

Day 3 – API Integration Report – Furniro

Table of Context:

- API integration process
- Adjustments made to schemas
- Screenshots of:
 1. API calls.
 2. Data successfully displayed in the frontend.
 3. Populated Sanity CMS fields.
- Code snippets for API integration and migration scripts

1. API integration process:

Overview

This report documents the process of integrating APIs for fetching and displaying product data in a Next.js application using Sanity CMS. The data was manually added to Sanity for better control and customization, ensuring alignment with project requirements. The integration involves querying data using GROQ, serving it through API routes, and consuming it in React components using useEffect.

1. Data Management in Sanity

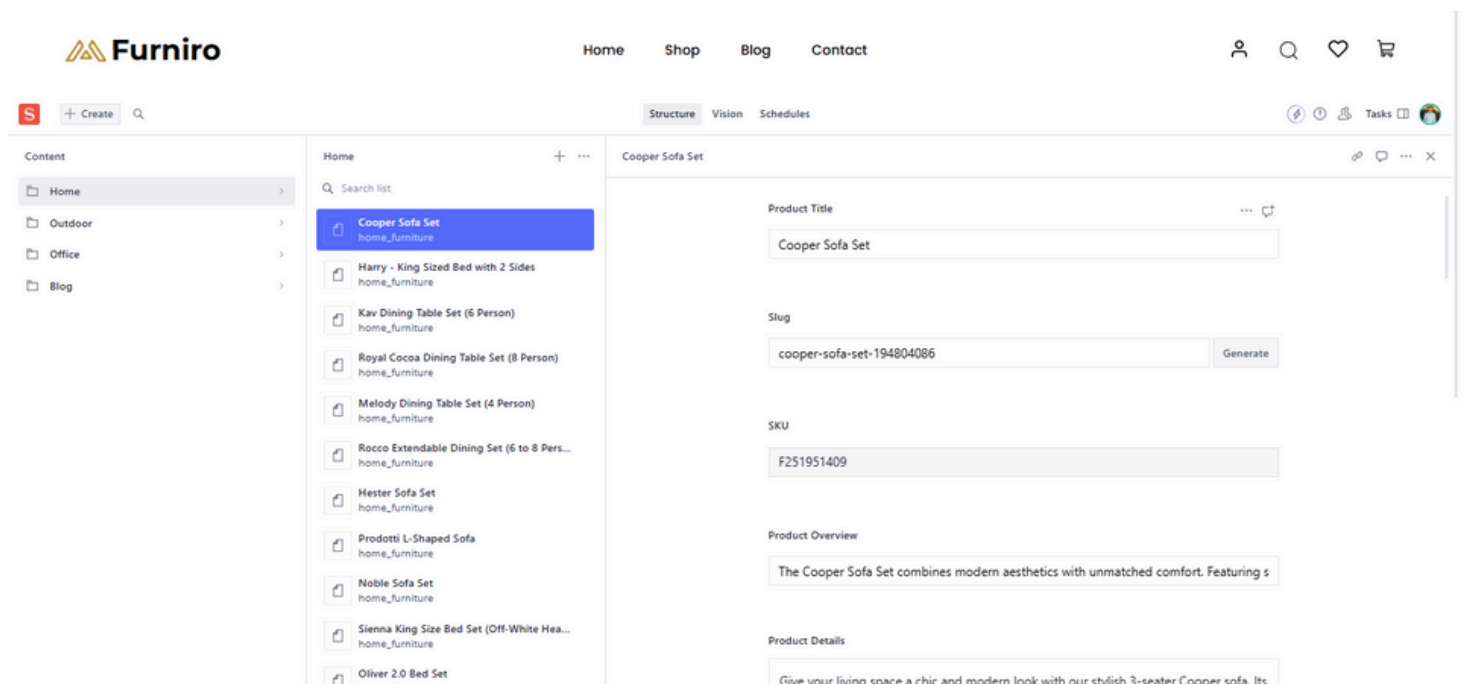
Data was manually added to Sanity Studio to meet specific requirements. This approach provided the flexibility to organize the data precisely as needed. Categories and subcategories of products included:

Categories:

- home_furniture
 - Subcategories: Bed, Sofa, Dining table
- Office
 - Subcategories: Office Chair, Meeting table, Bookshelves
- Outdoor
 - Subcategories: Outdoor Chair, Coffee table, Swings
- Blog

Tags:

- New Arrival
- Special Offer
- Featured
- Top trending

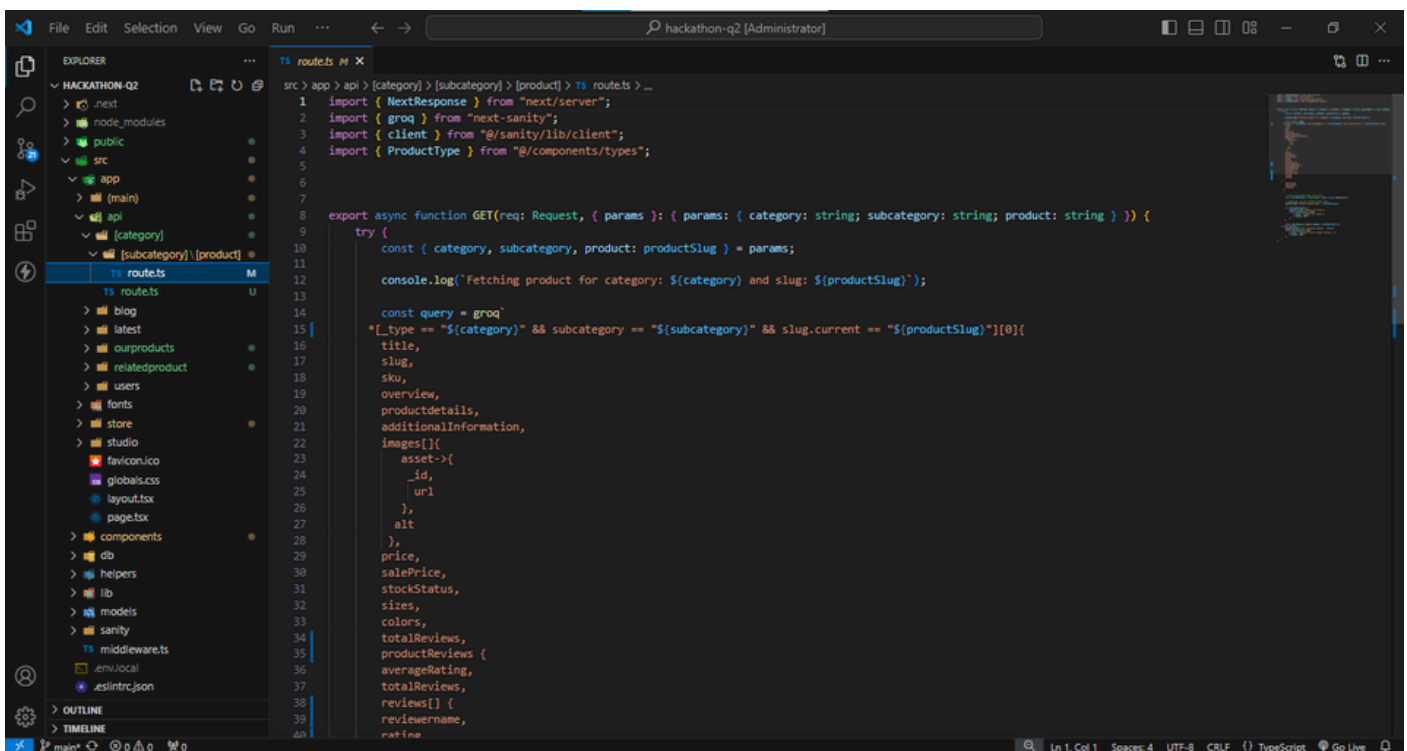


2. API Routes

Custom API routes were created in the Next.js application to fetch data from Sanity CMS. The routes serve as endpoints for querying specific product categories, subcategories, or individual product details.

Example API Endpoints:

1. All latest products:
`/api/products/latest`
2. Category and Subcategory Based:
`/api/products/[category]/[subcategory]`
3. Single product detail page:
`/api/products/[category]/[subcategory]/[product]`
4. Tagged Products: `/api/products/[tag]`



The screenshot shows a VS Code editor window with a file explorer on the left and a code editor on the right. The file explorer shows a project structure with folders like 'src', 'app', 'api', 'components', 'db', 'helpers', 'lib', 'models', 'sanity', 'middleware', and 'utils'. The 'api' folder is expanded, showing 'category', 'subcategory', and 'product' subfolders. The 'product' folder is selected, and the 'routes.ts' file is open in the editor. The code in 'routes.ts' defines a GET endpoint for fetching product details based on category, subcategory, and product slug. It uses Next.js's `getServerSideProps` and `getStaticProps` functions to fetch data from a Sanity CMS. The code is as follows:

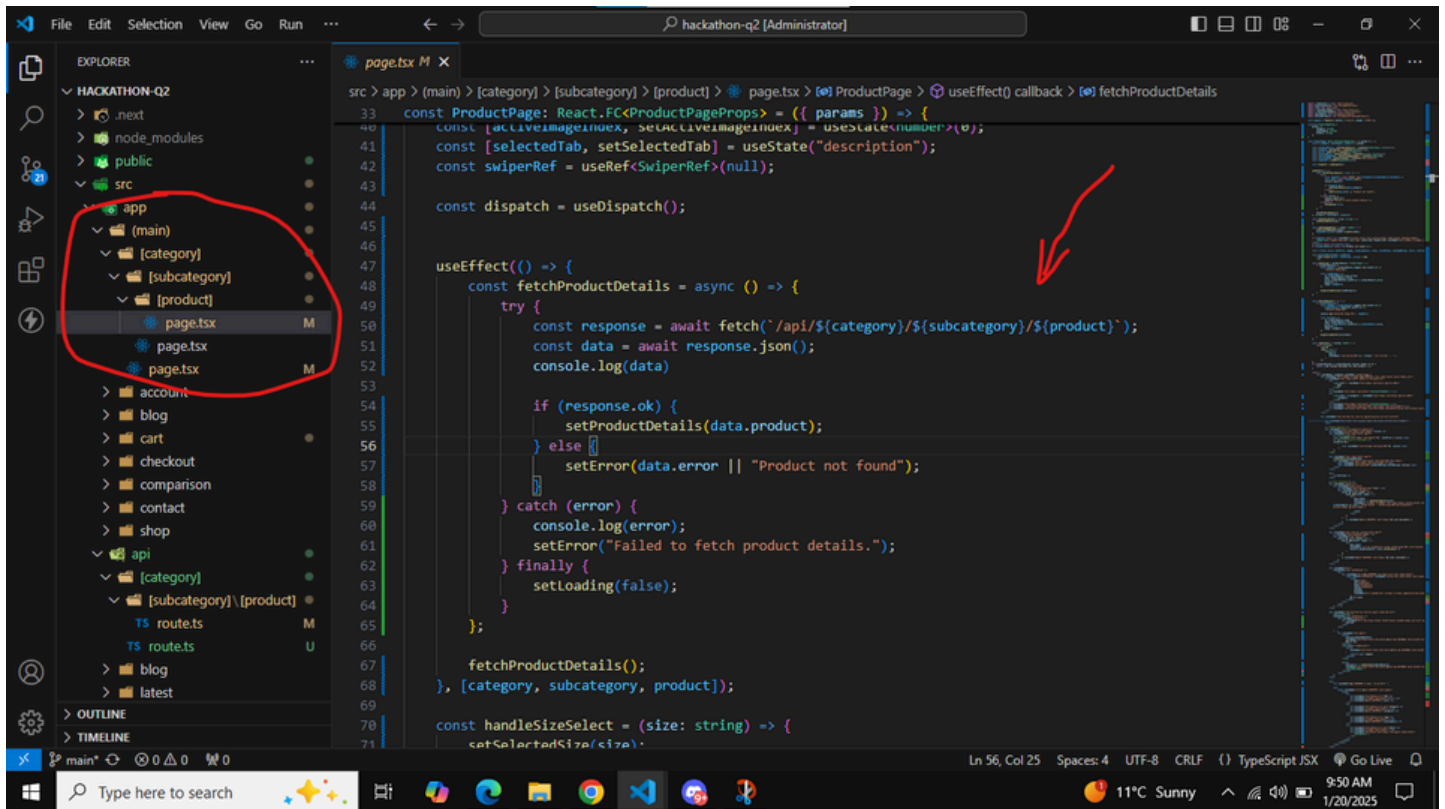
```
src > app > api > [category] > [subcategory] > [product] > TS routes > ...
1 import { NextResponse } from "next/server";
2 import { groq } from "next-sanity";
3 import { client } from "@sanity/lib/client";
4 import { ProductType } from "@components/types";
5
6
7
8
9
10 export async function GET(req: Request, { params }: { params: { category: string; subcategory: string; product: string } }) {
11   try {
12     const { category, subcategory, product: productSlug } = params;
13     console.log(`Fetching product for category: ${category} and slug: ${productSlug}`);
14
15     const query = groq`
16       *[_type == "${category}" && subcategory == "${subcategory}" && slug.current == "${productSlug}"][0]{
17         title,
18         slug,
19         sku,
20         overview,
21         productdetails,
22         additionalInformation,
23         images:[{
24           asset->{
25             _id,
26             url
27           },
28           alt
29         ]},
30         price,
31         salePrice,
32         stockStatus,
33         sizes,
34         colors,
35         totalReviews,
36         productReviews {
37           averageRating,
38           totalReviews,
39           reviews[] {
40             reviewername,
41             rating
42           }
43         }
44       `;
```

Fetching Data in Components:

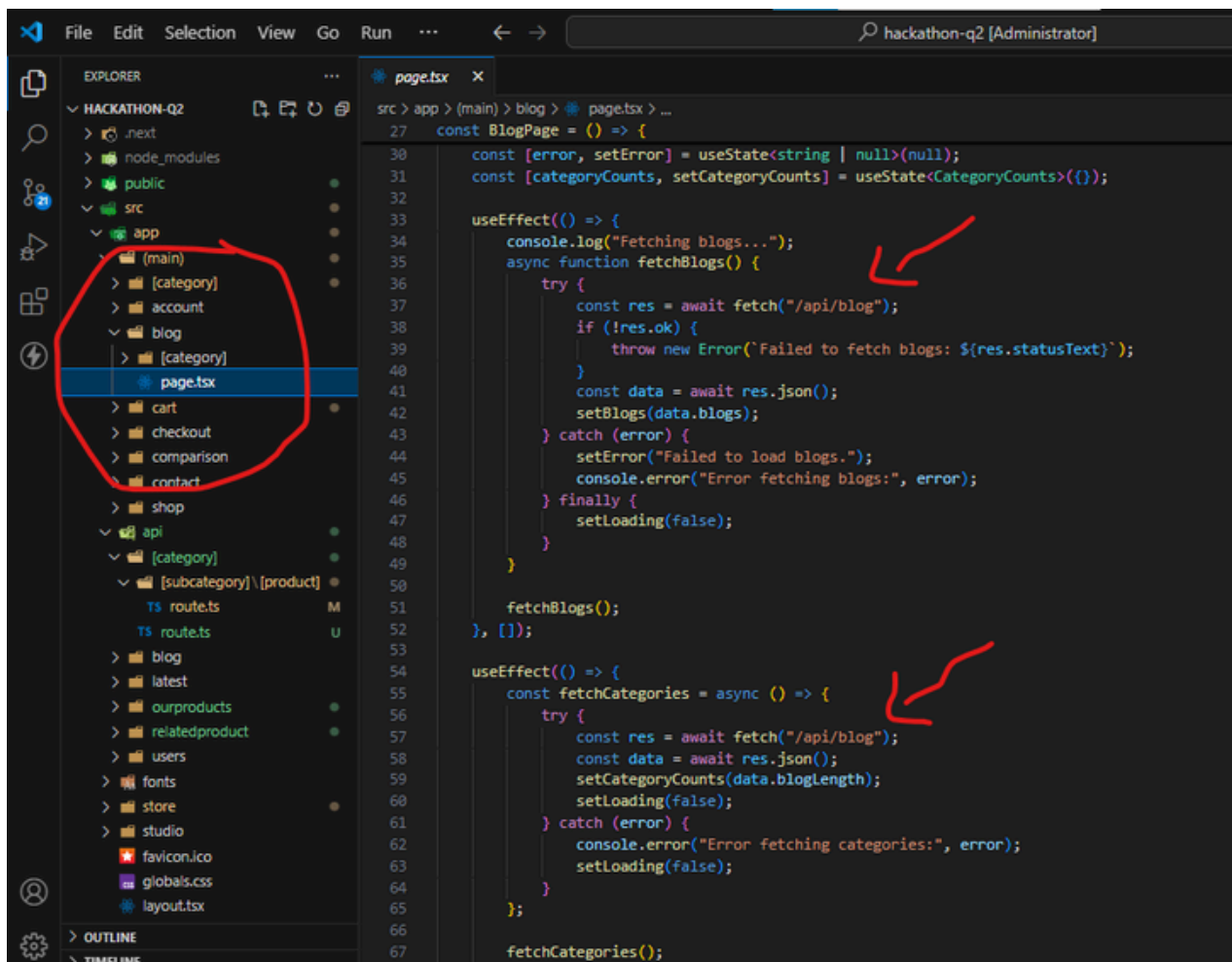
The useEffect hook is used in React components to fetch data from the API endpoints. Data is dynamically retrieved and displayed to the user. This approach ensures the UI updates automatically when the data changes.

Step-by-Step Explanation

- Initialize State: Create state variables for storing products and loading status.
- Define Fetch Function: Use fetch to call the API endpoint and retrieve the data.
- Handle Loading and Errors: Display loading indicators while data is being fetched and handle any errors that occur.
- Update State: Populate the state with the fetched data to render it in the UI.



and



API calls:

I called api from api routes , here you see how i called apis in frontend components and pages

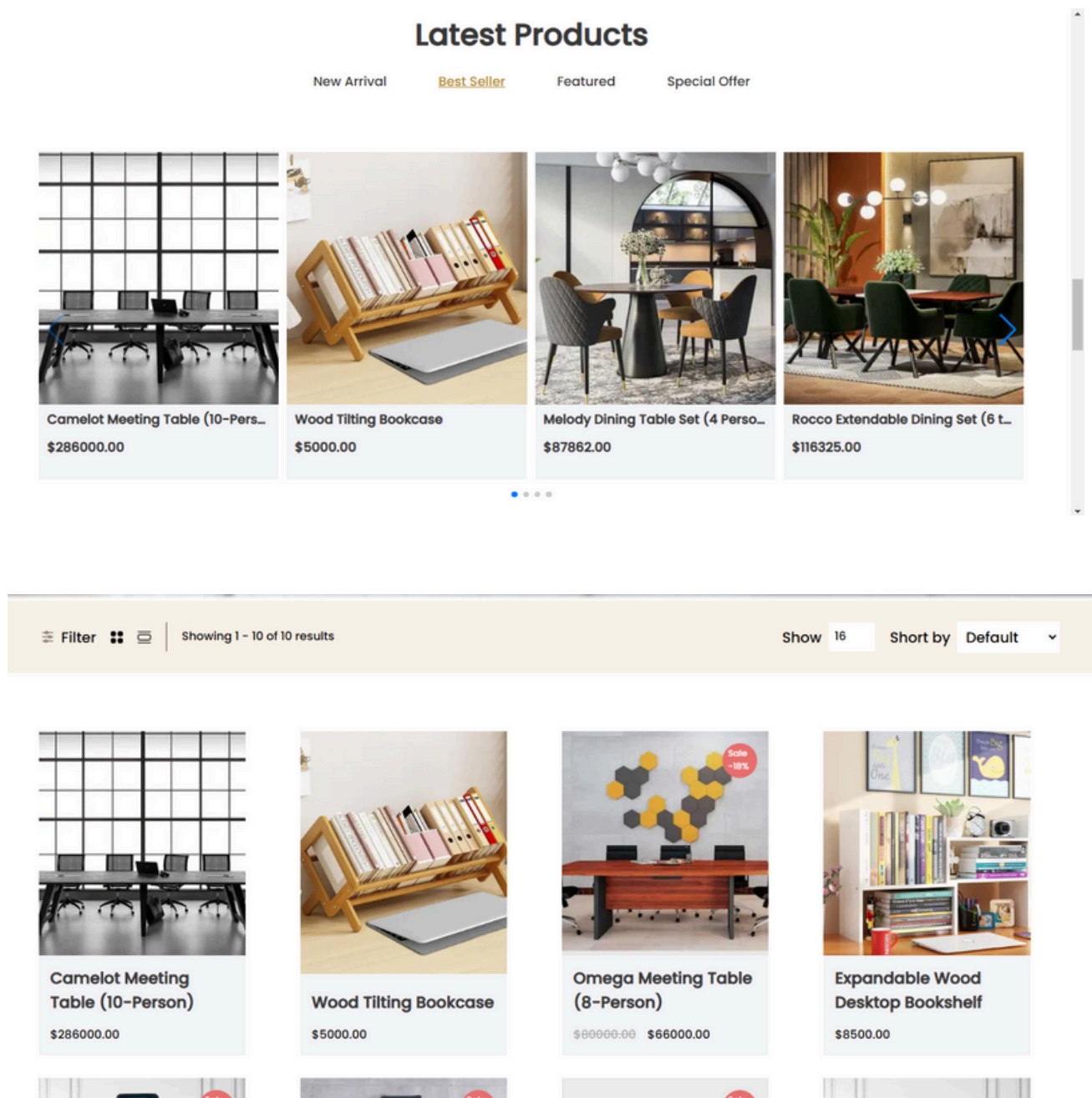
app/blog/[category]/[slug]

```
1  const fetchData = async () => {
2    try {
3      const res = await fetch(`/api/blog/${params.category}/${params.slug}`, { cache: "no-store" });
4      if (!res.ok) {
5        throw new Error("Failed to fetch the blog data");
6      }
7      const data = await res.json();
8      if (data.blog) {
9        setBlog(data.blog);
10     } else {
11       setError("Blog not found.");
12     }
13   } catch (err) {
14     setError("An error occurred.");
15     console.log(err)
16   } finally {
17     setLoading(false);
18   }
19 };
20
21 fetchData();
22 }, [params.category, params.slug]);
23
24 useEffect(() => {
25   if (!params.category || !params.slug) return;
26
27   const recentPost = async () => {
28     try {
29       const res = await fetch(`/api/blog/${params.category}/${params.slug}`, { cache: "no-store" });
30       if (!res.ok) {
31         throw new Error("Failed to fetch blog data");
32       }
33
34       const data = await res.json();
35
36       if (data.blog) {
37         setBlog(data.blog);
38       } else {
39         setError("Blog not found.");
40       }
41
42       if (data.recentPosts) {
43         setRecentPosts(data.recentPosts);
44       }
45     } catch (err: unknown) {
46       if (err instanceof Error) {
47         setError(err.message);
48       } else {
49         setError("An unknown error occurred.");
50       }
51     } finally {
52       setLoading(false);
53     }
54   };
55
56   recentPost();
57 }, [params.category, params.slug]);
58
59 useEffect(() => {
60   const fetchCategories = async () => {
61     try {
62       const res = await fetch("/api/blog");
63       const data = await res.json();
64       setCategoryCounts(data.blogLength);
65       setLoading(false);
66     } catch (error) {
67       console.error("Error fetching categories:", error);
68       setLoading(false);
69     }
70   };
71
72   fetchCategories();
73 }, []);
```

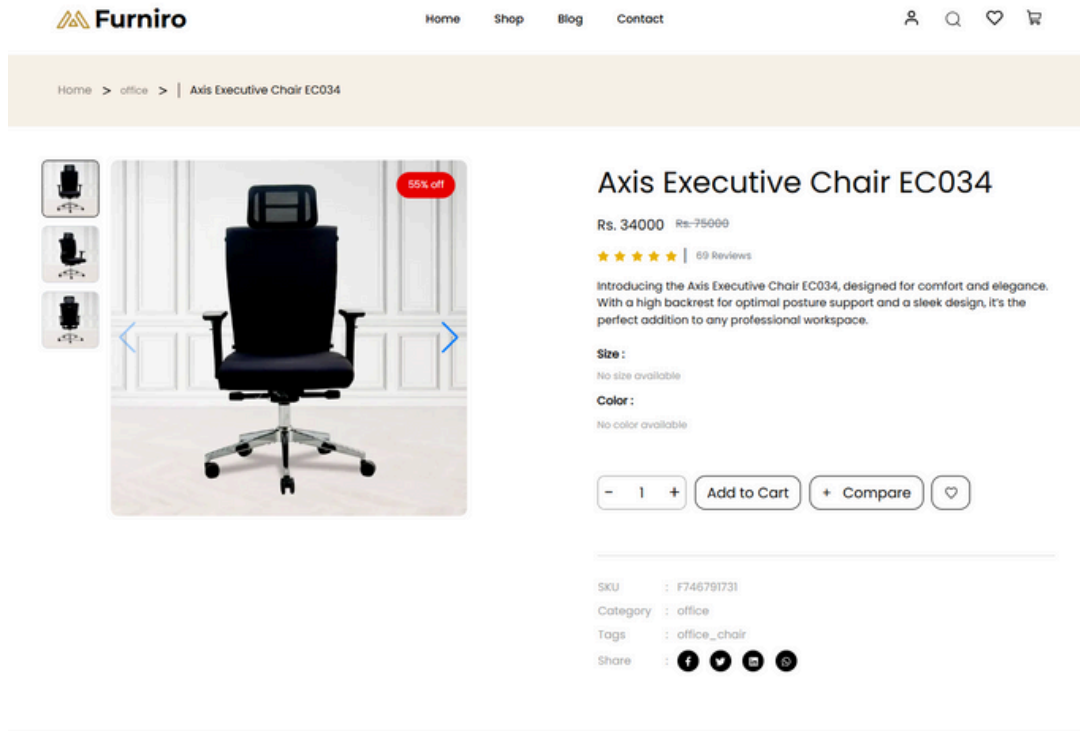
Data successfully displayed in the frontend:

After integrating and processing the data from the API through API routes, the next key objective was to ensure that the data was successfully displayed in the frontend of the application.

This step is critical to ensure that the user interface (UI) presents accurate and up-to-date information from the backend or external sources.



Product page

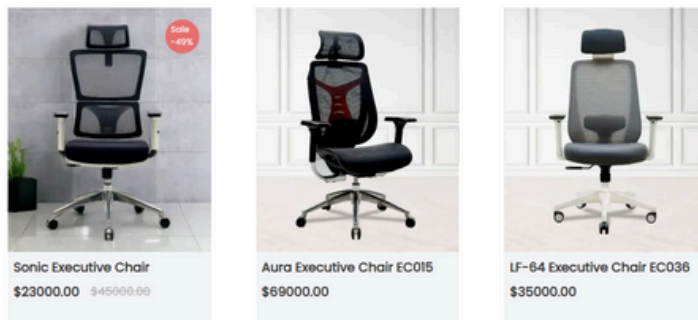


Description	Additional Information	Reviews [3]
-------------	------------------------	-------------

The Axis Executive Chair EC034 combines ergonomic design with sleek aesthetics, offering superior comfort and support for long hours of work. Featuring adjustable height, a high backrest, and padded armrests, it's crafted with premium materials for durability. Perfect for home offices or corporate setups, this chair ensures a professional and comfortable seating experience.



Related Products



Funiro.

400 University Drive Suite 200 Coral
Gables, FL 33134 USA

Links

[Home](#)[Shop](#)

About

Contact

Help

Payment Options

Returns

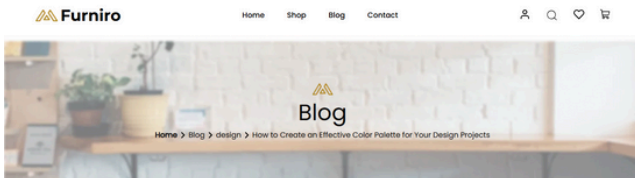
[Privacy Policies](#)

Newsletter

Enter Your Email Address

SUBSCRIBE

Blog



admin 1/6/2025 design

How to Create an Effective Color Palette for Your Design Projects

Color plays a central role in design, and selecting the right palette can elevate your projects from average to extraordinary. Colors are more than just aesthetic choices—they evoke emotions, reflect a brand's identity, and drive user behavior. A well-chosen color palette creates a harmonious visual experience that guides users through the interface. This blog will take you through the process of choosing an effective color palette, grounded in color theory, the psychology of colors, and practical tips for pairing and testing colors. Whether you're working on a website, mobile app, or logo, mastering color selection will ensure that your designs are not only visually stunning but also purposeful and functional.

In design, color is more than a decorative feature—it's a powerful tool that influences emotions, actions, and perceptions. The right color palette can convey your message more clearly, set the tone for your brand, and guide user behavior. As renowned color expert Leatrice Eiseman stated,

"Color is the language of the emotions."

Colors can stimulate feelings of excitement, calmness, trust, or urgency. When you choose the right colors, your design can evoke a particular mood or call users to action. This is why selecting an effective color palette is crucial in web design, branding, and marketing materials.

1. Understand Color Theory

Color theory is the foundation of a good color palette. It explains how colors interact with each other, and how you can mix primary, secondary, and tertiary colors to create visually appealing combinations. Familiarizing yourself with the color wheel is a good starting point, as it helps you understand complementary, analogous, and triadic color schemes.

Complementary Colors: Colors that are opposite on the wheel, like red and green, create contrast and excitement.

Analogous Colors: Colors that are next to each other, like blue, teal, and green, create harmony and calmness.

Triadic Colors: Three colors evenly spaced around the wheel, like red, yellow, and blue, create vibrant combinations.

2. Psychology of Colors

Colors communicate more than just their hue—they also invoke feelings and emotions. For instance:

Blue is often associated with trust, professionalism, and calmness. It's widely used in corporate branding and health-related sites.

Red is energetic, urgent, and stimulating. It's commonly used for call-to-action buttons.

Yellow conveys positivity and energy, but when overused, it can cause visual fatigue.

When designing, it's important to think about what each color communicates and how it aligns with your brand's message and the user experience.

3. Contrast and Accessibility

Effective color choices also account for contrast, ensuring that the text is readable against the background. High contrast between text and background helps ensure that your design is accessible to users with vision impairments. Using tools like contrast checkers can help you identify areas where your color choices may be difficult to read.

4. Consistency Across Platforms

Consistency is key in branding. Your color palette should remain the same across all mediums—websites, print materials, social media, and even packaging. Having a unified color scheme strengthens your brand's identity and creates a cohesive visual experience for users.

5. Testing and Iteration

Finally, once you've chosen a color palette, test it across different devices and lighting conditions. Colors can appear differently depending on the screen type or ambient lighting, so it's essential to test your palette on various screens.

"Colors, like features, follow the changes of the emotions." - Pablo Picasso

Categories

design	4
wood	2
crafts	4
interior	4
handmade	3

Recent Posts

- The Power of Minimalism in Modern Design 1/6/2025
- How to Choose the Perfect Sofa for Your Living Room 1/6/2025
- How to Incorporate Scandinavian Furniture into Your Home 1/6/2025

Furniro

Home Shop Blog Contact

Search icons

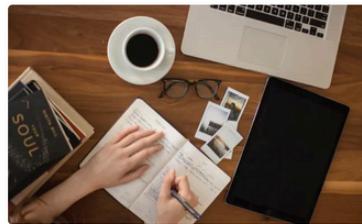


admin 1/6/2025 design

How to Create an Effective Color Palette for Your Design Projects

Color plays a central role in design, and selecting the right palette can elevate your projects from average to extraordinary. Colors are more than just aesthetic choices—they evoke emotions, reflect a brand's identity, and drive user behavior. A well-chosen color palette creates a harmonious visual experience that guides users through the interface. This blog will take you through the process of choosing an effective color palette, grounded in color theory, the psychology of colors, and practical tips for pairing and testing colors. Whether you're working on a website, mobile app, or logo, mastering color selection will ensure that your designs are not only visually stunning but also purposeful and functional.

Read more



admin 1/7/2025 wood

Going all-in with millennial design

Millennial design has transformed the way we approach furniture and home decor. This generation's influence has pushed the boundaries of traditional design, blending functionality, aesthetics, and sustainability seamlessly. At its core, millennial design values minimalism, natural elements, and personalized spaces, making it a perfect match for wooden furniture.

Read more



Indiqueir Rahman 1/7/2025 crafts

Embracing the Art of Crafts – Adding a Personal Touch to Your Space

Crafts are more than just decorative pieces—they are stories woven into every thread, stroke, and carving. They embody culture, tradition, and creativity, offering a unique way to personalize your space. In a world of mass production, handcrafted items stand out for their authenticity, charm, and intricate attention to detail. This blog explores the timeless appeal of crafts and their role in modern interior design, showing how these unique creations can add warmth, personality, and artistry to your home.