

Essential Tools and Libraries for Building a Dynamic Next.js Frontend

1.Next.js

- A modern framework for building fast and SEO-friendly web applications.
- Supports features like server-side rendering (SSR) and static site generation (SSG).

2.Sanity CMS

- API-driven **Headless CMS** perfect for managing dynamic content.
- Offers real-time updates and supports flexible content structures.

3.Shadcn UI

- A library of pre-designed UI components seamlessly integrated with Tailwind CSS.
- Customizable and developer-friendly, ideal for creating responsive interfaces.

4.React Icons

- A lightweight library providing multiple popular icon sets.
- Easy-to-use icons to visually enhance your user interface.

5.Tailwind CSS

- A utility-first CSS framework designed for rapid UI development.
- Perfect for creating responsive and modern layouts.

Comprehensive Overview of Website Pages and Their Functionalities

1.Home Page

- The primary landing page showcasing featured products, promotions, and navigation links for easy access to other sections.

2.Login Page

- Enables existing users to securely log in and access personalized features like order tracking and wishlists.

3.Signup Page

- Allows new users to create accounts, unlocking features like saved items and order history.

4.About Page

- Highlights your business mission, values, and story, helping build trust and credibility with users.

5.Shop Page

- Displays the product catalog with filtering and sorting options for seamless browsing and selection.

6.Product Details Page

- Provides comprehensive product information, including descriptions, images, and customer reviews, with an "Add to Cart" option.

7.Add to Cart Page

- Displays selected products with options to modify quantities or proceed to checkout.

8.Billing Detail Page

- Collects user information for payment and shipping during the checkout process.

9.Order Complete Page

- Confirms the order with an invoice, shipping details, and a thank-you message.

10.Wishlist Page

- Lets users save products they wish to purchase later for easy access.

11.Contact Us Page

- Provides a form or contact details for users to reach out with inquiries or support needs.

12.FAQ Page

- Answers common questions to provide quick help to users regarding products and services.

13.Blog Page

- Shares articles, news, and updates to engage visitors and improve SEO.

Third-Party APIs for Enhanced Functionality

1.Stripe / QuickPay (Payment Gateway APIs)

- I am using **Stripe** and **QuickPay** for secure and seamless payment processing.
- These APIs support various payment methods like credit/debit cards and mobile wallets.
- They handle transaction processing, refunds, and fraud detection to ensure a safe payment experience for users.

2.Shippo / ShipEngine (Shipping APIs)

- I am integrating **Shippo** and **ShipEngine** for shipping functionalities.
- These APIs provide real-time shipping rates, label generation, and package tracking.
- They help manage shipping logistics and ensure timely delivery, along with real-time tracking for customers.

Why I am Using Sanity CMS for My Project

1. Powerful Headless CMS

- Sanity is a powerful and flexible headless CMS that allows easy management and delivery of content across various platforms.
- It integrates seamlessly with front-end frameworks like Next.js, ensuring smooth data flow and content management.

2. Product Listing and Details

- I am using Sanity to manage and store product listing data and detailed product information.
- This allows for easy updates and content management without affecting the frontend, ensuring an efficient workflow.

3. User Authentication (Login/Signup)

- I have integrated login and signup functionality with Sanity to manage user accounts and ensure secure authentication.
- All login/signup data is stored in Sanity, providing an organized and secure approach to user management.

4.Customer Data Management

- When customers enter their details, I store this information in Sanity, providing a streamlined way to manage customer data.
- This centralizes the storage of customer information, allowing for better customer support and future communications.

5.Order Data Storage

- When a customer completes an order, I store all relevant order data in Sanity, making it easily accessible for tracking and management.
- This enables a central location for managing order details and customer history.

By using Sanity, I ensure that all critical data (product listings, orders, user details) is stored in one centralized, scalable, and easy-to-manage system, making it an ideal choice for my project.

Next.js App

Product list page

Product detail page

Fetch products from sanity

Fetch product details from
sanity (slug)

Display list of products on
shop page

Show product description
and info

Order and customer data ,
login / singup

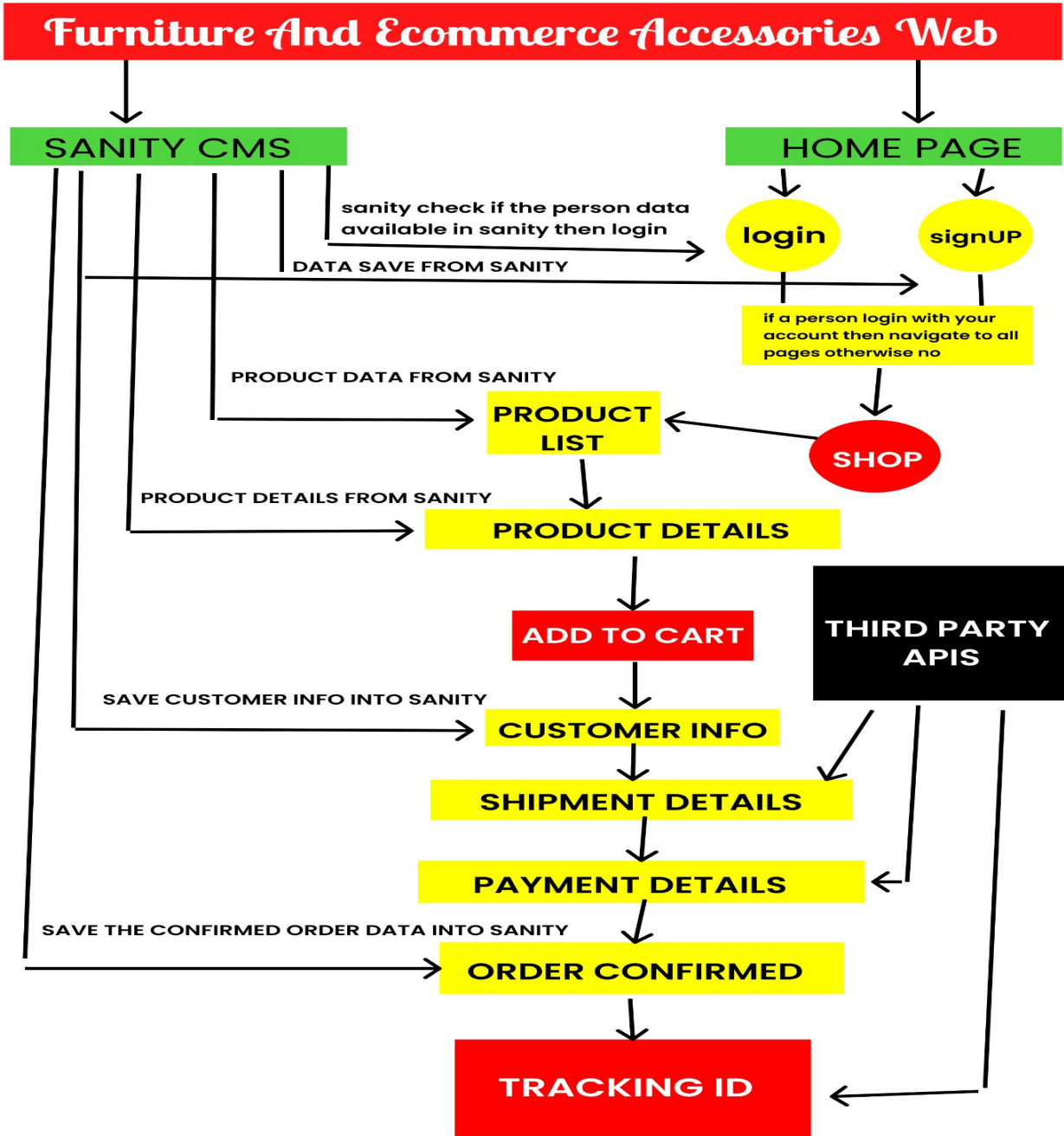
Save order data into sanyi
after confirmation

Save customer info into
sanity

Register new customer and
save data from sanity

Login user after checking
sanity if user data is availabe

WEBSITE ARCHITECTURE



ENDPOINT NAME	HTTP METHOD	DESCRIPTION	RESPONSE EXAMPLE
User Signup	POST	Add a user data in sanity CMS	{name:"ali",email:"abc@gmail.com",password:1234}
Get Product List	GET	fetches all product data	[{id:1,name:"chair",price:100,}]
Get Single Product Data	GET	fetch specific product data using id	{id:1,name:"chair",price:100,}
Post Order Confirmation	POST	create a new order in sanity CMS	{id:1,name:"ali",product:"chair"}
Update Order Status	PUT	Update the status of an order	{status:"shipped",success:true}
Shipping And Tracking	GET	fetch shipping tracking data	{trackinginfo:{carrier:"UPS",Status:"pending"}}
Payment Gateways	POST	Initiate payment via stripe etc..	{ "status": "success", "transaction_id": "txn_123456789" }