Day 4 - Technical Report

Overview

Integrated Sanity CMS with a Next.js frontend to build a dynamic, scalable e-commerce website featuring product catalogs and category sections.

Key Components

- **Sanity CMS:**
  - Custom schemas for `product` and `category`.
  - GROQ queries with projections for optimized data fetching.
- **Next.js Frontend:**
  - Dynamic routing for category pages.
  - Reusable components such as `ProductCard` and `CategorySection`.
  - Image handling with fallback placeholders.

Steps Taken

1. **Sanity Setup:**
   - Created custom schemas.
   - Defined relationships between products and categories.
   - Implemented queries to fetch only necessary data (e.g., image URLs).
2. **Next.js Integration:**
   - Developed dynamic routes (e.g., `/category/[categorySlug]`).
   - Built UI components and integrated Tailwind CSS for styling.
   - Handled missing images with placeholder logic.

Challenges & Solutions

- **Rendering Objects as React Children:**
  - **Issue:** React error when an object was rendered instead of a string.
  - **Solution:** Updated queries to project string URLs (e.g., `"imageUrl": image.asset->url`).
- **Missing Category Images:**
  - **Issue:** Some categories lacked images.
  - **Solution:** Implemented fallback using a placeholder image (e.g., `/placeholder.svg`).

Best Practices

- **Modular Code:** Reusable and maintainable components.
- **Optimized Data Fetching:** Use of projections to avoid over-fetching.
- **Responsive Design:** Consistent UI across devices with Tailwind CSS.
- **Error Handling:** Proper management of missing or faulty data.

Conclusion

The project successfully integrated a headless CMS with a modern frontend, addressing key challenges and following industry best practices to deliver a robust e-commerce platform.