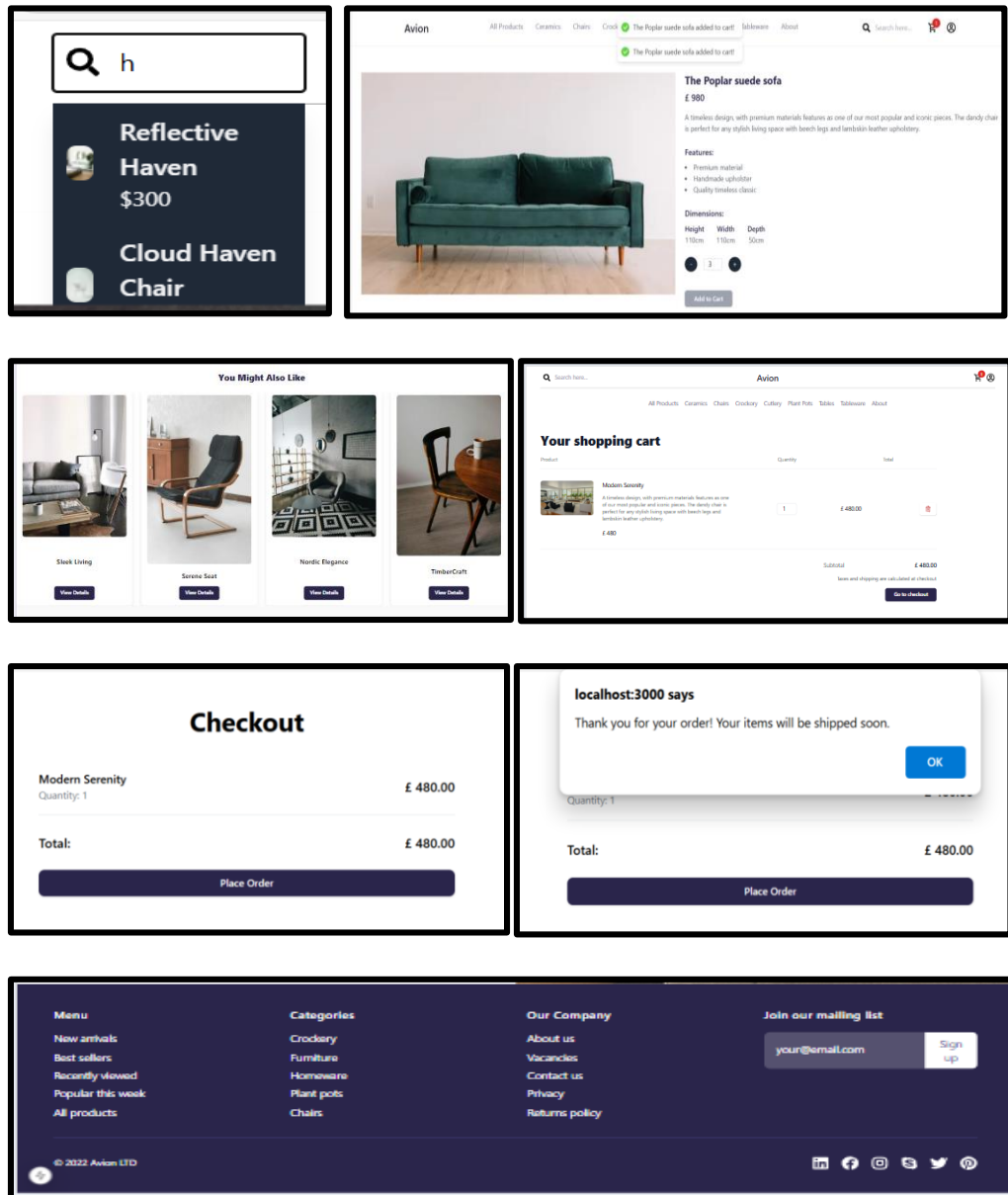# Day 4 - Dynamic Frontend Components

Built Components

1. **Product Listing Component**
   - Dynamically fetched and displayed product data from Sanity CMS or API.
   - Presented product names and prices in a clean, responsive grid layout for enhanced user experience.
2. **Product Detail Component**
   - Used dynamic routing to render individual product details, including name, price, description, and additional data.
   - Integrated an "Add to Cart" button with full functionality to update the cart.
3. **Category Component**
   - Fetched categories dynamically from the data source.
   - Allowed users to filter products based on selected categories, ensuring personalized browsing.
4. **Advanced Search Bar**
   - Implemented faceted search to combine multiple filters and provide precise results.
   - Limited suggestions to 4-5 results for a clean and focused experience.
5. **Cart Component**
   - Enabled users to view and manage their cart with options to update quantities or remove items.
6. **Checkout Process with Notification System**
   - Review items in the cart with total price calculation.
   - Confirm the order and display a success notification.
7. **Related Products Component**
   - Displayed similar products on product detail pages, dynamically recommended based on tags or categories.
8. **Footer and Header Components**
   - Ensured consistent navigation and branding across pages with links to Home, About, Contact, and other key sections.
   - Focused on responsiveness and accessibility for seamless interaction across devices.
9. **Notifications Component**
   - Implemented a system to notify users of actions like adding products to the cart, search errors, or successful order completion.

Screenshots:

Self-Validation Checklist for Day 4

Task	✓
Frontend Component Development	✓
Styling and Responsiveness	✓
Code Quality	✓
Documentation and Submission	✓
Final Review	✓

Challenges and Solutions

1. **Challenge:** Handling API delays while fetching data.
   Solution: Used and optimized performance with data caching tools.
2. **Challenge:** Managing complex state for cart, filters, and checkout.
   Solution: Reusable components effectively for seamless user interaction.
3. **Challenge:** Ensuring the checkout process is intuitive.
   Solution: Structured the process with clear steps, user validation, and error handling notifications.