

Hackathon 2025 - Day 04

Amna Aftab

483865

Friday(9am-12pm)

Project Report: E-Commerce Website in Next.js

1. Introduction

In this project, I have developed an e-commerce website using **Next.js**, which allows users to browse products, view detailed product pages, manage a shopping cart, and complete a purchase process. The main components of the website are:

Header: Provides navigation across different pages.

Footer: Displays general information and links.

Product List: Dynamically displays a list of products.

Dynamic Product Pages: Displays detailed information about individual products.

Cart: Allows users to manage products in their shopping cart.

2. Components Developed

2.1 Header Component

The **Header** component is designed to provide site navigation, including links to the homepage, product list, and cart. It helps users easily navigate the e-commerce site from any page. I have implemented a basic navigation menu and included the cart icon to display the number of items in the cart.

- **Functionality:** Displays site logo, navigation links (e.g., Home, Shop, Cart), and cart item count.

2.2 Footer Component

The **Footer** component appears at the bottom of each page and contains general information about the site. It includes links such as "About Us", "Contact", and copyright information. The footer provides users with quick access to important sections of the website.

- **Functionality:** Displays general information and essential links like "Contact Us" and copyright notices.

2.3 Product List

The **Product List** component is responsible for fetching and displaying a collection of products dynamically. It uses external data (e.g., from a CMS like Sanity) to render a grid of product cards, each containing an image, title, and brief description. Each product card links to a detailed page for more information.

- **Functionality:** Displays products dynamically from a data source. Each product links to its detailed page.

2.4 Dynamic Product Page

The **Dynamic Product Page** provides detailed information about a specific product when a user clicks on it. This page dynamically loads data based on the product's unique identifier (e.g., product ID), displaying the product's image, description, price, and other relevant information. Additionally, users can add products to their cart from this page.

- **Functionality:** Fetches and displays detailed information of a single product based on dynamic routing (e.g., product ID). Provides an option to add the product to the shopping cart.

2.5 Cart Component

The **Cart** component manages the shopping cart functionality. It allows users to view items that they have added to the cart, modify quantities, remove items, and proceed to checkout. The cart also updates in real-time as products are added or removed.

- **Functionality:** Allows users to add items to their cart, update quantities, and remove items. Provides a summary of the total cart value and offers a checkout option.
-

3. Technologies Used

- **Next.js:** For building the server-side rendered application with dynamic routing and data fetching.
 - **React:** To build the user interface components.
 - **Redux:** To manage the state of the shopping cart, ensuring a consistent user experience across the app.
 - **Tailwind CSS:** For styling the components, providing a responsive and visually appealing design.
 - **Sanity CMS:** Used to fetch product data dynamically for the product list and product details.
-

4. Challenges Faced

-

State Management: Ensuring proper state management in the cart was one of the primary challenges. Using Redux, I handled adding, removing, and updating items in the cart efficiently.

Dynamic Data Fetching: Another challenge was implementing dynamic data fetching from Sanity CMS, especially for the

dynamic product pages. I had to ensure that the correct product information was fetched and displayed without errors.

Responsive Design: Making the website responsive across various screen sizes and devices was essential. Tailwind CSS helped in making the layout flexible and easy to manage.

-

5. Future Enhancements

- **User Authentication:** Implement user login and registration features so users can save their cart and order history.
- **Payment Integration:** Integrate a payment gateway (e.g., Stripe) to allow users to complete their purchases directly on the website.
- **Shipment Integration:** Integrate shipment (Shipengine) to generate shipping rates, shipment label and shipment tracking Id to track the status of product delivery.
- **Product Filters and Sorting:** Add features like product filtering and sorting based on price, category, and popularity.

6. Conclusion

So far, I have successfully developed the **Header**, **Footer**, **Product List**, **Dynamic Product Page**, and **Cart** components. The website is fully functional, allowing users to browse products, view detailed product pages, and manage their shopping cart. The next steps involve improving the user experience and adding additional features like payment integration and user authentication.

This project demonstrates the power of **Next.js** in building modern, dynamic, and user-friendly e-commerce websites with seamless state management and responsive design.