

# 🌟 Hackathon Day 4: Bandage Marketplace Template 🌟

## Building Dynamic Frontend Components

### 🎯 Objective

On Day 4, the focus is on designing and building dynamic frontend components to display marketplace data fetched from Sanity CMS or external APIs. The goal is to:

- Develop modular and reusable components for scalability.
- Practice real-world techniques for creating responsive and efficient web applications.
- Ensure the frontend effectively renders and organizes data for an intuitive user experience.

This hands-on task will help students better understand how to bridge backend data with a polished, user-friendly interface.

### Key Learning Outcomes

- ✅ **Build Dynamic Frontend Components:** Create components that efficiently fetch and display data from Sanity CMS or APIs.
- ✅ **Reusable and Modular Design:** Develop components that are easy to maintain, extend, and reuse, promoting scalability.
- ✅ **State Management:** Apply effective state management techniques to handle data flow and ensure smooth interactions between components.
- ✅ **Responsive Design & UX/UI Best Practices:** Focus on building responsive layouts and implementing user-centric design principles for a seamless experience.
- ✅ **Real-World Preparation:** Gain hands-on experience by simulating professional workflows, setting the foundation for real-world client projects.

### Key Components to Build

#### 1. Product Listing Component 🛒

- Dynamically display a list of products fetched from the CMS or API.
- Include features like pagination or infinite scrolling for handling large datasets.

TypeScript Interface:




```
interface Cards {  
  image: string;  
  heading: string;  
  alt: string;  
  department: string;  
  originalPrice: string;  
  discountedPrice: string;  
  detailsLink: string;  
}
```

- 

---

## 2. Product Detail Component

- Show detailed product information, including:

- Description 
- Available Sizes 
- Colors 
- Ratings and Reviews 
- Stock Availability 

TypeScript Interface:

```
interface IProducts {  
  id: string;  
  heading: string;  
  subheading: string;  
  image: {  
    asset: {  
      _ref: string;  
      _type: string;  
    };  
  };  
  price: {  
    originalPrice: string;  
    discountedPrice: string;  
  };  
  description: string;  
  size: string[];  
  color: string[];  
  rating: number;  
  reviews: number;  
  stock: number;
```

}

•

---

### 3. Category Component 📁

- Showcase product categories dynamically.
- Allow users to filter and sort products by category.

---

### 4. Cart Component 🛒

- Manage items added to the cart with:
  - Quantity Controls **+****-**
  - Total Price Calculation 💰

---

### 5. Wishlist Component 💖

- Enable users to save products for later.
- Allow adding/removing items from the wishlist.

---

### 6. User Profile Component 👤

- **Display user information, such as:**
  - Name
  - Email ✉️
  - Profile Picture 🖼️
  - Order History 📦

---

### 7. Reviews and Ratings Component ★


- Allow users to leave reviews and ratings for products.
- Display:
  - Average Ratings
  - Review Count

---




### 8. Related Products Component 🔗

- Suggest similar or complementary products dynamically.
- 

## 9. Header Component

- **Create a responsive navigation bar with:**
    - Links to key pages (Home, Shop, Profile, etc.).
    - User Account Dropdown 
- 

## 10. Footer Component

- **Add links to pages like:**
    - About Us
    - Privacy Policy
    - Contact Us 
  - **Include:**
    - Social Media Icons 
    - Newsletter Subscription Form 
- 

## Tools and Technologies

- Frontend Framework: ReactJS, Next.js
  - State Management: Context API
  - CMS/Backend: Sanity CMS or APIs
  - Styling: CSS, Tailwind CSS
  - Responsive Design: Flexbox, Grid, and Media Queries
- 

## Task Notes

- Write clean, modular, and reusable code.
- Test components for scalability and responsiveness.
- Follow professional workflows and best practices for development.

This setup provides a comprehensive and scalable approach to building a responsive marketplace UI while focusing on real-world development standards.