

Day 4 - Dynamic Frontend Components Furnishful

1. Introduction

This document summarizes the development process, challenges faced, and best practices followed while building dynamic frontend components for our marketplace. The aim was to create a user-friendly, efficient, and scalable frontend for the marketplace.

2. Functionalities Implemented

2.1 Product Listing Component

- Developed to render product data dynamically in a grid layout. □ Key fields included: ○ Product Name ○ Price ○ Image ○ Stock Status
- Example layout: cards displaying product details, making it visually appealing and easy to navigate.

2.2 Product Detail Component

- Created individual product detail pages using dynamic routing in Next.js.
- Included detailed fields such as:
 - Product Description
 - Price
 - Available Sizes or Colors
- Ensured accurate routing and data rendering to provide a seamless user experience.

2.3 Category Component

- Displayed categories dynamically fetched from the data source.
- Enabled filtering of products by selected categories, enhancing user experience by allowing easy navigation.

2.4 Search Bar

- Implemented search functionality to filter products by name or tags.
- Improved user experience by providing quick access to desired products.

2.5 Cart Component □ Displayed added items, quantity,

and total price.

- Used state management to track cart items effectively, ensuring smooth functionality.

2.6 Wishlist Component

- Allowed users to save products for future reference.
- Used local storage or a global state management tool to persist data, ensuring user preferences were maintained.

2.7 Checkout Flow Component

- Created a multi-step form for checkout, including fields for:
 - Billing and shipping address
 - Payment details (mock implementation)
- Ensured a smooth and user-friendly checkout process.

2.8 Footer and Header Components

- Built consistent navigation and branding elements.
- Included links to key pages (e.g., Home, About, Contact).
- Ensured responsiveness and accessibility to provide a cohesive user experience.

3. API Integration and Dynamic Routing

- Developed scripts and logic to integrate APIs for fetching product data.
- Handled dynamic routing to individual product detail pages, ensuring efficient data flow and accurate page rendering.

4. Challenges Faced

4.1 Max Event Listeners Error

- **Problem:** Encountered issues with the maximum number of event listeners.
 - **Solution:** Properly managed and removed event listeners to prevent memory leaks.

4.2 Expected Token Error

- **Problem:** Faced syntax errors related to unexpected tokens.
 - **Solution:** Debugged and reviewed code to ensure proper syntax and structure.

5. Best Practices Followed

- **Code Copies:** Maintained copies of the code to prevent data loss during development.
- **Component-Based Architecture:** Adopted a component-based architecture to keep the code neat and clean. This approach made the development process

6. Conclusion

This report outlines the steps taken and the strategies employed to overcome challenges during the development of dynamic frontend components for our marketplace. The adoption of best practices ensured that the codebase remained clean and manageable.