

Technical Report: Day 4 - Dynamic Frontend Components

1. Steps Taken to Build and Integrate Components

*Project Setup & API Integration

- Initialized a **Next.js** project and configured the folder structure.
- Integrated **Sanity CMS** to fetch dynamic marketplace data.
- Set up **Clerk authentication** to manage user sessions.
- Deployed the project on **Vercel** for live testing and debugging.

*Building Key Components:

- **Product Listing Component:**

Displayed products dynamically using a grid layout.

- **Product Detail Component:**

Implemented dynamic routing in Next.js for individual product pages.

- **Search & Filters:**

Created a search bar and filter panel for enhanced user experience.

- **Cart & Wishlist:**

Added state management for tracking cart and wishlist items.

- **Checkout Flow:**

Developed a multi-step checkout form with mock payment integration.

- **Review Page:**

Added a review section where users can submit and view feedback for products.

***Styling & Performance Optimization**

- Used **Tailwind CSS** for responsive design and styling.
- Implemented **lazy loading** for images to improve performance.
- Applied **pagination and infinite scrolling** for better data handling.

2. Challenges Faced

- Error in **Clerk authentication** setup.
- Issues while fetching data from **Sanity CMS**.
- Deployment problems on **Vercel**.

3. Best Practices Followed

- **Modular Component Design:** Ensured reusability by creating separate components for different functionalities.
- **State Management:** Used **React Context API** for global state handling.
- **Error Handling:** Implemented proper error messages for API failures.
- **Responsive UI:** Designed components to be mobile-friendly and accessible.
- **Code Documentation:** Added comments and structured the code for maintainability.