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 and Footer Components**
- **Notifications Component**
- **Wishlist Component**

3. **Steps Taken for Implementation**
4. **Challenges and Solutions**
5. **Functional Deliverables**
6. **Code Deliverables**
7. **Documentation**
8. **Submission**
9. **Conclusion**

Objective

On Day 4, the focus was on designing and developing dynamic frontend components for displaying marketplace data fetched from Sanity CMS or APIs. The goal was to build modular, reusable, and responsive components while applying real-world practices for scalable web application development.

Components Implemented

1. Product Listing Component

- Rendered products dynamically in a grid layout.
- Displayed fields like Product Name, Price, Image, and Stock Status.

2. Product Detail Component

- Created dynamic routes to render individual product pages.
- Included fields such as Product Description, Price, and Available Sizes.

3. Category Component

- Displayed categories dynamically fetched from the database.
- Enabled category-based filtering.

4. Search Bar

- Implemented functionality to filter products by name or tags.

5. Cart Component

- Tracked items, quantities, and total prices using state management.

6. Pagination Component

- Added pagination to divide product lists into manageable pages.

7. Related Products Component

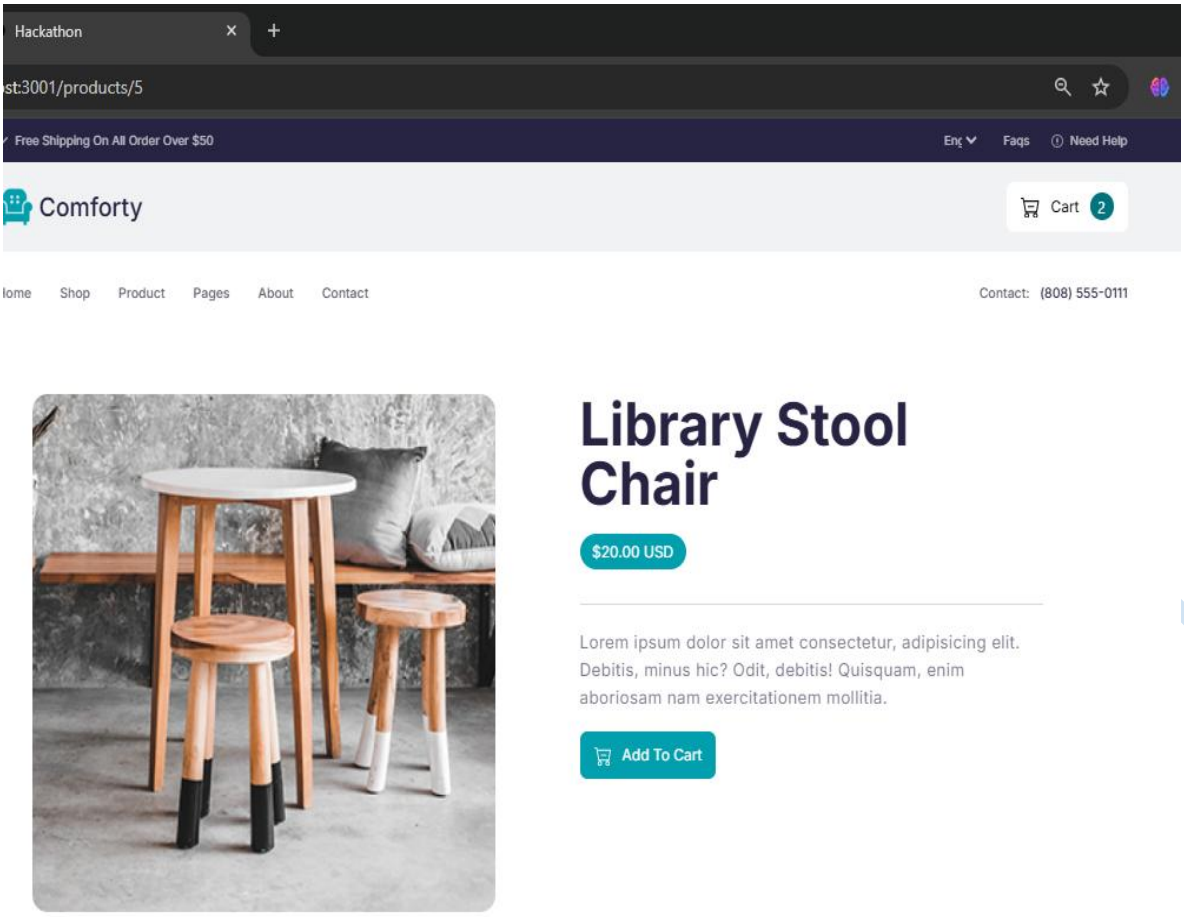
- Suggested products on the Product Detail page based on tags or categories.

8. Header and Footer Components

- Built consistent navigation elements with links to key pages.

9. Notifications Component

- Added toast notifications for actions like adding items to the cart.



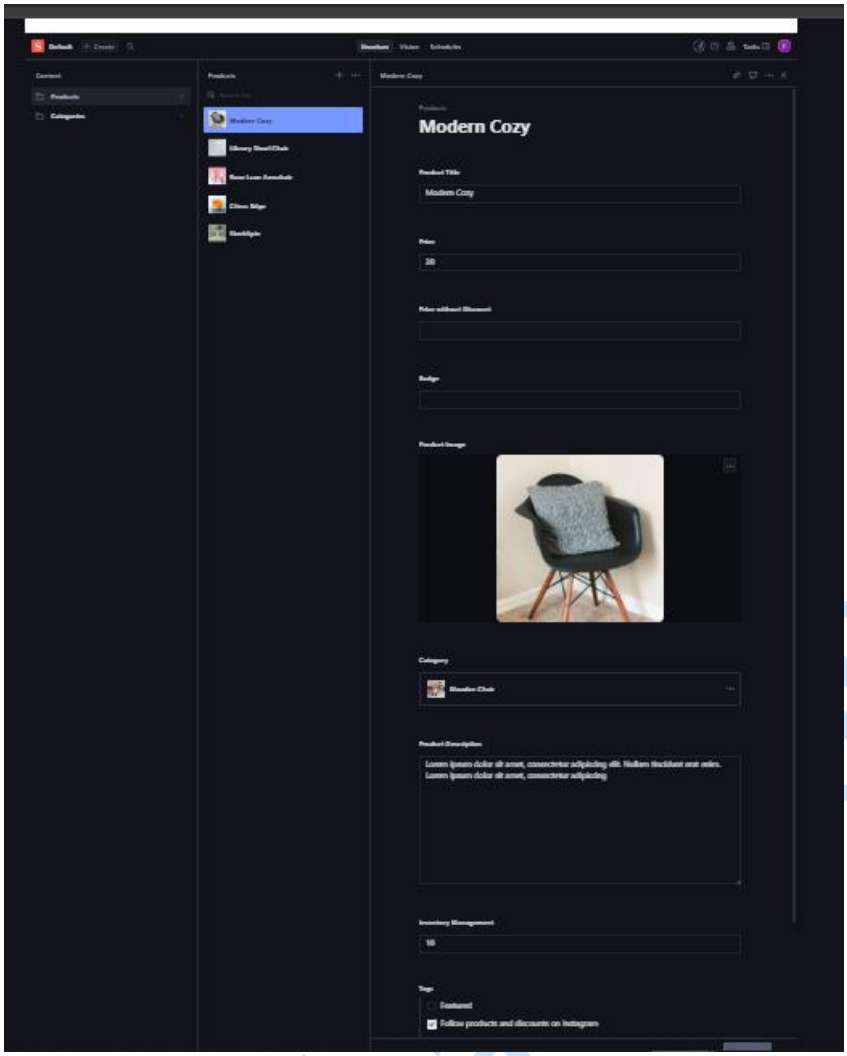
10. Wishlist Component

- Allowed users to save products using local storage for persistence.

Steps Taken for Implementation

1.Setup

- Ensured the Next.js project was connected to Sanity CMS.
- Tested API endpoints for data retrieval.



2.Component Design

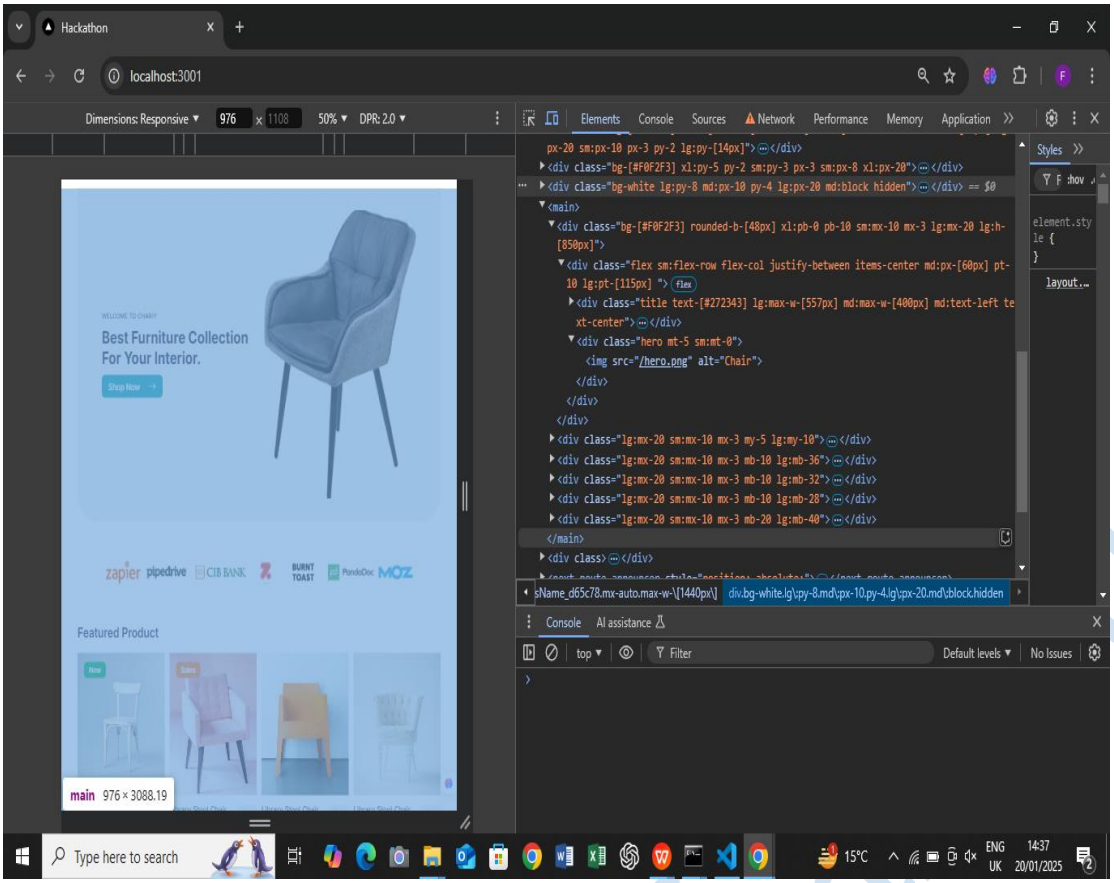
- Used modular design principles to create reusable components like ProductCard and CategoryFilter.
- Leveraged props to pass dynamic data into components

3.State Management

- Managed global state using useContext for cart and wishlist data.

4.Styling

- Used Tailwind CSS for responsive and modern design.
- Implemented mobile-first design strategies.



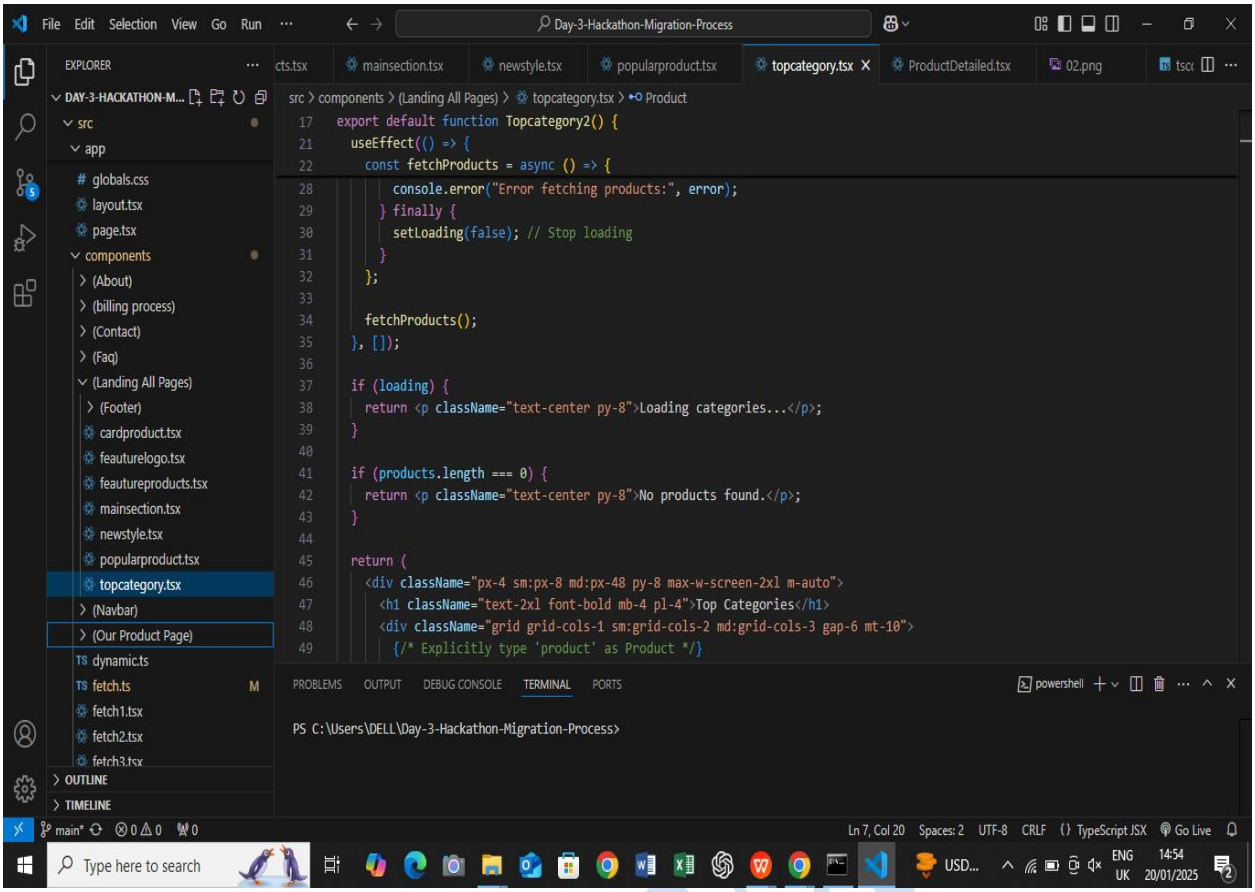
5.Performance Optimization

- Applied lazy loading for images and pagination for large datasets.

Challenges and Solutions

1. Dynamic Routing

Challenge: Ensuring accurate data loading for individual product pages.
Solution: Used Next.js dynamic routing and fallback options for improved UX.

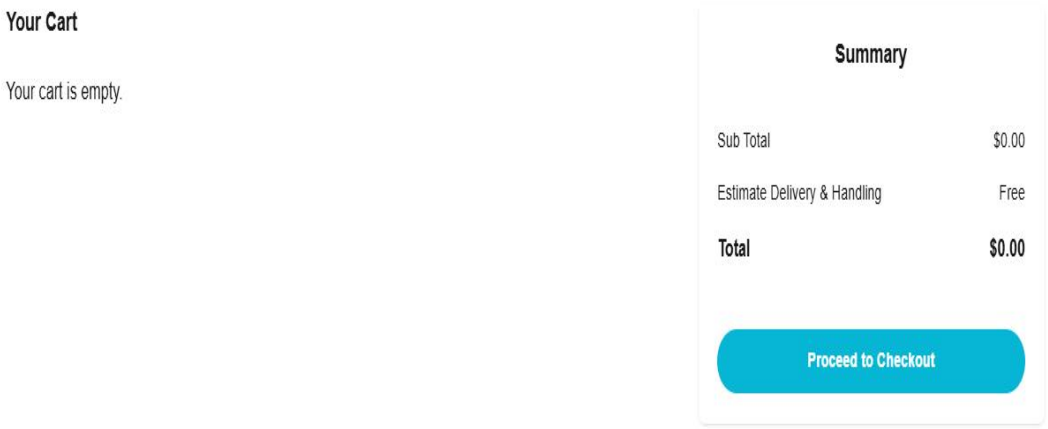


2. Category Filtering

Challenge: Syncing selected categories with displayed data.
Solution: Used query parameters to dynamically filter products.

3. State Management

Challenge: Maintaining a consistent cart state across pages.
Solution: Utilized React context to share state globally.



Functional Deliverables

1. A fully functional product listing page displaying dynamic data.
2. Individual product detail pages using dynamic routing.
3. Search functionality for products.
4. Pagination for large product datasets.
5. Responsive design for all components.

Code Deliverable

Key Components:

- ProductCard
- ProductList
- CategoryFilter
- SearchBar

Scripts:

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

Documentation

1. **Steps Taken**
    - Detailed the process of building and integrating components.
2. **Challenges Faced**
    - Outlined issues and their respective solutions.
3. **Best Practices**
    - Highlighted modularity, reusability, and responsive design principles.

Conclusion

Day 4 provided an opportunity to work on real-world frontend challenges, emphasizing the importance of dynamic data handling and scalable component design. By implementing responsive and modular components, a solid foundation was established for building a professional and user-friendly marketplace. The integration of state management, filtering, and dynamic routing enhanced the application's interactivity and usability. These components and practices will serve as a critical step in preparing for future client projects and professional development.