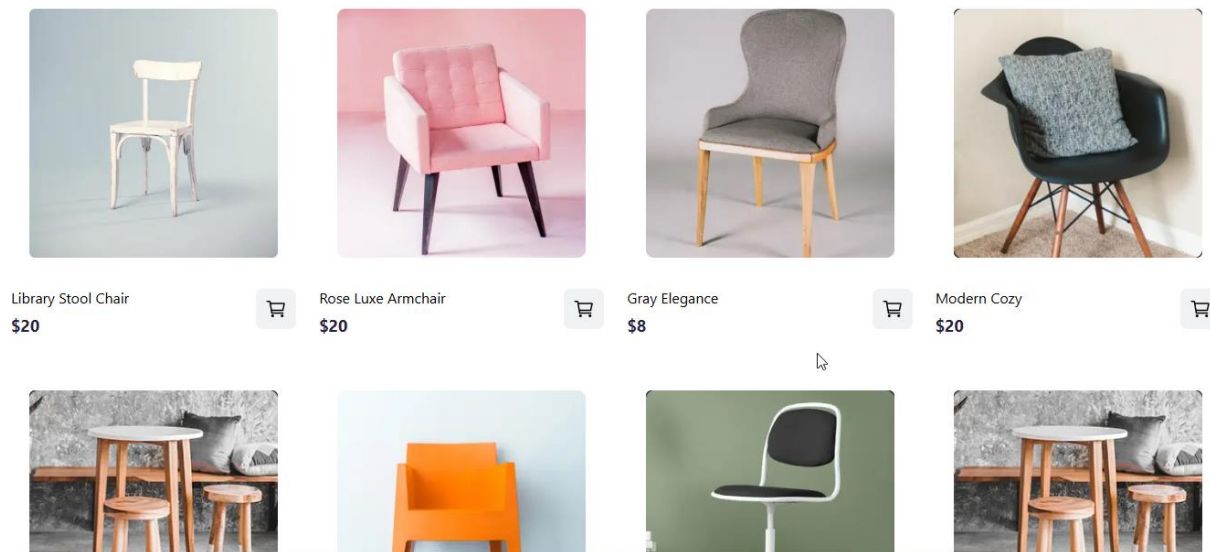


# DAY 5 - TESTING, ERROR HANDLING, AND BACKEND INTEGRATION REFINEMENT

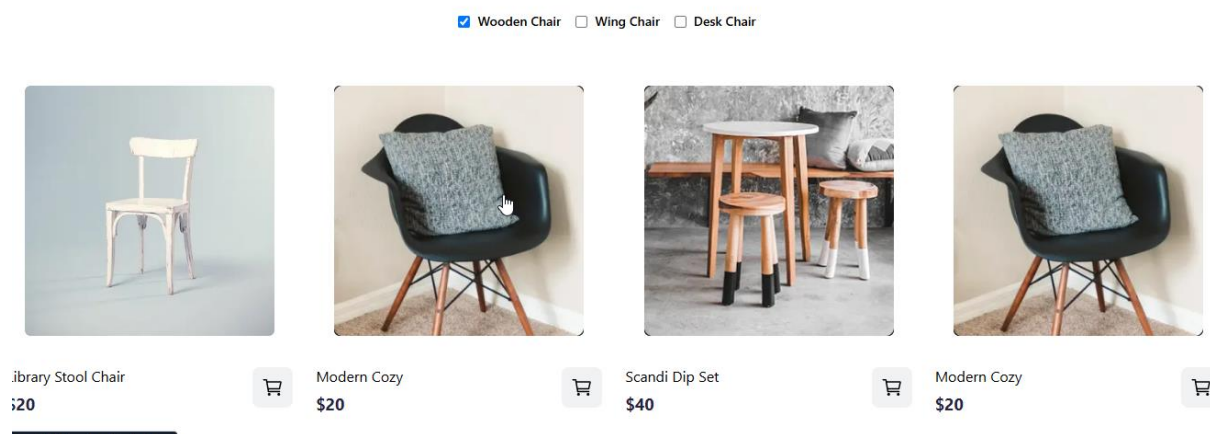
Product listing :

All product successfully display on front-end



Filters and search :

Category filter work perfectly



Cart operations :

Add to cart functionality work perfectly



Modern Cozy

\$20



Modern Cozy

\$20

Quantity 1



Buy Now

Clear Cart

Dynamic routing:

All product are dynamic using their unique product id



## Rose Luxe Armchair

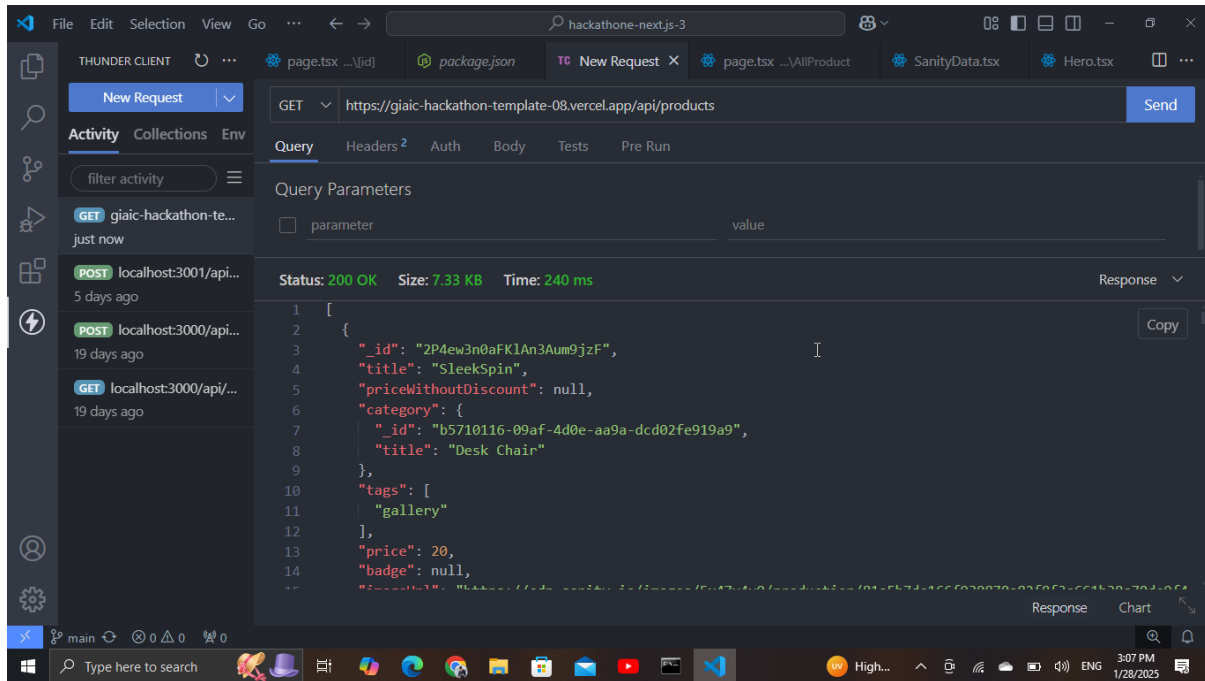
\$20 USD

Lorem ipsum dolor sit amet, consectetur adipiscing elit.  
Nullam tincidunt erat enim. Lorem ipsum dolor sit amet,  
consectetur adipiscing

 Add To Cart

API response testing:

API successfully work



Add Error Messages:

Using try and catch method for fetch the data and show the error

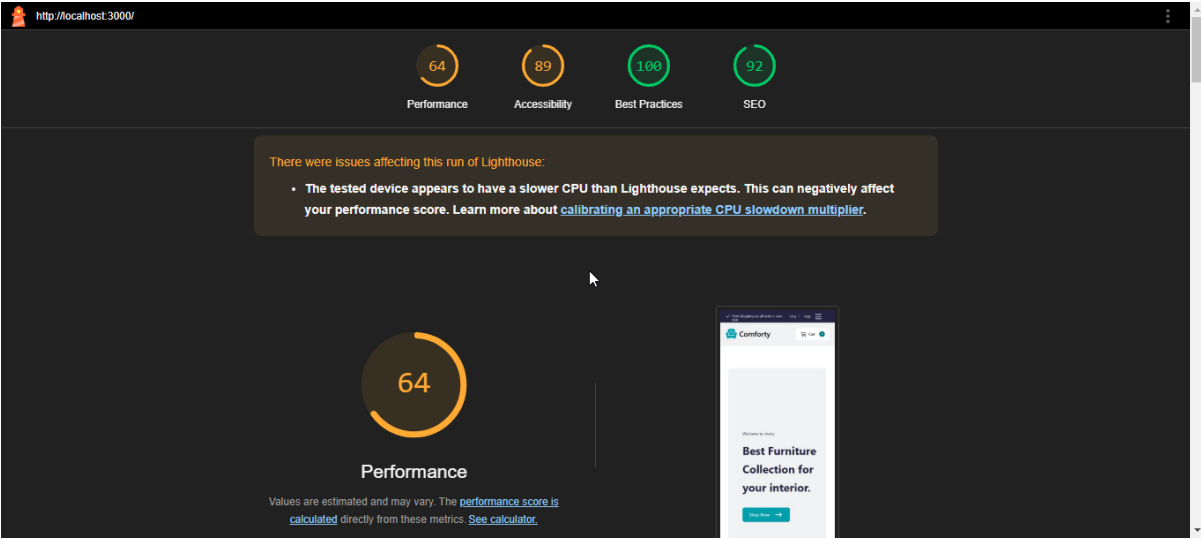
```
const SanityData = () => {
  const { addToCart } = useCart();
  const [products, setProducts] = useState<SanityData[]>([]); // State

  useEffect(() => {
    const fetchProducts = async () => {
      const productTitles = [
        "Rose Luxe Armchair",
        "Citrus Edge",
        "Ivory Charm",
        "Library Stool Chair",
      ];

      try {
        // Fetch products from Sanity
        const fetchData: SanityData[] = await client.fetch(
          `*[_type == 'products' && title in $titles]{
            id,
            title,
            price,
            priceWithoutDiscount,
            badge,
            image,
            category->,
            description,
            inventory,
            tags
          }`,
          { titles: productTitles }
        );
        setProducts(fetchData); // Update state with fetched products
      } catch (error) {
        console.error("Error fetching products:", error);
      }
    };

    fetchProducts();
  }, []); // Empty dependency array ensures this runs only once
}
```

Use Lighthouse to identify speed and performance issues.



```
LH:status Auditing: Avoids requesting the geolocation permission on page load +10ms
LH:status Auditing: No issues in the 'Issues' panel in Chrome Devtools +5ms
LH:status Auditing: Avoids `document.write()` +4ms
LH:status Auditing: Detected JavaScript libraries +5ms
LH:status Auditing: Avoids requesting the notification permission on page load +3ms
LH:status Auditing: Allows users to paste into input fields +4ms
LH:status Auditing: Use HTTP/2 +5ms
LH:status Auditing: Uses passive listeners to improve scrolling performance +7ms
LH:status Auditing: Document has a meta description +6ms
LH:status Auditing: Page has successful HTTP status code +7ms
LH:status Auditing: Document uses legible font sizes +4ms
LH:status Auditing: Links have descriptive text +15ms
LH:status Auditing: Links are crawlable +4ms
LH:status Auditing: Page isn't blocked from indexing +5ms
LH:status Auditing: robots.txt is valid +6ms
LH:status Auditing: Document has a valid `hreflang` +7ms
LH:status Auditing: Document has a valid `rel=canonical` +4ms
LH:status Auditing: Structured data is valid +5ms
LH:status Auditing: Page didn't prevent back/forward cache restoration +3ms
LH:status Generating results... +7ms
LH:Printer html output written to D:\Work 1\1 QUTER 2 ASSIGNMENT\hackathone-next.js-3\localhost_2025-01-28_15-33-02.report.html +111ms
LH:ChromeLauncher Killing Chrome instance 936 +2ms
PS D:\Work 1\1 QUTER 2 ASSIGNMENT\hackathone-next.js-3> 
```

Test Case ID	Test Case Description	Test Step	Expected Result	Actual Result	Status	Remarks

1	Validate product listing page	Ensure responsiveness on mobile	Products displayed correctly	Products displayed correctly	Passed	Work Perfect fully
2	Test API error handling	Disconnect API> Refresh page	Show fallback UI with error message	Error message shown	Passed	No issue found
3	Check cart functionality	Add product to cart > Verify cart contents	Cart updates with added product	Cart updates as expected	Passed	Works as expected
4	Ensure responsiveness on mobile	Resize browser window > Check layout	Layout adjusts properly to screen size	Responsive layout working as intended	Passed	Test successful