**Day 4 Report: Building Dynamic Frontend Components for Your Marketplace (Nike app)**

**Task Overview:**

The objective of this task was to implement dynamic frontend components for a marketplace application. The goal was to integrate APIs with Sanity, fetch data using GROQ queries, and display the fetched data on the frontend. Additionally, dynamic routing, responsive design, and an interactive shopping cart were implemented.

**Implementation Details:**

**1. API Integration with Sanity:**

* Integrated the Sanity content platform into the application for managing and fetching product data.
* Utilized Sanity’s rich APIs and schemas to create a structured and scalable backend for the marketplace.

**2. GROQ Query to Fetch Data:**

* Constructed a GROQ query to fetch relevant product details, including:
  + Product Name
  + Price
  + Status
  + Category
  + Colors
  + Image URL
  + Slug (added specifically for dynamic routing)
* Example GROQ Query:
* const query = `\*[\_type == "product"] {
* productName,
* price,
* status,
* category,
* colors,
* "imageUrl": image.asset->url,
* Slug (added specifically for dynamic routing)
* Example GROQ Query:



* Implemented this query in the frontend using a custom fetch function.

**3. Schema Enhancement:**

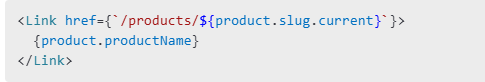
* Modified the Sanity schema to include a **Slug** field to enable unique, user-friendly URLs for each product.
* Ensured the slug is generated automatically based on the product name for consistency.

**4. Thunder Client Usage:**

* Used Thunder Client to test the API endpoints during the development process.
* Verified that all data is fetched correctly and API responses are as expected.

**5. Dynamic Routing:**

* Implemented dynamic routing for product pages using the slug field from Sanity.
* Each product dynamically generates a new page when clicked, displaying detailed information.
* Example of dynamic route:



**6. Add to Cart Button and Functionality:**

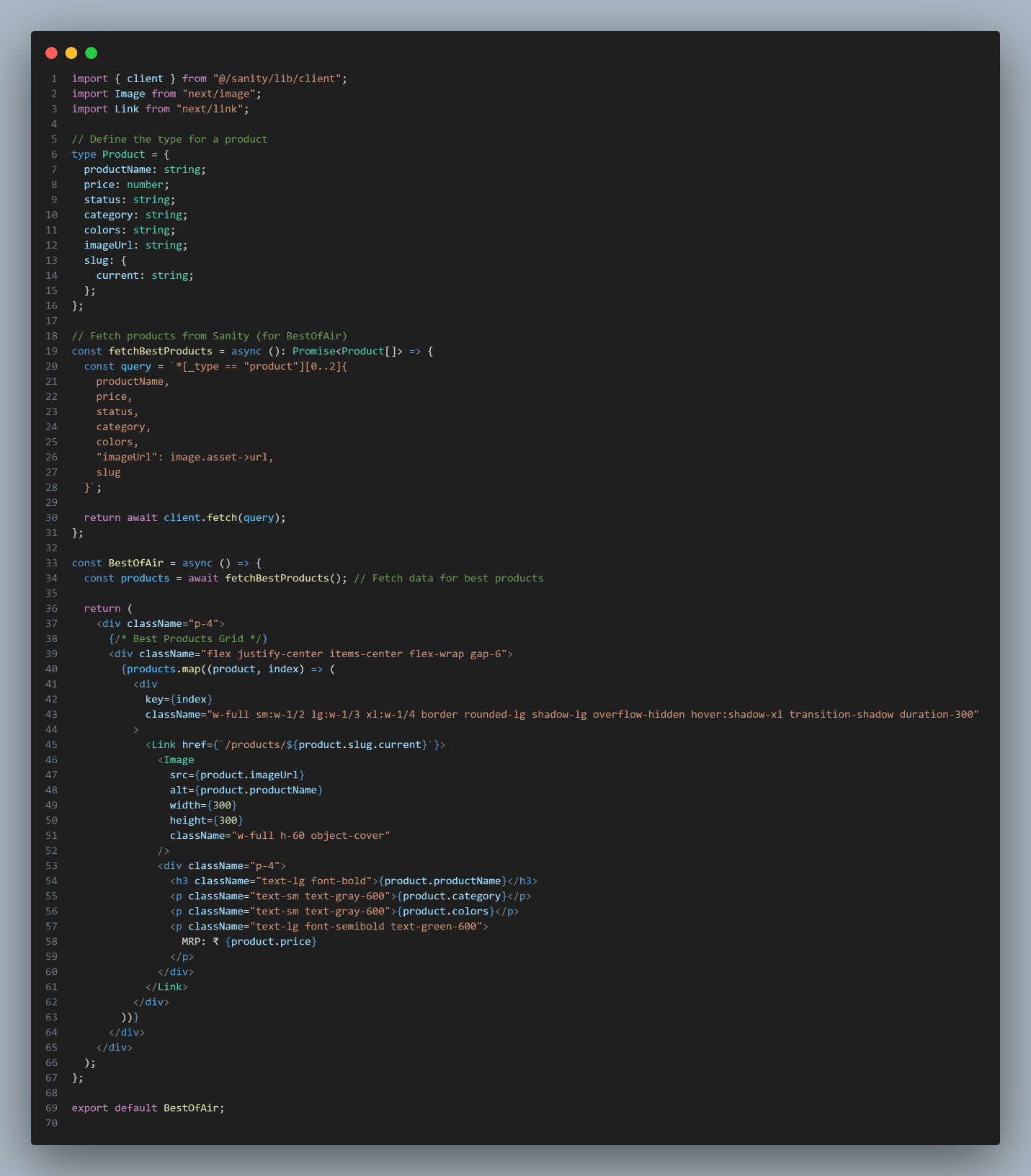
* Added an **Add to Cart** button for each product.
* Implemented a shopping cart that opens on button click, allowing users to view selected items.
* Cart updates dynamically as items are added or removed.

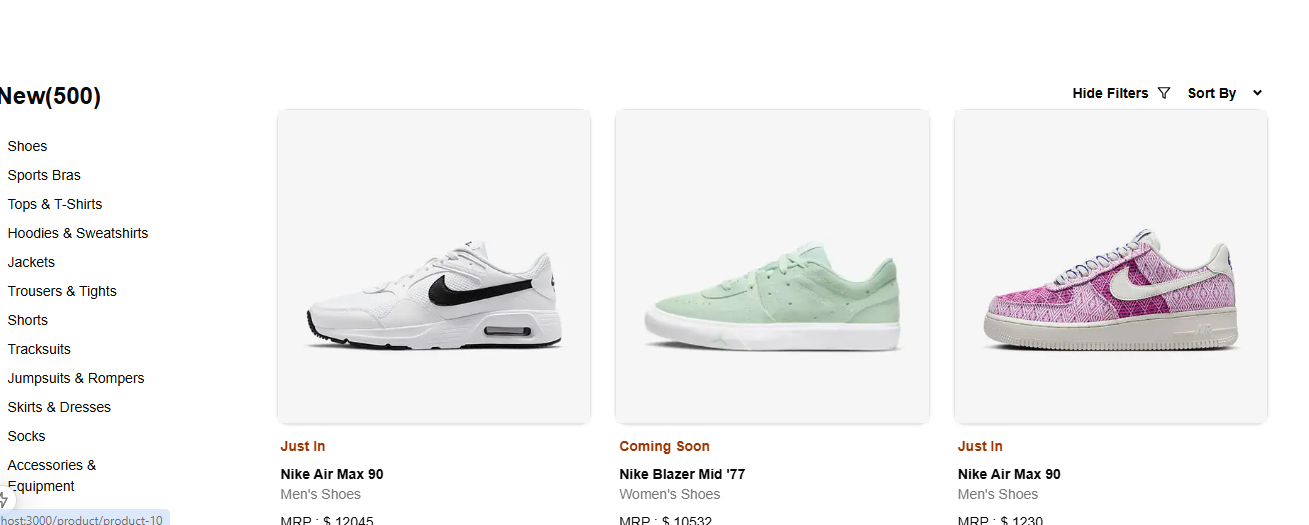
**7. Responsiveness:**

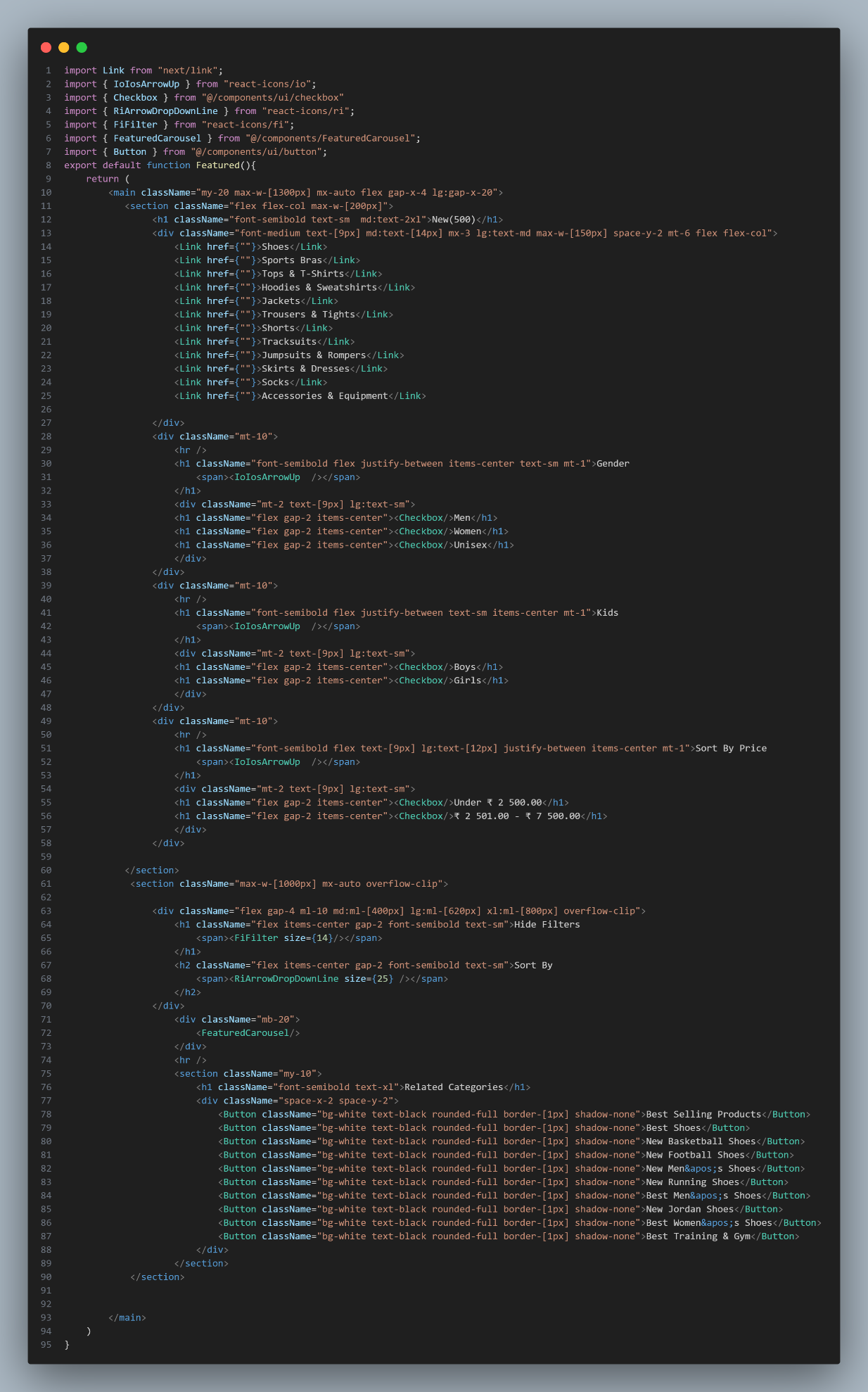
* Designed all components to be fully responsive using a combination of:
  + Tailwind CSS for quick prototyping.
  + Media queries for additional customization.
* Ensured the layout adapts seamlessly to mobile, tablet, and desktop screens.

**Progress Achieved:**

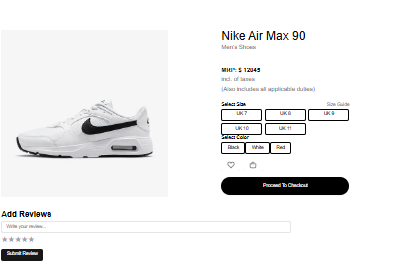
* Successfully integrated API with Sanity.
* Data fetched and displayed on the frontend using GROQ queries.
* Dynamic routing implemented for individual product pages.
* Added interactive cart functionality.
* Improved responsiveness for a seamless user experience.
* Further refine the shopping cart’s user interface.
* Implement user authentication to personalize the cart experience.
* Add filters and sorting options for products on the frontend.
* Optimize the application for performance improvements.

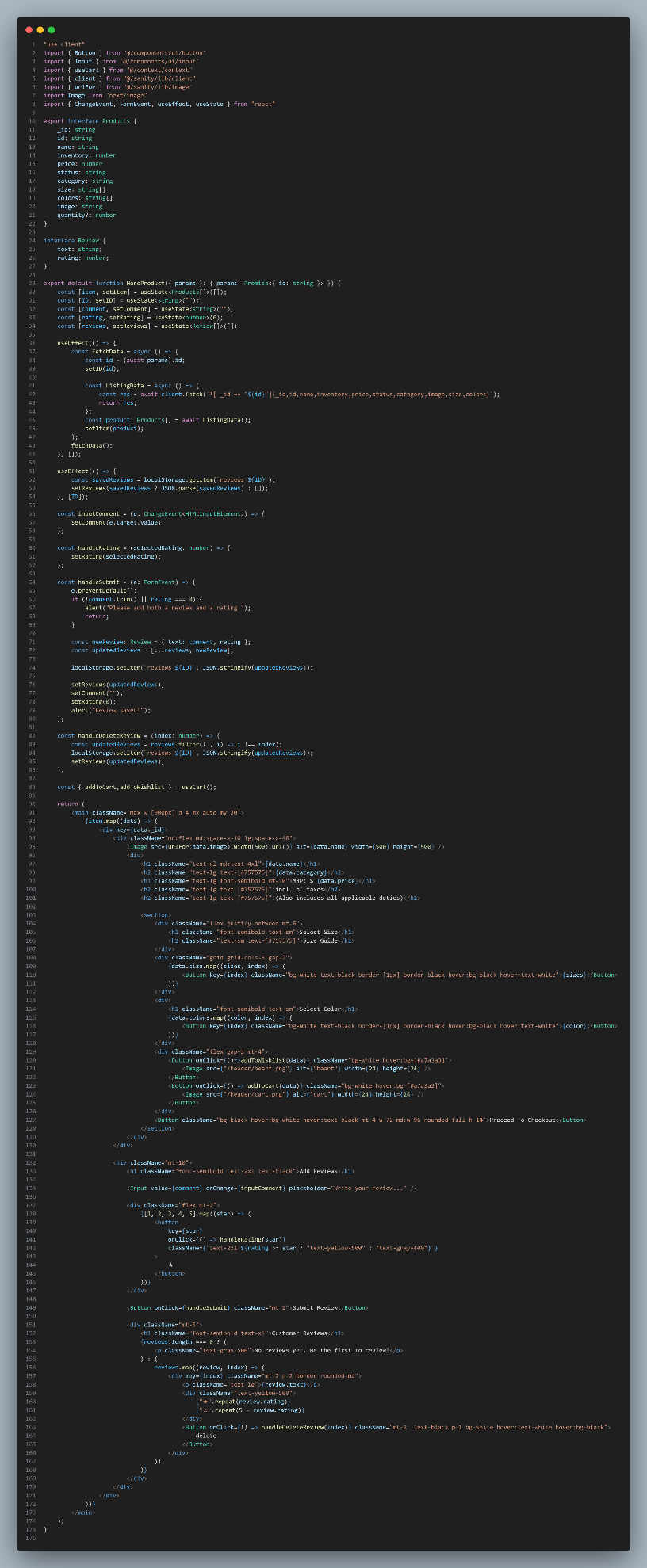


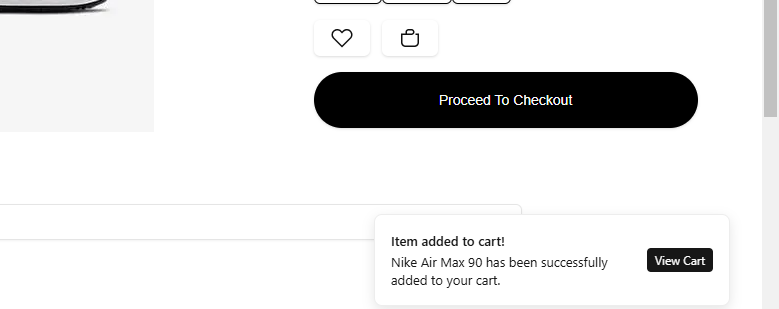


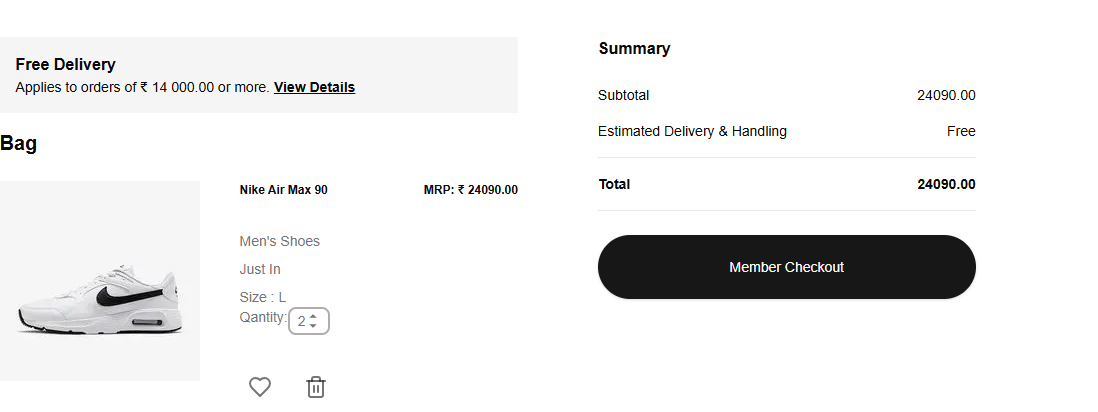


Dynamic page with add to cart and add reviews:

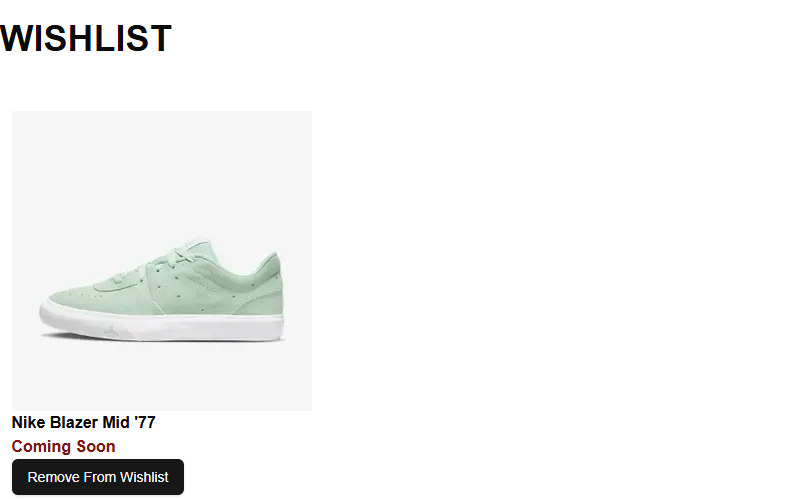




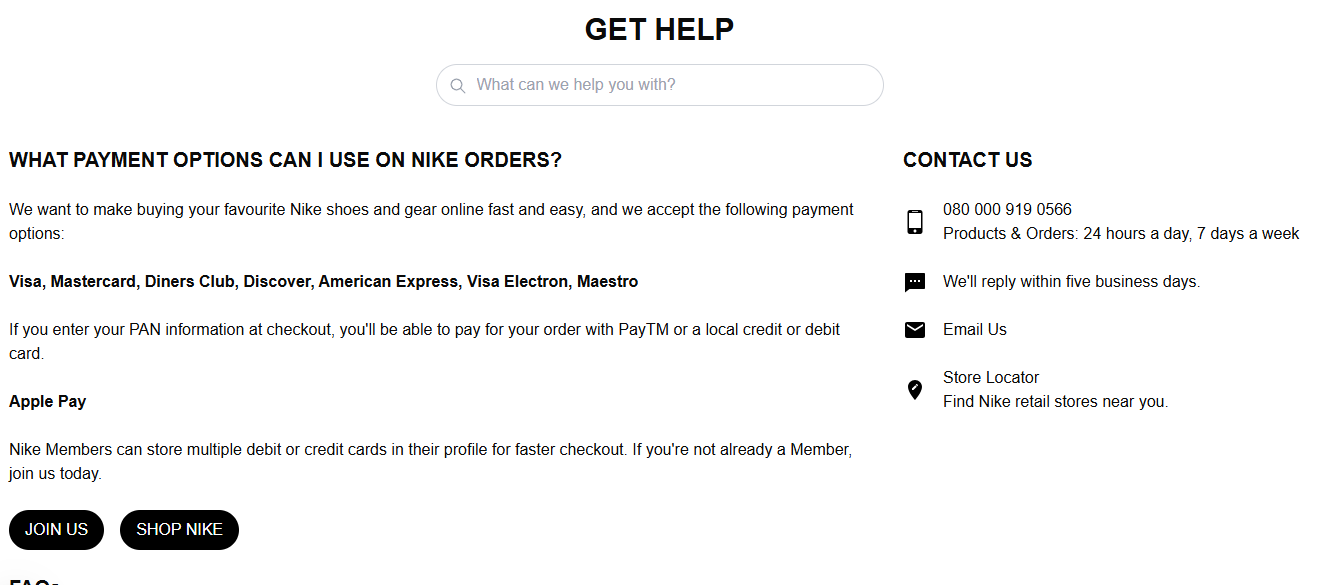




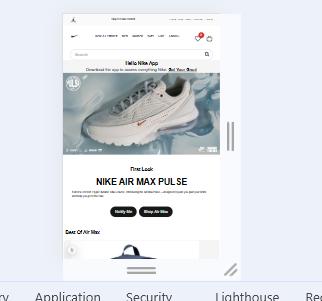




Help-us Page



**Responsiveness:**



**Conclusion:**

The Day 4 task was successfully completed with dynamic frontend components integrated into the marketplace application. The features implemented provide a strong foundation for further enhancements and deliver a smooth user experience. The use of Sanity and GROQ made the data management and fetching process efficient, while Thunder Client ensured accurate API testing and debugging.