

Day 4 - Dynamic Frontend Components - [Meubel House]

Date: 19-01-2025

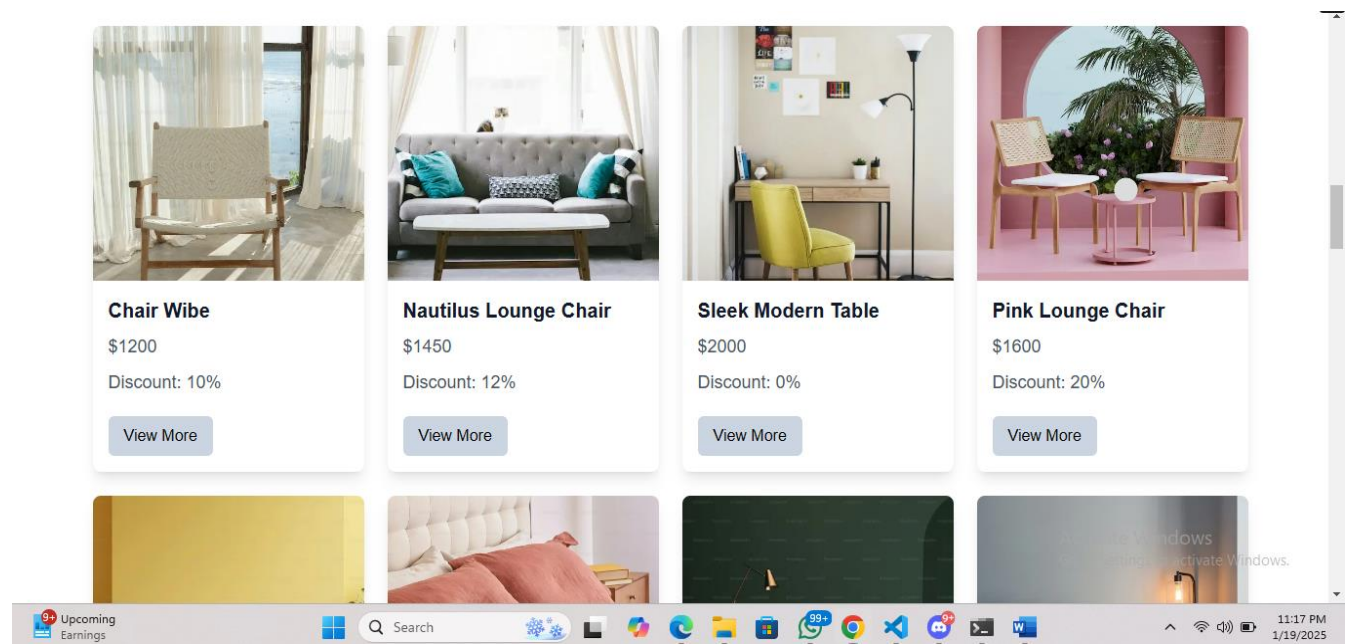
Accomplishments for “Day 4”:

1. **Product Display Enhancements:**
2. **Sanity CMS Integration:**
3. **Dynamic Routing**
4. **Filter Functionality Implementation:**
5. **Top Picks functionality**

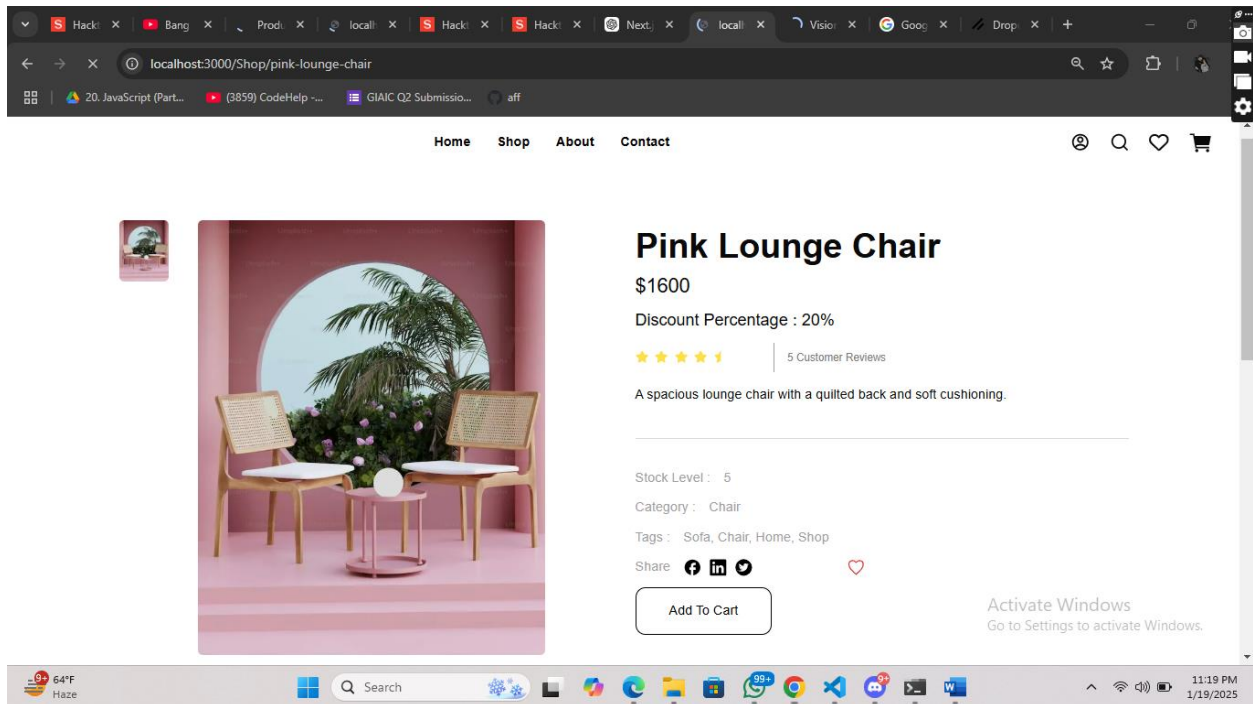
Functional Deliverables:

♣ Screenshots showcasing:

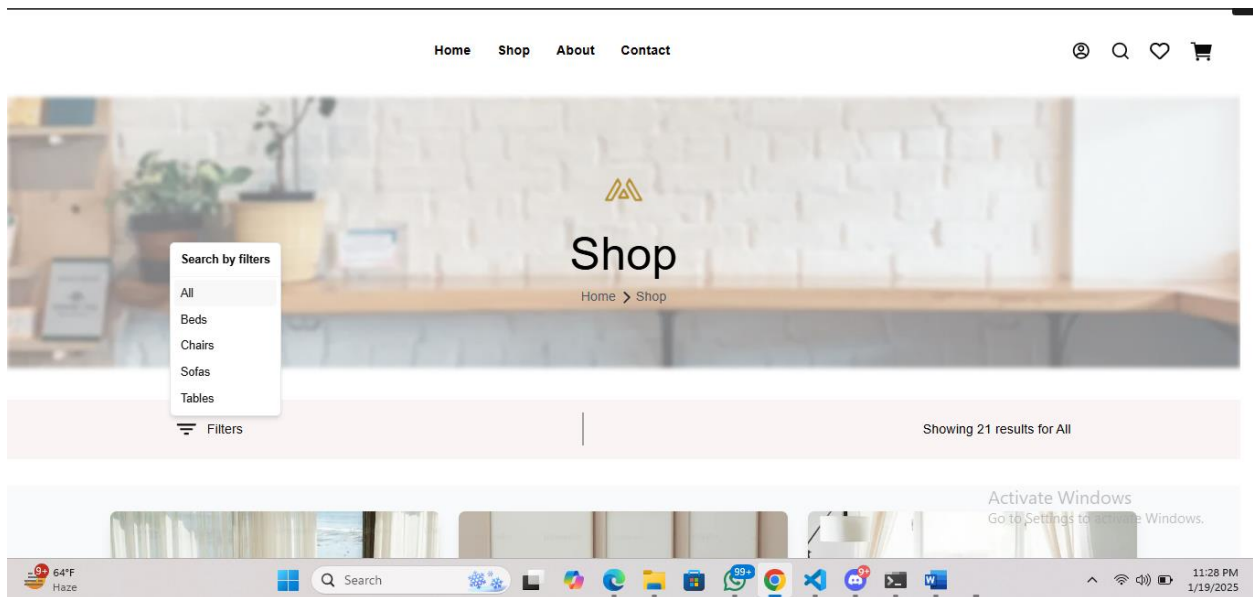
1. The product listing page with dynamic data.



2. Individual product detail pages with accurate routing and data rendering.



3. Working category filters.



♣ Code Deliverables:

1. Code snippets for key components.

For dynamic data

For filter code snippet:

```
1  "use client";
2
3  import { client } from "@sanity/lib/client";
4  import React, { useState, useEffect } from "react";
5  import Link from "next/link";
6  import Image from "next/image";
7  import { ToFilter } from "react-icons/105";
8  import {
9    DropdownMenu,
10    DropdownMenuContent,
11    DropdownMenuItem,
12    DropdownMenuLabel,
13    DropdownMenuSeparator,
14    DropdownMenuTrigger,
15  } from "@components/ui/dropdown-menu";
16
17  const FilterSection = () => {
18    const [allProducts, setAllProducts] = useState([]); // Store all products
19    const [products, setProducts] = useState([]); // Store filtered products
20    const [filter, setFilter] = useState<string>("All"); // Store selected filter
21
22    // Fetch products from Sanity
23    useEffect(() => {
24      const fetchProducts = async () => {
25        const query = `*[ _type == "product" ] {
26          id, name, "imageUrl": image.asset->url,
27          price, description, discountPercentage,
28          isFeaturedProduct, stockLevel, category,
29          "slug": slug.current
30        }`;
31        const fetchedProducts = await client.fetch(query);
32        setAllProducts(fetchedProducts);
33        setProducts(fetchedProducts); // Show all products by default
34      };
35      fetchProducts();
36    }, []);
37
38    // Function to filter products based on the category
39    const handleFilterChange = (category: string) => {
40      setFilter(category);
41
42      if (category === "All") {
43        setProducts(allProducts);
44      } else {
45        const filtered = allProducts.filter(
46          (product: any) => product.category === category
47        );
48        setProducts(filtered);
49      }
50    };
51
52    return (
53      <div>
54        <div className="w-full bg-[#f4f4f4] mt-8 p-4">
55          <div className="flex flex-wrap justify-around items-center gap-4 lg:gap-8">
56            { /* Filters Section */ }
57            <div className="flex items-center space-x-4 lg:space-x-5">
58              <div className="flex items-center space-x-3">
59                <div filter className="w-5 h-5 md:w-[25px] md:h-[25px] cursor-pointer" />
60                <DropdownMenu>
61                  <DropdownMenuTrigger>Filters</DropdownMenuTrigger>
62                  <DropdownMenuContent>
63                    <DropdownMenuLabel>Search by filters</DropdownMenuLabel>
64                    <DropdownMenuSeparator />
65                    <DropdownMenuItem onClick={() => handleFilterChange("All")}>
66                      All
67                    </DropdownMenuItem>
68                    <DropdownMenuItem onClick={() => handleFilterChange("Bed")}>
69                      Bed
70                    </DropdownMenuItem>
71                    <DropdownMenuItem onClick={() => handleFilterChange("Chair")}>
72                      Chairs
73                    </DropdownMenuItem>
74                    <DropdownMenuItem onClick={() => handleFilterChange("Sofa")}>
75                      Sofas
76                    </DropdownMenuItem>
77                    <DropdownMenuItem onClick={() => handleFilterChange("Table")}>
78                      Tables
79                    </DropdownMenuItem>
80                  </DropdownMenuContent>
81                </DropdownMenu>
82              </div>
83            </div>
84
85            { /* Divider */ }
86            <div className="hidden lg:block border border-[#9f9f9f] h-[40px]"></div>
87
88            { /* Showing Results */ }
89            <p className="text-sm md:text-base font-[400] text-black">
90              Showing {products.length} results for {filter}
91            </p>
92          </div>
93        </div>
94
95        { /* Products Section */ }
96        <div className="bg-gray-50 py-8 mt-8">
97          <div className="max-w-7xl mx-auto px-4">
98            <div className="grid grid-cols-1 sm:grid-cols-2 lg:grid-cols-3 gap-6">
99              {products.map((product: any) => (
100                <div
101                  key={product.id}
102                  className="bg-white shadow-lg rounded-lg overflow-hidden transition duration-500 transform hover:scale-105"
103                >
104                  <Image
105                    src={product.imageUrl}
106                    alt={product.name}
107                    width={400}
108                    height={400}
109                    className="w-100 h-64 object-cover"
110                  />
111                  <div className="p-4">
112                    <h3 className="text-gray-600 mt-2">{product.name}</h3>
113                    <p className="text-gray-600 mt-2">
114                      {product.price}
115                    </p>
116                    <p className="text-gray-600 mt-2">
117                      Discount: {product.discountPercentage}%
118                    </p>
119                    <Link href={`/${product.slug}`}>
120                      <button className="bg-slate-300 px-4 py-2 rounded-md mt-5">
121                        View More
122                      </button>
123                    </Link>
124                  </div>
125                </div>
126              ))}
127            </div>
128          </div>
129        </div>
130      </div>
131    );
132  };
133
134  export default FilterSection;
135
```

♣ Documentation:

- **Steps Taken to Build and Integrate Components:**

Set up Sanity CMS as the backend for product data.

Fetches product data dynamically using API calls in Next.js.

Created reusable components for ProductCard, ProductList, and Filter.

Integrated category filters for seamless user interactions.

- **Challenges Faced and Solutions Implemented:**

Challenge: Debugging the filter functionality to dynamically update the product list.

- **Solution:** Refactored the filter logic to handle category-based updates efficiently.

Challenge: Managing Add to cart functionality.

- **Solution:** Used Redux Toolkit for managing add to cart functionality.

Challenge: Ensuring responsive design across various devices.

- **Solution:** Leveraged Tailwind CSS for consistent and adaptive layouts.

- **Best Practices Followed:**

Kept components reusable and modular for easier maintenance.

Followed Next.js conventions for dynamic routing and API integration.

Ensured responsive design and accessibility for all components.