

# Document Title: "Day 4 - Dynamic Frontend Components - [Avion]"

## Steps Taken to Build and Integrate Components:

1. **Fetch Products from Sanity:** Used `useEffect` and `fetch` (or `useState` with `fetch`) to get data dynamically on the client side.
2. **Created View Collection Page:** Displayed all fetched products in a grid layout using a `ProductList` component.
3. **Implemented Dynamic Routing:** Used `Next.js [slug].tsx` for individual product pages, retrieving data based on the clicked product.
4. **Styled Components:** Applied Tailwind CSS for responsiveness and a clean UI.

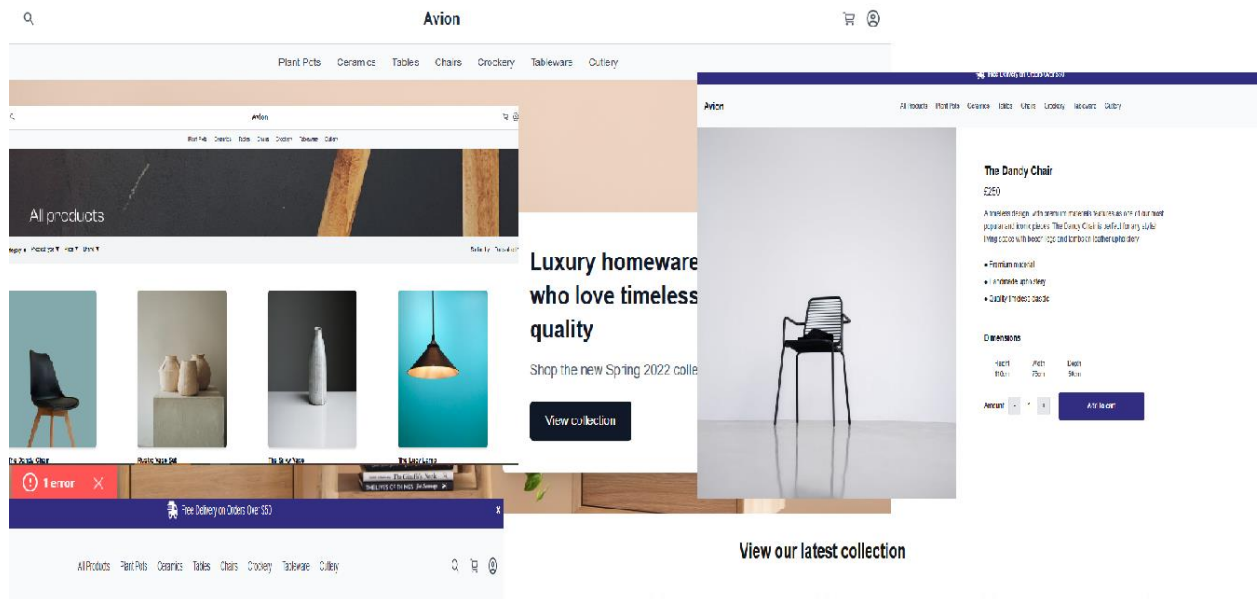
## Challenges Faced and Solutions Implemented:

- **Issue:** API fetching delays affected product rendering.
  - **Solution:** Used a loading state (`useState`) to improve UX while fetching data.
- **Issue:** Routing errors when clicking on products.
  - **Solution:** Ensured `Next.js` dynamic routes matched Sanity slugs correctly.
- **Issue:** UI responsiveness across different devices.
  - **Solution:** Used Tailwind CSS flex/grid for better layout handling.

## Best Practices Followed:

- **Client-Side Fetching:** Used `useEffect` for dynamic data retrieval from Sanity.
- **Modular Code:** Created reusable components (`Product`, `ProductListing`).
- **Optimized Performance:** Used `useState` to manage UI states efficiently.
- **Scalability:** Maintained a well-structured project folder hierarchy.

Screen shots of my website:



## Checkout

Summary

Item 1	\$250
Item 2	\$150
Item 3	\$400

Discount code

Apply

Shipping Information

Full Name

John Doe

Address

123 Main St

City

New York

The Poplar suede sofa

\$980

Tropical Vibe

\$550

Sleek Living

\$300

Serene Seat

\$350