**Day 4 - Dynamic Frontend Components**

1. **Introduction:**

This document outlines the steps taken to develop and integrate dynamic frontend components in the Bandage Online Shopping project. These components were built to provide a dynamic user experience by displaying products and managing interactions seamlessly.

**2. Development Process**

* **Project Setup**: Connected the Next.js project to the Sanity CMS (or other APIs), ensuring data fetching was successful.
* **Dynamic Components**: Built modular components such as:
  + **Product Listing**: Displayed products in a grid layout.
  + **Product Detail Page**: Integrated dynamic routing to show individual product details.
  + **Search and Filters**: Implemented a search bar and filter functionality for product categorization.
* **State Management**: Used React’s useState and useContext for managing local and global states like the cart and user data.
* **Responsive Design**: Styled using Tailwind CSS to ensure mobile-first responsiveness.

**3. Challenges and Solutions**

* **Data Fetching Issues**: Solved by verifying API configurations and adding error handling.
* **Routing Problems**: Addressed dynamic routing issues using Next.js’s getServerSideProps.
* **State Management**: Managed component state through React’s context API for a better data flow.

**4. Best Practices**

* Ensured code modularity and reusability, following DRY principles.
* Applied responsive design techniques for a mobile-friendly layout.
* Implemented efficient state management for smoother data handling.

**5. Conclusion**

The dynamic frontend components are successfully integrated, making the project interactive and responsive. This provides a solid foundation for further features like user authentication and multi-language support.