1. Frontend (Next.js)

The frontend of my project is built using **Next.js**, which will handle:

- **Pages**: These are the components representing various sections of the website (like `HeroSection`, `Header`, `Footer`, `Cart`, `ProductCard`, etc.).
- **Dynamic Content**: The content displayed on my pages (such as text, images, and pricing) is dynamically fetched from **Sanity CMS** (and potentially other APIs).
- **Routing**: Pages are routed using Next.js routing mechanisms, with dynamic pages for products and collections.
- **Styling**: i am using **Tailwind CSS** for styling, which is configured to work seamlessly with Next.js.
- **Components**: Common UI components like buttons, inputs, and forms are abstracted into reusable components (e.g., `Email`, `Header`, `Footer`, etc.).

2. Sanity CMS

My backend content management system (CMS) is **Sanity CMS**, which provides structured content (like images, text, headings, product details, etc.) for my frontend pages. i fetch this content using the **Sanity client** in my Next.js app, and this data is dynamically inserted into components/pages.

Key Functionalities:

- **Fetch Data**: The content is fetched via API calls to **Sanity CMS** (using the `client.fetch()` method) for specific content types, such as:
    - **Landing Page** (hero image, heading, paragraph)
    - **Ceramics Collection** (product cards, images, pricing)
    - **Email Signup** (for the newsletter)
- **Dynamic Fields**: i fetch dynamic fields like product prices, images, and headings, which are then rendered in the frontend.

  Based on the files and components you've provided, here's a high-level system architecture for your Next.js project:

3. 3rd Party APIs

- i am using **3rd Party APIs** for specific features that aren't handled by Sanity CMS. These APIs could be:
- **Payment Processing**: APIs like **Stripe**, **PayPal**, or other third-party payment gateways could be used in the checkout flow (if applicable).
- **Analytics**: You might be using analytics services like **Google Analytics** or **Mixpanel** to track user interactions and page views.
- **Shipping Services**: Integrations with shipping providers (e.g., **Shippo**, **DHL**, etc.) could be used to provide real-time shipping calculations or shipping labels.
- **Social Media**: Integration with social media APIs for sharing content or logging in via social accounts.

- ## Architecture Flow:

- ***Frontend:***

- **Next.js** app serves as the frontend layer.
- The **HeroSection** component pulls dynamic data from Sanity CMS (like hero images, headings, paragraphs).
    - **Header** and **Footer** are reusable components that appear on every page, with routing links for different categories and product pages.
    - **CeramicsCollection** and **ProductCard** are components that render dynamic product details fetched from Sanity.
    - **Cart** functionality handles adding products and reviewing items before checkout.

- ***Backend (Sanity CMS):***

    -
- **Sanity CMS** is used to manage and store data for products, categories, collections, and marketing content (like hero sections and email signups).
- You can manage products, categories (like Ceramics, Chairs, Tables), and landing page data directly within the **Sanity Studio**.
- You fetch data via the **Sanity Client** in the frontend and render it dynamically

+-------------------------+

- | Frontend (Next.js) |
- |------------------------ |
- | - HeroSection |
- | - Header |
- | - Footer |
- | - ProductCard |
- | - Cart |
- | - Email |
- |------------------------ |
- | Fetches Data from: |
- | - Sanity CMS |
- | - 3rd Party APIs |
- +-------------------------+
- |
- | API Calls
- |
- +-------------------------+ +----------------------+
- | Sanity CMS | <--> | 3rd Party APIs |
- |-------------------------| |----------------------|
- | - Manage Products | | - Payment Processing |
- | - Manage Categories | | - Analytics |
- | - Manage Content (e.g., | | - Shipping Services |
- | Hero Section, Emails) | | |
- +-------------------------+ +----------------------+
-
-
-