

# Hackathon Day-4

## Day 4 – Dynamic Frontend Components Report for E-Commerce

### 1. Introduction

This report outlines the development of interactive and responsive frontend components for **E-Commerce**. The key focus areas include:

- **Interactive Product Display**
- **Product card component displayed on the homepage**
- **Creating a structured and organized product listing page**
- **Designing a detailed and interactive product detail page**
- **Developing a fully functional shopping cart for order management**
- **Processing order submission on the checkout page and storing data in Sanity**

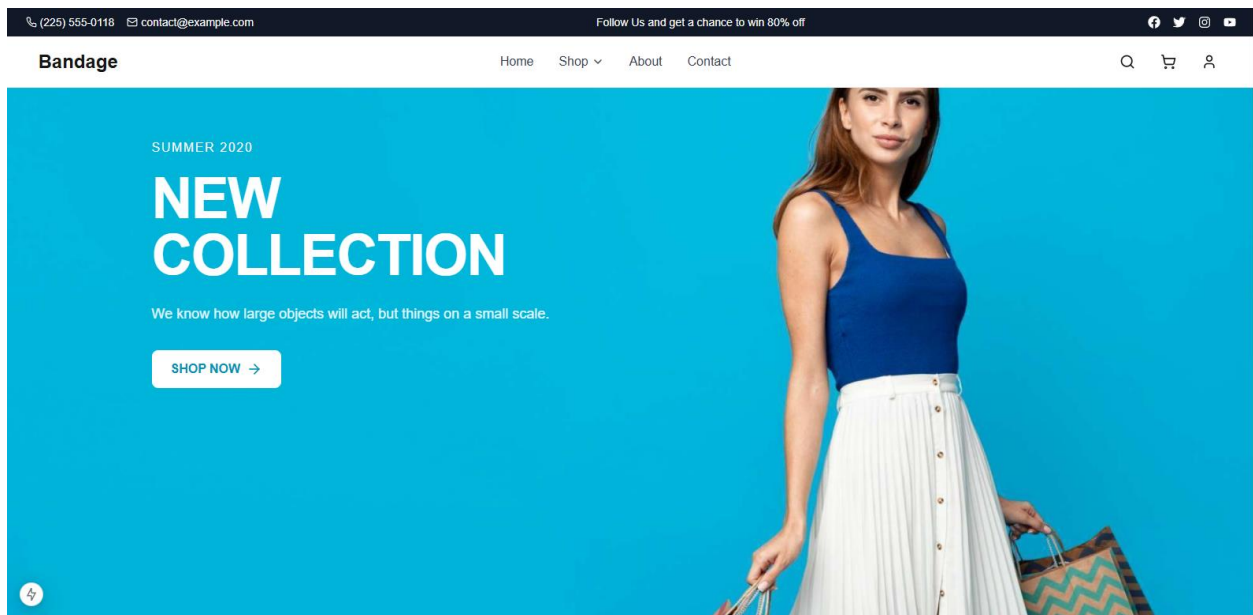
### 2. Summary Of Tasks

#### Step 1: Developing The Product Component

- Retrieved product data dynamically from the API.
- Fetched product data and presented it in a grid layout within a dedicated homepage component, highlighting names and prices.

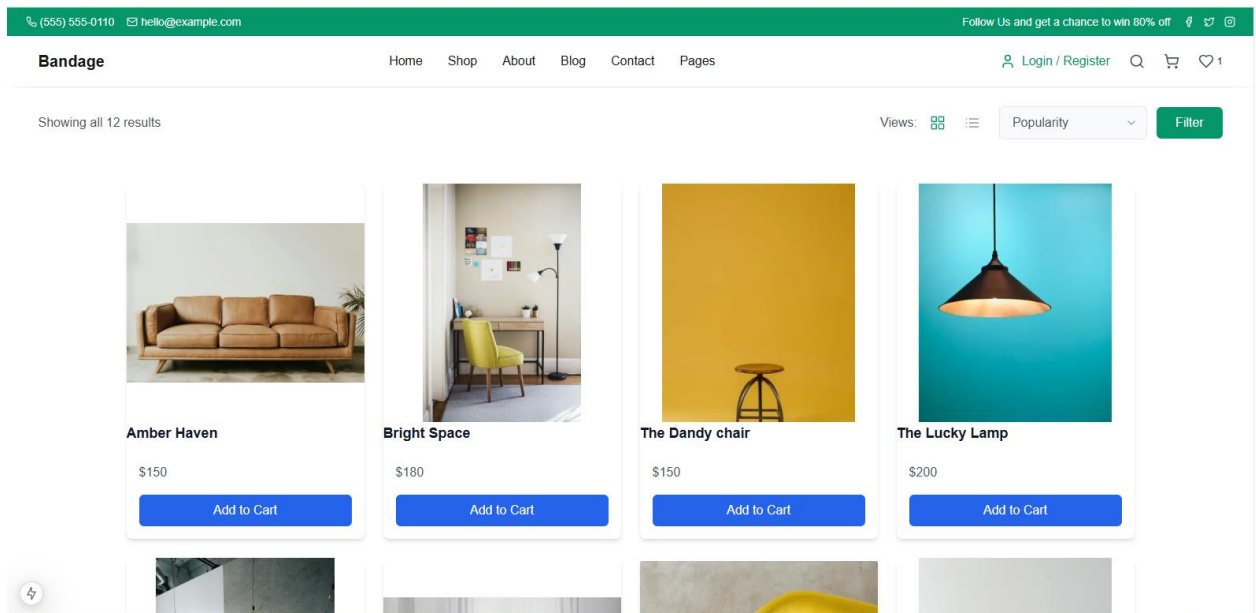


```
3 import { notFound } from 'next/navigation';
4 import ProductDetails from '@app/components/ProductDetail';
5
6 // Fetch product details using the _id value.
7 const productDetails = async (id: string) => {
8   const product = await client.fetch(
9     `*[ _type == "product" && _id == $id ][0] {
10       _id,
11       title,
12       productImage,
13       price,
14       description,
15       inventory,
16       quantity,
17       tags,
18       discountPercentage,
19       isNew
20     }`,
21     { id }
22   );
23
24   console.log("Fetched Product:", product); // Debugging line
25   return product || null;
26 }
27
```



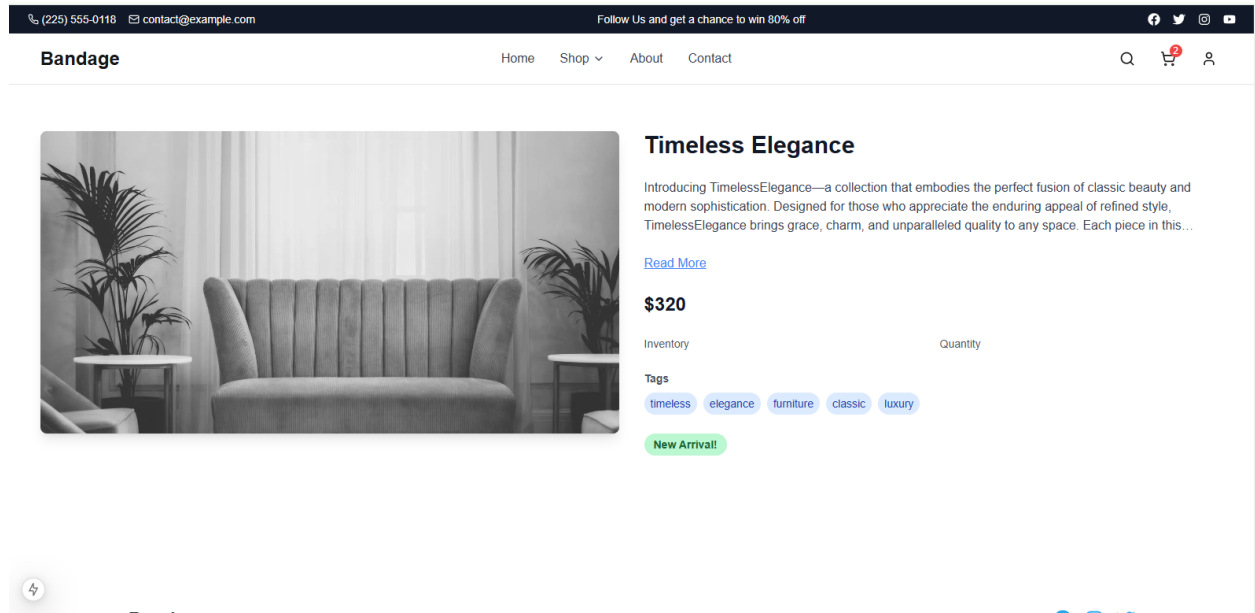
### Step 3: Creating the Product Listing Page

- Retrieved product data dynamically from the API.
- Showcased products in a flexible grid layout, highlighting names and prices.



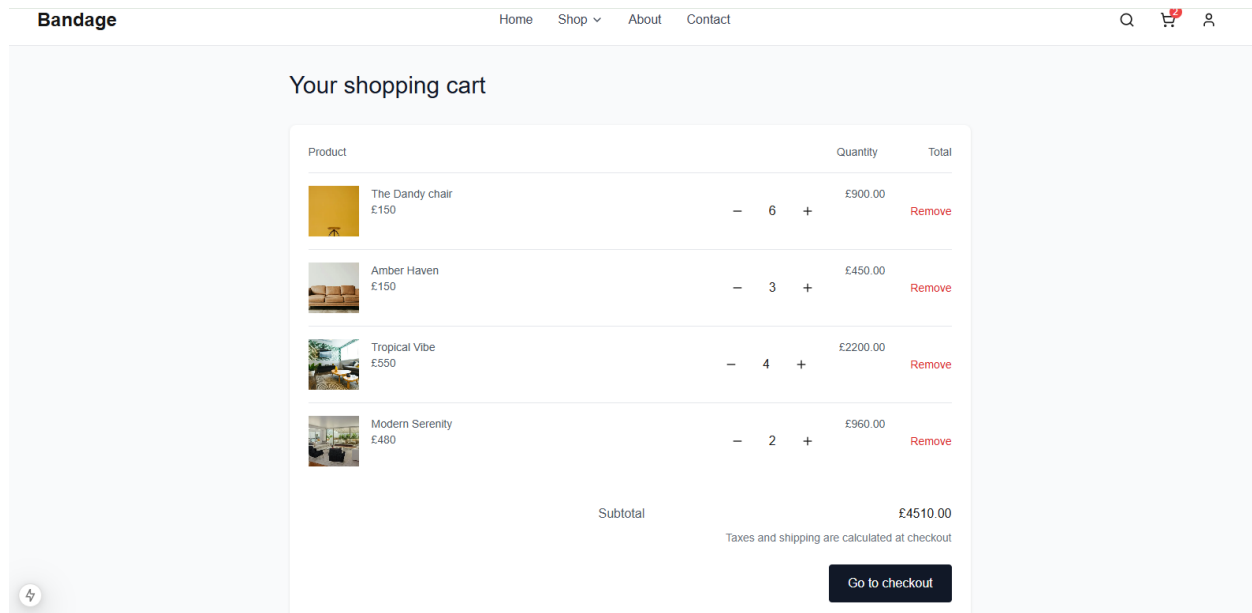
### Step 4: Product Detail Page

- Developed a dynamic product detail page with individual routing.
- Implemented an "Add to Cart" button for seamless shopping. Clicking it successfully adds items to the shopping cart.



## Step 5: Implementing Shopping Cart Functionality

- Allowed users to seamlessly add and remove products from the shopping cart.
- Implemented real-time cart updates and a seamless checkout process.







## Step 6: Successful Checkout

- Selecting the "Checkout" button navigates the user to the checkout page.
- Users can proceed by filling in their checkout details.

Cart > Check Out

### Order Summary

|   |   |
|---|---|
|  | The Dandy chair<br>Quantity: 6<br><b>\$900.00</b> |
|  | Amber Haven<br>Quantity: 3<br><b>\$450.00</b>     |
|  | Tropical Vibe<br>Quantity: 4<br><b>\$2200.00</b>  |
|  | Modern Serenity<br>Quantity: 2<br><b>\$960.00</b> |

SubTotal: \$4510.00

Discount: -\$0.00

**Total: \$4510.00**

### Billing Information

First Name

Last Name

Address

Email

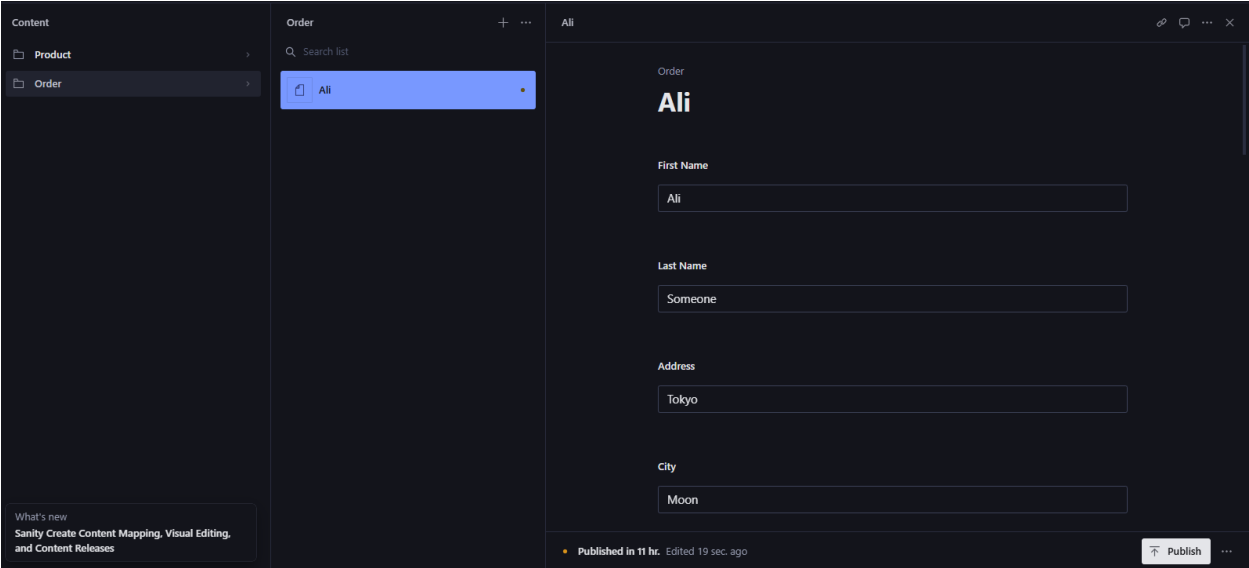
Phone

Zip Code

City

[Place Order](#)

Placing an order on the checkout page will transmit the data to **Sanity**.



Below is a structured **Self-Validation Checklist** in table format, derived from your report. It offers a clear framework for verification and ensures all tasks are effectively completed.

### Self-Validation Checklist for Frontend Development

|                                       |   |   |
|---------------------------------------|---|---|
| <b>Frontend Component Development</b> | ✓ | ✗ |
|---------------------------------------|---|---|

- ✓ Interactive product display
- ✗ Fully functional product detail page
- ✗ Category filtering method
- ✗ Search functionality
- ✓ Shopping cart functionality (add/remove items)
- ✓ Order checkout functionality
- ✗ Authentication page ( Login / Signup )

|                                  |   |   |
|----------------------------------|---|---|
| <b>Design and responsiveness</b> | ✓ | ✗ |
|----------------------------------|---|---|

- ✓ Responsive grid layout for product listings
- ✓ Consistent UI design across all pages
- ✓ Mobile responsiveness

## **Code Quality** ✓ X

- ✓ Reusable and modular components
- ✓ Well-structured and maintainable code
- ✓ Efficient state management and data binding
- ✓ Performance optimization and fast loading speed

## **Proper documentation and final submission** ✓ X

- ✓ Instructions for setup and application execution
- ✓ Well-documented code with comments and explanations
- × Timely submission before the deadline

## **Comprehensive final review** ✓ X

- ✓ Complete functionality testing (products, cart, checkout)
- ✓ UI/UX evaluated for a seamless user experience
- ✓ No bugs/issues identified in the development

## **Adherence to professional development practices** ✓ X

- ✓ Well-structured and reusable component architecture
- ✓ Efficient state management and real-time data binding
- ✓ Adaptive and intuitive UI/UX design
- ✓ Comprehensive documentation for code and workflows

## **Conclusion**

The interactive frontend components were successfully developed, improving the user experience with responsive designs and dynamic cart features. Attached screenshots confirm the completed implementation.