

Report: Dynamic Frontend Components for Marketplace

Prepared By: LAIBA NAZ

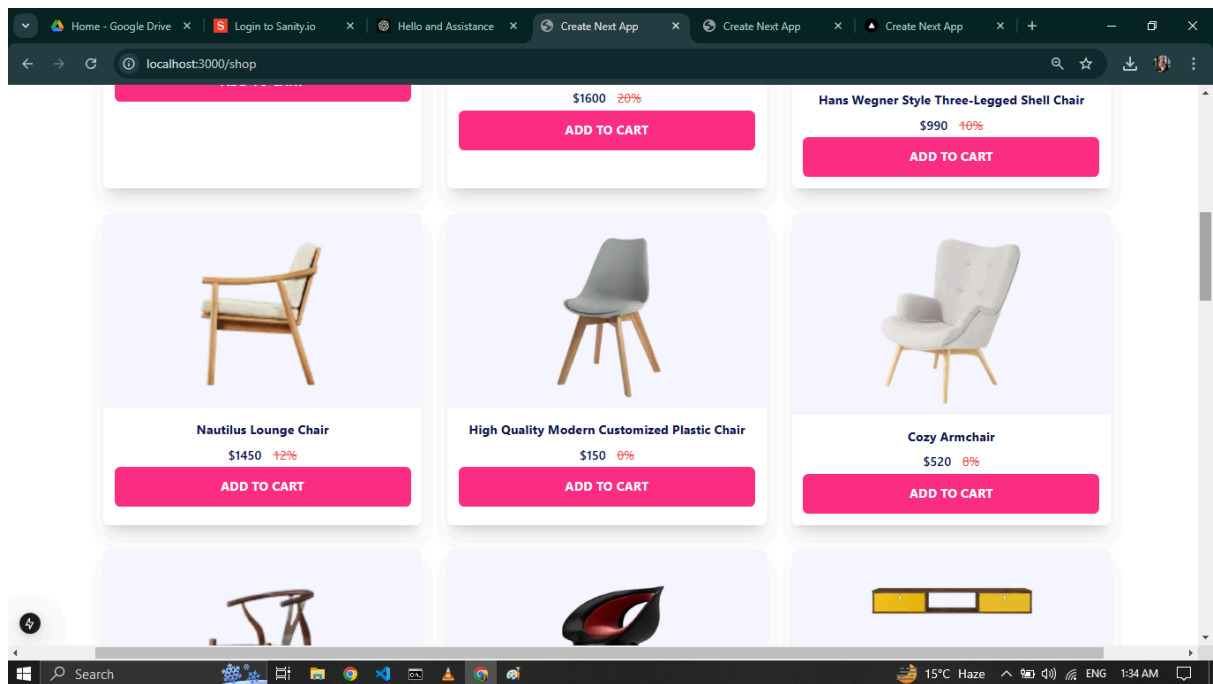
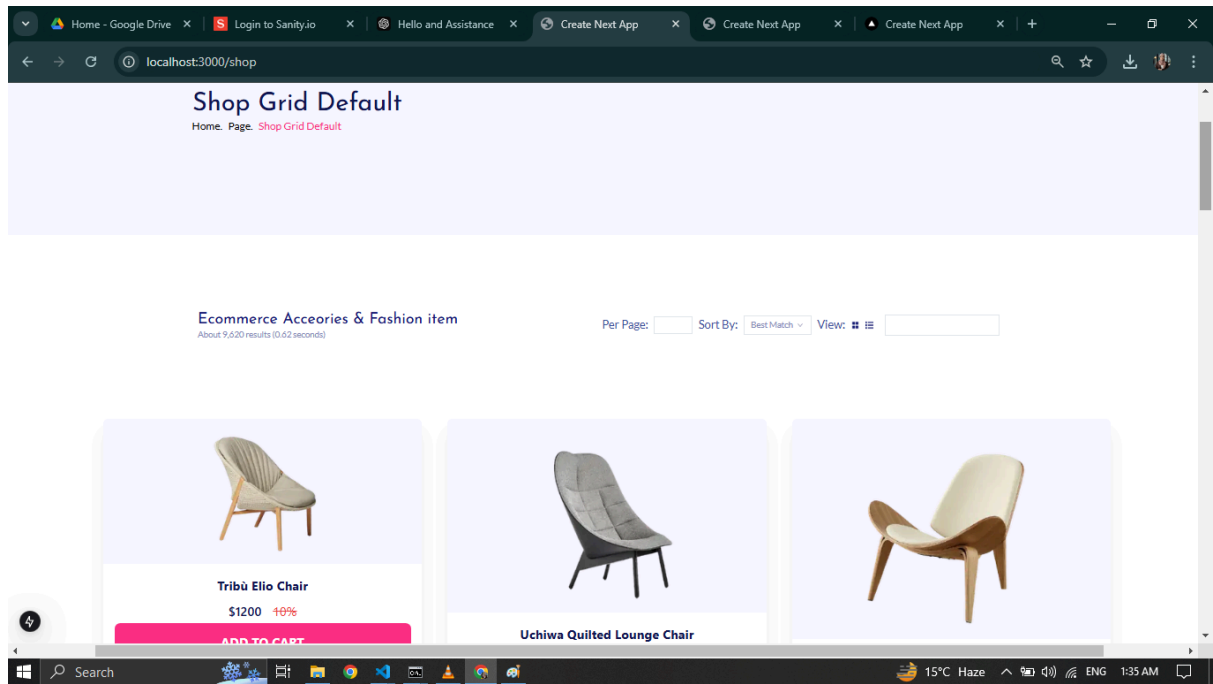
Objective:

The aim of this project was to build dynamic frontend components for a marketplace application. The focus was limited to three components: Product Listing, Add to Cart, and Dynamic Routing. These components ensure a seamless user experience while interacting with the marketplace platform.

Components Implemented

1. Product Listing Component:

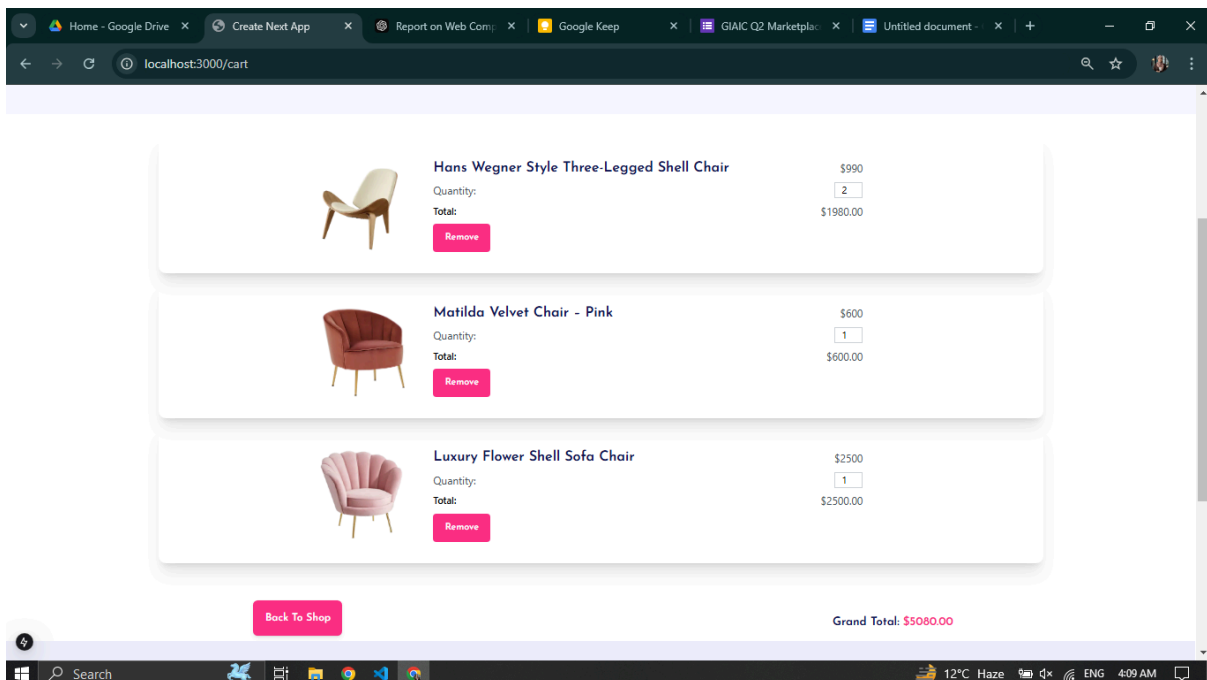
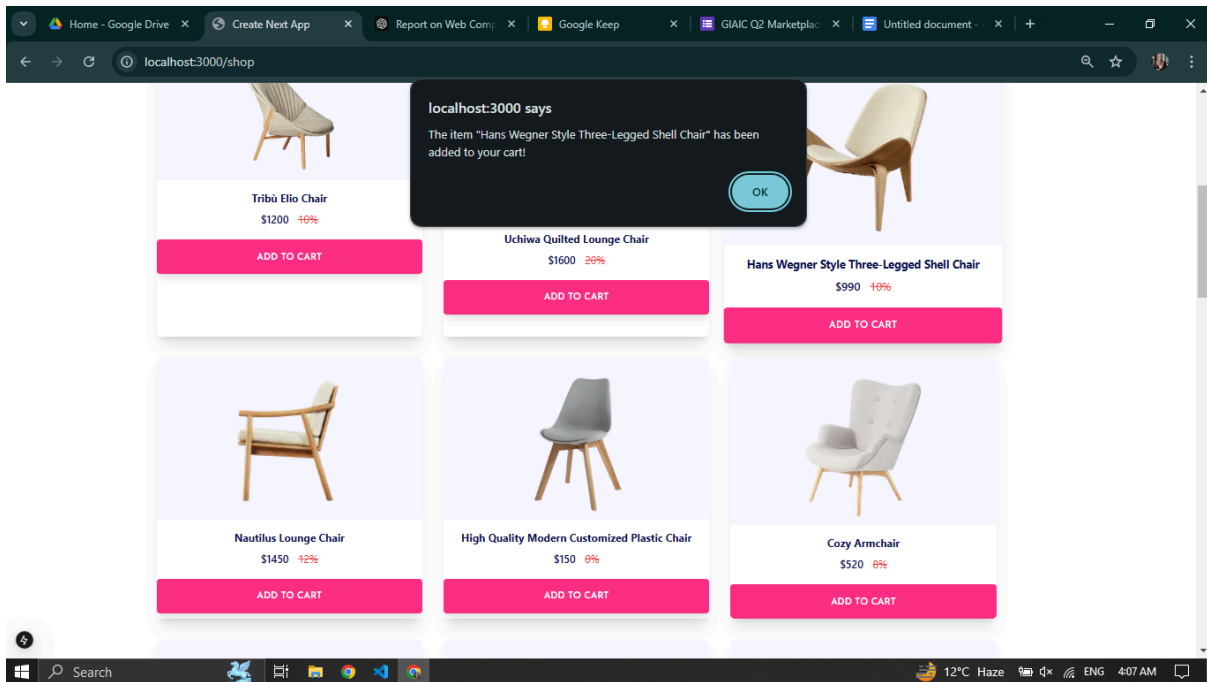
- **Objective:** Display products dynamically in a grid layout fetched from the data source.
- **Features Implemented:**
 - Rendered product details including:
 - Product Name
 - Price
 - Image
 - Stock Status
 - Grid layout with cards for each product.
 - Data fetched dynamically from Sanity CMS or API endpoints.
- **Challenges Faced:**
 - Managing data loading states.
 - Handling errors during API calls.
- **Solutions:**
 - Implemented loading indicators and error handling logic.



2. Add to Cart Component:

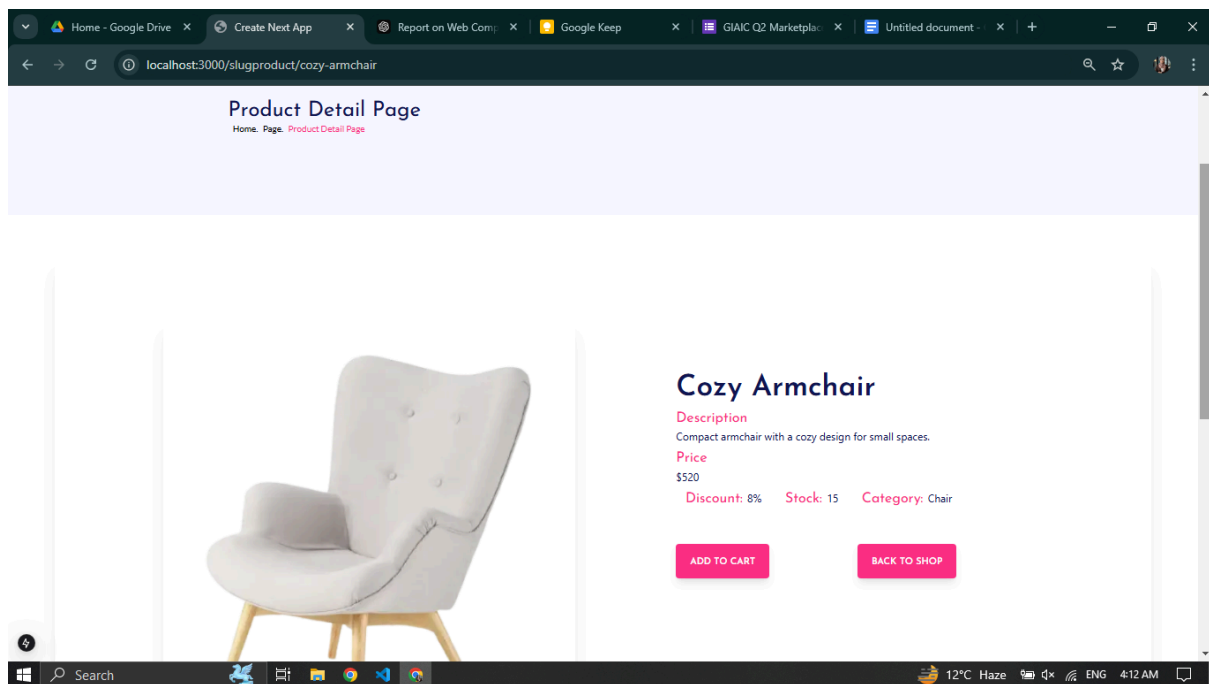
- **Objective:** Allow users to add products to a cart and manage cart items dynamically.
- **Features Implemented:**
 - Added items to the cart with quantity management.

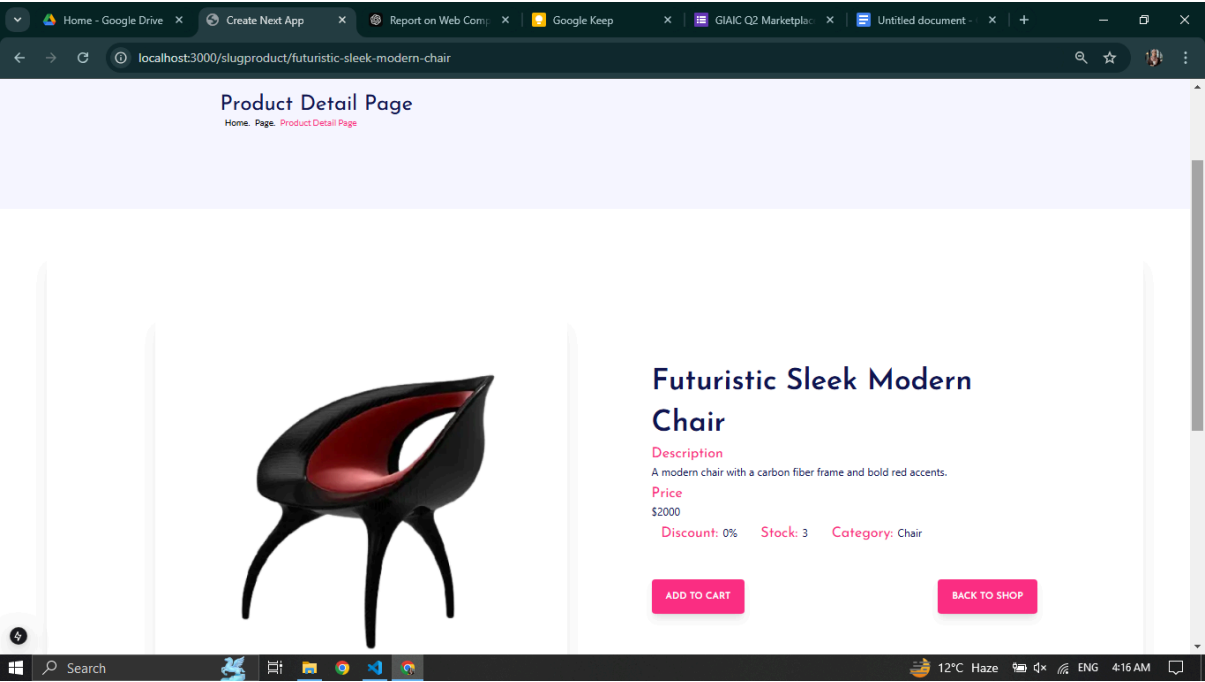
- Remove items from cart.
- Displayed total price calculation in real-time.
- Used state management (React Context or useState) to track cart items.
- **Challenges Faced:**
 - Managing cart updates across components.
- **Solutions:**
 - Used a global state management solution (React Context) to ensure
 - consistent cart updates.



3. Dynamic Routing:

- **Objective:** Enable dynamic product detail pages for individual products.
- **Features Implemented:**
 - Utilized Next.js dynamic routing for individual product pages.
 - Rendered product-specific details such as:
 - Product Description
 - Available Sizes or Colors
 - Pricing Details
 - Integrated proper fallback handling for non-existent product IDs.
- **Challenges Faced:**
 - Managing non-existent routes.
- **Solutions:**
 - Implemented error handling for invalid product IDs using Next.js 404 pages.





Key Learning Outcomes:

1. **Dynamic Data Handling:** Successfully fetched and displayed data from external APIs or CMS.
2. **State Management:** Applied effective state management techniques to handle cart functionality.
3. **Responsive Design:** Ensured that components are mobile-friendly and adhere to UI/UX best practices.
4. **Dynamic Routing:** Learned how to utilize Next.js routing effectively for a seamless user experience.

Challenges and Resolutions:

Challenge	Resolution
API errors during product fetching	Added error boundary and retry mechanisms
Cart updates consistency	Used React Context for global state
Dynamic route handling	Implemented fallback logic and 404 pages

Conclusion:

The project focused on three critical components of the marketplace application: Product Listing, Add to Cart, and Dynamic Routing. These components were implemented successfully, adhering to best practices for scalability and user experience. Future enhancements could include integrating additional features such as filtering, search, and pagination.

Submission Details:

- **Screenshots:** Provided as evidence of functional components.
 - **Code Snippets:** Key implementations for the three components.
 - **GitHub Repository:** [Insert Repository Link]
-

Prepared By: Laiba Naz