Technical Report: E-Commerce Website Development

1. Steps Taken to Build and Integrate Components:

- **Sanity Integration**: Integrated Sanity as the headless CMS to manage product data. Configured a custom schema to store product details such as name, price, description, category, and images.
- **Product Listing Page**: Developed a page that fetches product data from Sanity and displays it dynamically. Implemented features for better user experience.
- **Product Detail Page**: Created a detailed view for each product. This page dynamically loads detailed information like the product's name, price, description, and image based on the selected product.
- **Category-wise Search Page**: Implemented a search functionality that allows users to filter products based on category. This page fetches filtered product data from Sanity using query parameters.

2. Challenges Faced and Solutions Implemented:

- **Challenge 1**: Efficient Data Fetching
  Data fetching from Sanity was optimized by using selective queries that fetch only necessary product details like name, price, and image. This ensured better performance and quicker load times.
  Solution: Efficient queries and optimized API calls were implemented to fetch minimal data for each page.
- **Challenge 2**: Dynamic Routing for Product Detail Page
  Handling dynamic routes for product detail pages was tricky, especially with the slug system. It required making sure each product's page loads based on its specific slug.
  Solution: Used dynamic routing in Next.js with slugs to ensure each product's detailed information loads correctly based on the URL.
- **Challenge 3**: Category-wise Featured Product Search
  Implementing category-based filtering to show relevant products based on the featured category clicked by the user was quite challenging. The main issue was filtering the data dynamically based on the category or feature, while ensuring the correct products are shown based on the user's click.
  Solution: Although it remains a challenge, efforts were made to fetch products based on category filters. However, displaying only relevant products dynamically based on the feature clicked is still a work in progress and remains difficult to fully optimize.

3. Best Practices Followed During Development:

- **Component-Based Architecture**: Employed a modular approach by creating reusable components like product cards, detail sections, and filters.

- **Code Optimization**: Used selective querying to fetch minimal data, ensuring better performance.
- **Responsive Design**: Implemented responsive grid systems to ensure the website works seamlessly across devices.
- **Error Handling and User Experience**: Incorporated error handling to ensure smooth user interactions even during data fetching.
- **Clean Code and Version Control**: Maintained clean, well-commented code and utilized Git for version control to ensure smooth development.

Outcome:

The project successfully integrates Sanity with Next.js to create an e-commerce website with a seamless user experience. Challenges such as efficient data fetching and dynamic routing were addressed smoothly. However, implementing a fully functional **category-wise featured product search** is still a work in progress and remains difficult to implement efficiently.

==========xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx==========