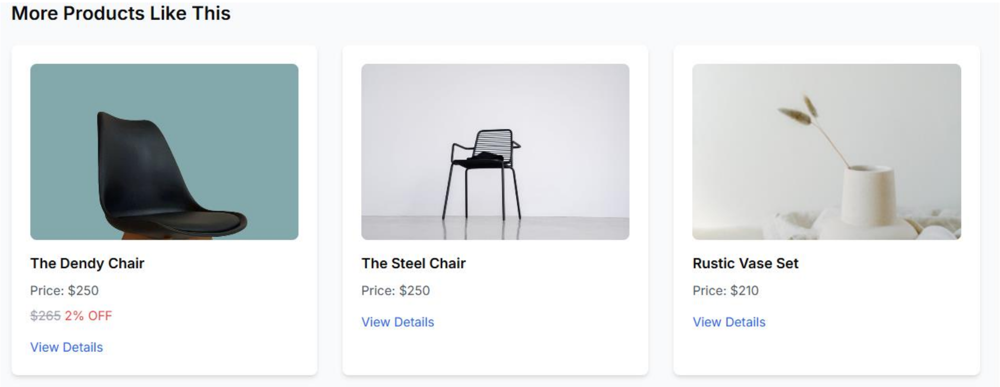


A timeless design, with premium materials features as one of our most popular and iconic pieces. The dandy chair is perfect for any stylish living space with beech legs and lambskin leather upholstery.

Rating: ☆☆☆☆☆

Quantity: 1

- Dynamic Features:
  - Clicking a product navigates the user to the product detail page using dynamic routes (slug-to-slug navigation).
  - Related products are displayed at the bottom of the page.



More Products Like This

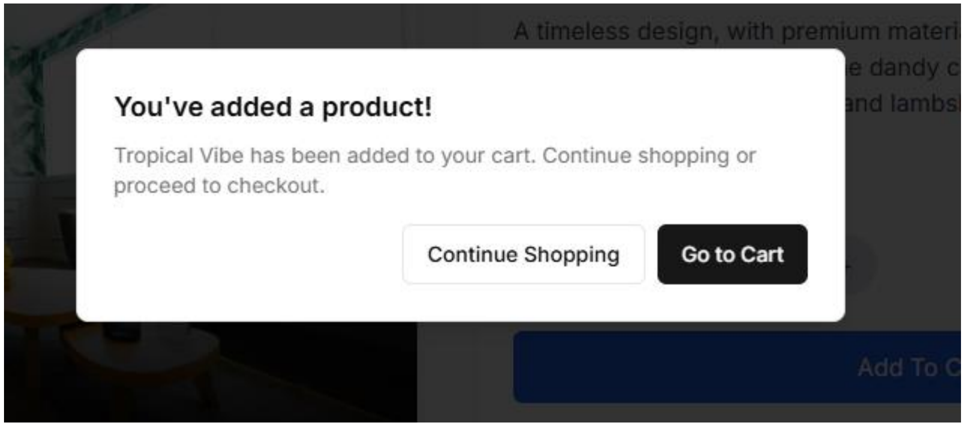
The Dendy Chair
Price: $250
~~$265~~ 2% OFF

The Steel Chair
Price: $250

Rustic Vase Set
Price: $210

3. Add to Cart Functionality

- When the user clicks "Add to Cart":
  - The product is added to the cart.
  - A dialog box (created using ShadCN UI components) confirms the addition with the message: “Your {product.name} has been added to the cart.”
  - Options in the dialog: Go to Cart or Continue Shopping.



A timeless design, with premium materi
e dandy c
and lambs

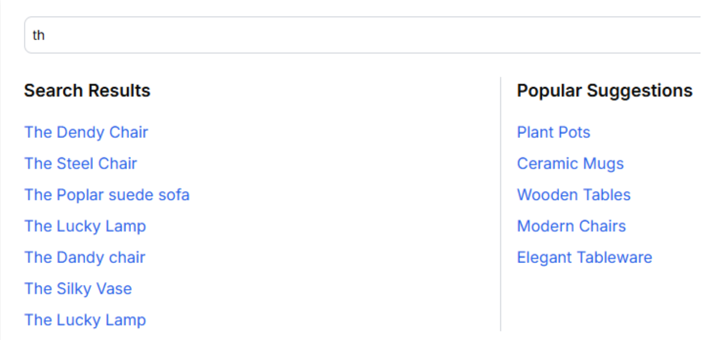
You've added a product!

Tropical Vibe has been added to your cart. Continue shopping or proceed to checkout.

- Quantity updates dynamically: If the user selects a quantity (e.g., 4 chairs), the cart reflects the updated total price and quantity.
- A "Remove All Items" button clears the cart instantly.

4. Search Functionality

- A search bar was added to allow users to filter products by name.
- Implemented a query in Sanity CMS to retrieve matching products.



th

Search Results

- The Dendy Chair
- The Steel Chair
- The Poplar suede sofa
- The Lucky Lamp
- The Dandy chair
- The Silky Vase
- The Lucky Lamp

Popular Suggestions

- Plant Pots
- Ceramic Mugs
- Wooden Tables
- Modern Chairs
- Elegant Tableware

- On clicking a search result, the user is navigated to the product detail pa

Objectives

1. Display all products dynamically fetched from Sanity CMS.
2. Navigate users to a detailed product page upon clicking a product.
3. Enhance the product detail page with a visually appealing layout and interactive features.
4. Implement "Add to Cart" functionality with dynamic alerts and quantity management.
5. Provide a search functionality to filter and navigate products.

Libraries & Tools Used

- Sanity CMS: Backend and database for managing product data.
- Next.js: Framework for building the frontend and dynamic routing.
- Tailwind CSS: For styling and creating responsive layouts.
- ShadCN UI: Components for creating the alert dialog box.

Challenges Faced

1. Search Functionality:
    - Initial confusion about querying product data from Sanity CMS.
    - Solution: Researched Sanity’s GROQ querying system to filter data effectively.
2. Slug-to-Slug Navigation:
    - Difficulty in implementing dynamic routing for individual product pages.
    - Solution: Utilized Next.js dynamic routes and ensured data consistency in Sanity CMS.

Conclusion

Day 04 demonstrated our ability to tackle complex frontend challenges and deliver a seamless user experience. The progress achieved not only enhanced the marketplace's functionality but also prepared us for future improvements. By addressing critical challenges, such as dynamic routing and search integration.

FOLLOW ME ON