

DAY 4 – DYNAMIC FRONTEND

COMPONENTS COMFORTY

MARKETPLACE.

Index.

1. Objective
2. Components Implemented
 - Product Listing Component
 - Product Detail Component
 - Category Component
 - Search Bar
 - Cart Component
 - Pagination Component
 - Related Products Component
 - Header aur Footer Components
 - Notifications Component
 - Wishlist Component
3. Steps Taken for Implementation
4. Challenges aur Solutions
5. Functional Deliverables
6. Code Deliverables
7. Documentation
8. Submission
9. Conclusion
10. Index.1

11. Implementation Steps:	4
12. Challenges aur Solutions.....	7
13. 1..... Dynamic Routing:	7
14. 2..... Category Filtering:	8
15. 3..... State Management:	8
16. Functional Deliverables:.....	8
17. Documentation aur Submission:.....	9
18.	

Day 4 mein focus tha dynamic frontend components ky design aur development par, jo marketplace data ko display karega Sanity CMS ya APIs se fetch karke. Jiss mein hamara Goal tha modular, reusable, aur responsive components banane ka, aur scalable web application development ke liye real-world practices apply karna ka.

1. Product Listing Component

- Products ko dynamic grid layout mein render kiya.
- Fields show kiye jaise Product Name, Price, Image, aur Stock Status.

2. Product Detail Component:

- Individual product pages ke liye dynamic routes banaye.
- Fields include kiye jaise Product Description, Price, aur Available Sizes.

3. Category Component:

- Categories ko database se dynamically fetch karke display kiya.

- Category-based filtering enable kiya.

4. Search Bar:

- Products ko name ya tags ke basis par filter karne ki functionality implement ki.

5. Cart Component:

- Items, quantities, aur total prices ko state management se track kiya

6. Pagination Component:

- Product lists ko manageable pages mein divide karne ke liye pagination add ki.

7. Related Products Component:

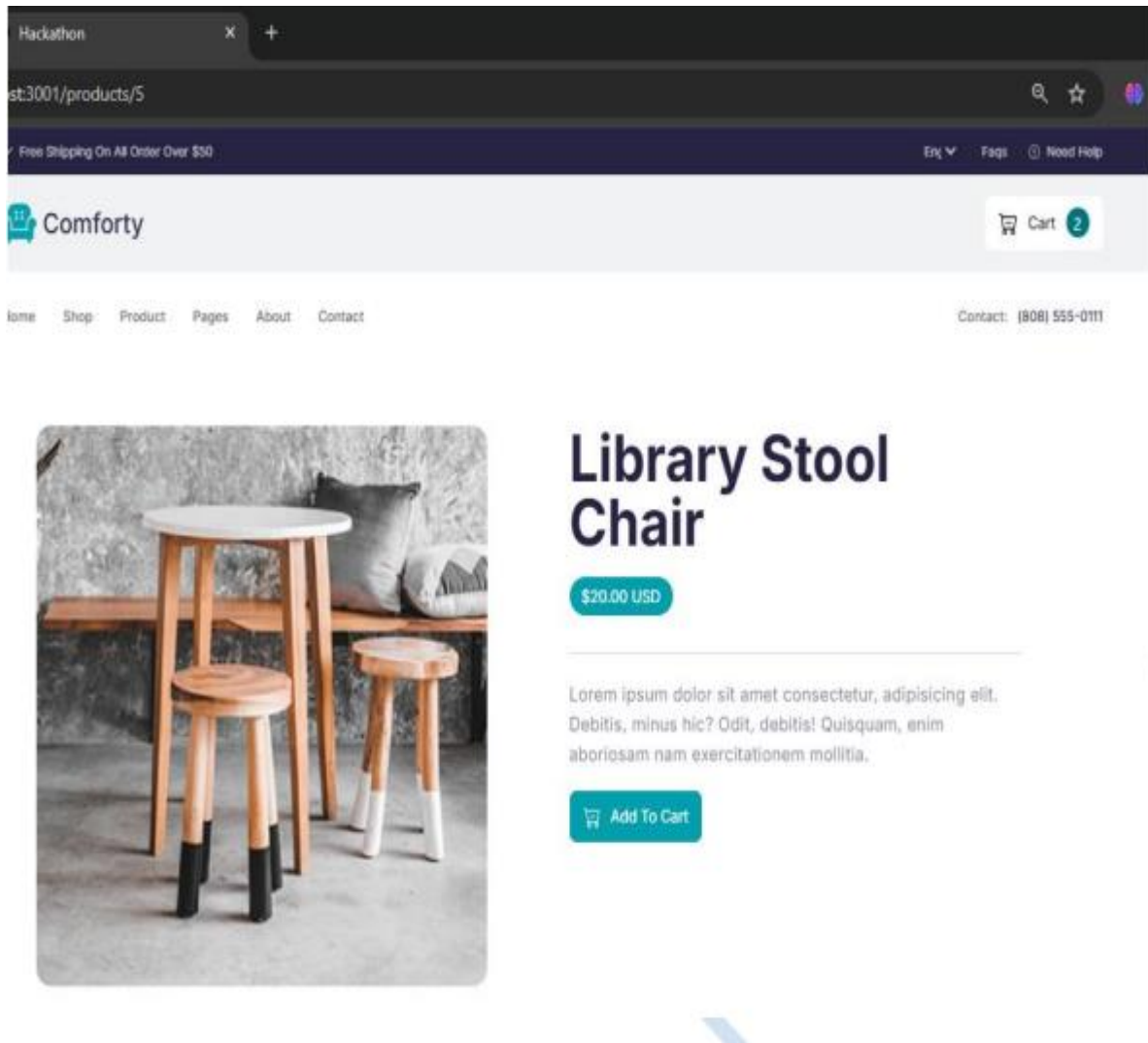
- Tags ya categories ke basis par Product Detail page par related products suggest kiye.

8. Header aur Footer Components:

- Consistent navigation elements banaye with links to key pages.

9. Notifications Component:

- Actions jaise cart mein items add karne ke liye toast notifications add ki.



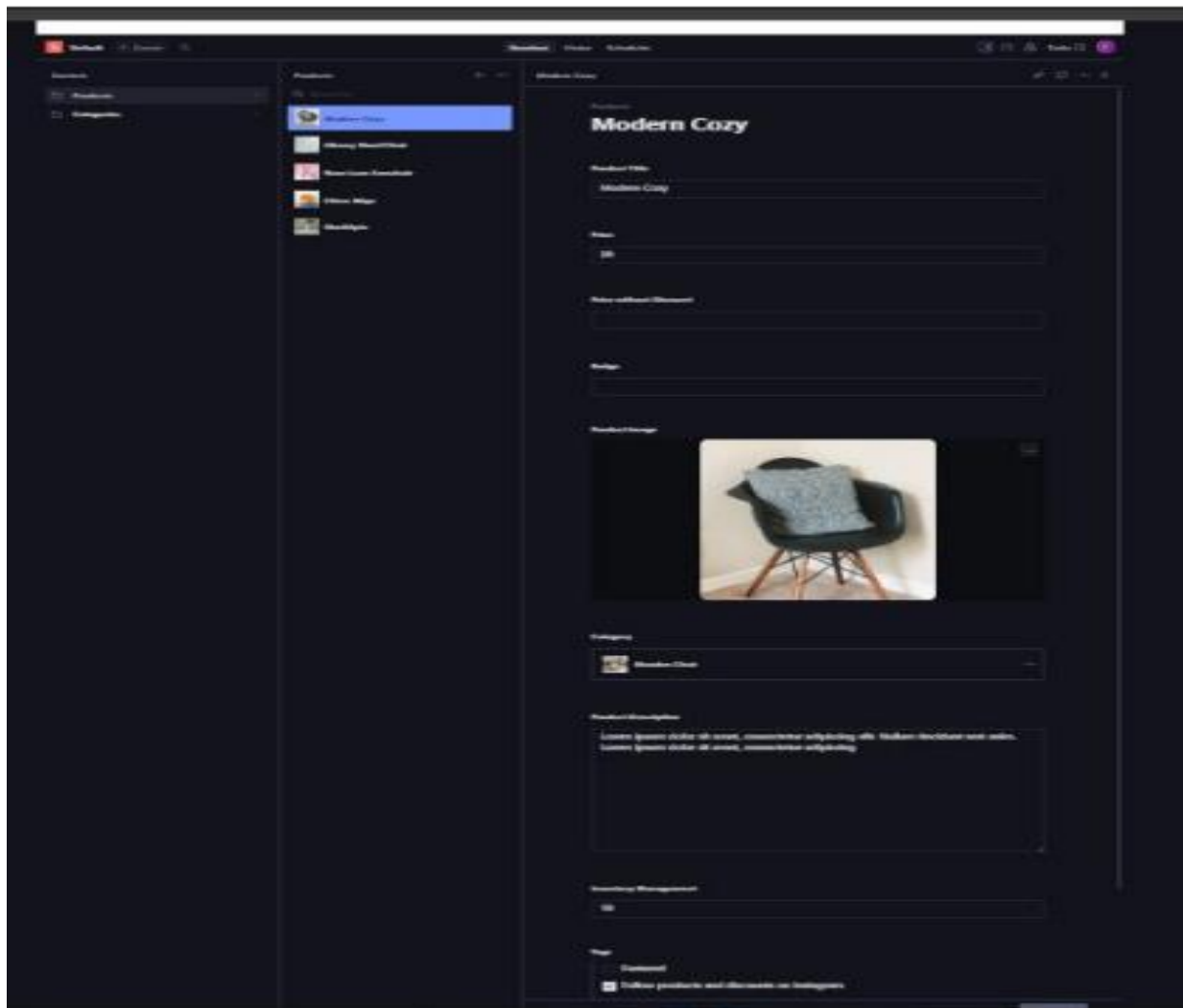
10. Wishlist Component:

- Users ko products save karne ki permission di using local storage for persistence.

Implementation Steps:

1. Setup:

- Ensure kiya ki Next.js project Sanity CMS se connected ho.
- API endpoints ko data retrieval ke liye test kiya.



2. Component Design:

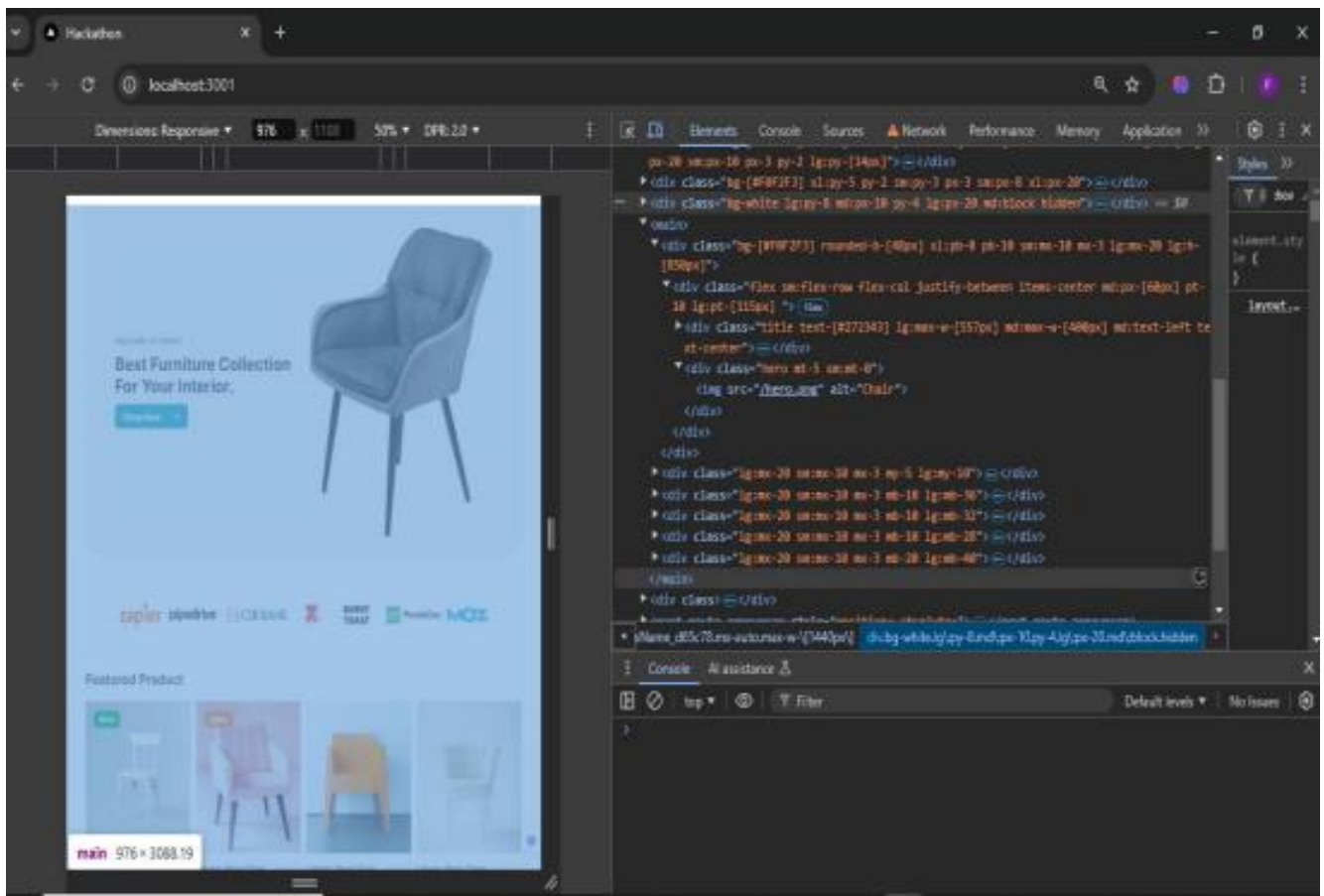
- Modular design principles ko use karke reusable components banaye jaise ProductCard aur CategoryFilter.
- Props use karke dynamic data ko components mein pass kiya.

3. State Management:

- Cart aur wishlist data ke liye global state ko manage karne ke liye useContext ka use kiya.

4. Styling:

- Responsive aur modern design ke liye Tailwind CSS use kiya.
- Mobile-first design strategies implement ki.



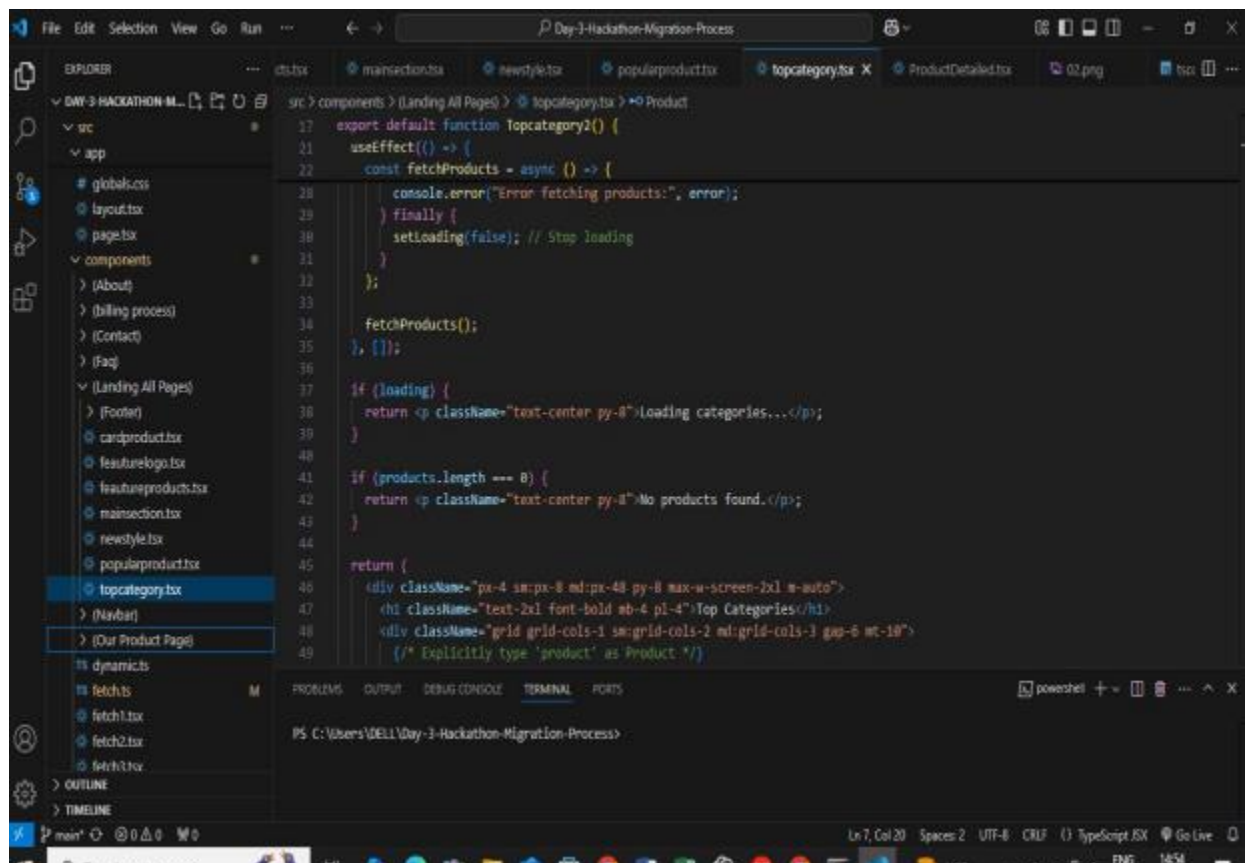
5. Performance Optimization

- Lazy loading aur pagination ka use kiya large datasets ke liye.

Challenges aur Solutions.

1. Dynamic Routing:

- a. Challenge: Accurate data loading ensure karna for individual product pages.
- b. Solution: Next.js dynamic routing aur fallback options use kiya for improved UX.



```
File Edit Selection View Go Run ... P Day-3-Hackathon-Migration-Process
EXPLORER
src > components > (Landing All Pages) > topcategory.tsx > Product
src
  app
    globals.css
    layout.tsx
    page.tsx
  components
    (About)
    (Billing process)
    (Contact)
    (Faq)
    (Landing All Pages)
    (Footer)
    cardproduct.tsx
    featurelogo.tsx
    featureproducts.tsx
    mainsection.tsx
    newstyle.tsx
    popularproduct.tsx
    topcategory.tsx
    (Navbar)
    (Our Product Page)
  ts dynamics
    fetch.ts
    fetch1.tsx
    fetch2.tsx
    fetch3.tsx
  OUTLINE
  TIMELINE
main'

src > components > (Landing All Pages) > topcategory.tsx > Product
17 export default function Topcategory2() {
21   useEffect(() => {
22     const fetchProducts = async () => {
28       console.error("Error fetching products:", error);
29     } finally {
30       setLoading(false); // Stop loading
31     }
32   };
33
34   fetchProducts();
35 }, []);
36
37 if (loading) {
38   return <p className="text-center py-8">Loading categories...</p>;
39 }
40
41 if (products.length === 0) {
42   return <p className="text-center py-8">No products found.</p>;
43 }
44
45 return (
46   <div className="px-4 sm:px-8 md:px-48 py-8 max-w-screen-2xl m-auto">
47     <h1 className="text-2xl font-bold mb-4 pl-4">Top Categories</h1>
48     <div className="grid grid-cols-1 sm:grid-cols-2 md:grid-cols-3 gap-6 mt-10">
49       {/* Explicitly type 'product' as Product */}
50     </div>
51   </div>
52 );
53 }
```

2. Category Filtering:

- a. Challenge: Selected categories ko displayed data ke sath sync karna.
- b. Solution: Query parameters use kiye for dynamic product filtering.

3. State Management:

- a. Challenge: Cart state ko pages ke across consistent rakhna.
- b. Solution: React context use karke globally state share ki.

Your Cart

Your cart is empty.

Summary	
Sub Total	\$0.00
Estimate Delivery & Handling	Free
Total	\$0.00
Proceed to Checkout	

Functional Deliverables:

- 1. Ek fully functional product listing page jo dynamic data display kare.
- 2. Dynamic routing use karke individual product detail pages.
- 3. Products ke liye search functionality.
- 4. Large product datasets ke liye pagination.
- 5. Responsive design for all components.

Key Components:

Made by LR

- ProductCard
- ProductList
- CategoryFilter
- SearchBar

Scripts:

- Dynamic routing logic for product details.
- Filtering aur search functionalities.

Documentation aur Submission:

1. Steps Taken: Components banane aur integrate karne ka process detail mein likha.
2. Challenges Faced: Issues aur unke respective solutions outline kiye.
3. Best Practices: Modularity, reusability, aur responsive design principles ko highlight kiya.

Day 4 ek opportunity tha real-world frontend challenges par kaam karne ka, jo dynamic data handling aur scalable component design ke importance ko emphasize karta hai. Responsive aur modular components implement karke ek solid foundation banaya gaya jo professional aur user-friendly marketplace development ke liye zaruri hai. State management, filtering, aur dynamic routing ki integration ne application ke interactivity aur usability ko enhance kiya. Ye components aur practices future client projects aur professional development ke liye critical step ke taur par serve karenge.

Best Of Luck!

Made by LR