System Architecture

Frontend Layer

Next.js 15 with TypeScript

Purpose: This layer is responsible for rendering the website, handling routing, and ensuring client-side interactivity. It consumes APIs from the backend and displays dynamic content to users.

Key Features:

- ❖ Pages: About, Product, Cart, Checkout, Contact, Shop, etc.
- ❖ Dynamic routing for product pages using getStaticProps and getStaticPaths.
- ❖ Integration with Tailwind CSS for styling.
- ❖ Components like AboutHero, ProductCard, ShoppingCart, CheckoutForm, and BlogHero for modular UI.

Sanity CMS

- ❖ Purpose: This layer manages the content for the website, such as product details, blog posts, and team information.

Functionality:

- ❖ Schemas for products, blogs, testimonials, and other content types.

- API endpoints for fetching structured data from Sanity.

Integration:

- Frontend fetches product and content data from Sanity’s GROQ API.

- Backend APIs Layer

- Next.js API routes

- Purpose: To create custom backend APIs for handling specific business logic.

Key APIs:

- **Product Data API:** Fetch product details from Sanity.

- **Shipment Tracking API**: Fetch and update shipment details using third-party services like USPS, UPS, or FedEx.

- **User Authentication API**: Handles user login and registration.

Third-Party Integrations Layer

- **Third-Party APIs:**

- **Shipment Tracking API:** Integrated with APIs such as USPS, UPS, or FedEx for real-time shipment tracking.

- **Payment Gateway:** Integrated with platforms like Stripe or Razorpay for secure payments.

- Provides additional functionality that complements the website’s core features.

- Handles payment transactions and delivery logistics.

Database Layer

- Local Storage (for temporary cart data) and Sanity CMS (for persistent data).

Purpose:

- Sanity CMS stores product, blog, and user content.

- Local Storage temporarily holds cart and session data.

- Deployment & Hosting Layer

- Frontend Hosting: Vercel for Next.js deployment.

CMS Hosting: Sanity CMS cloud hosting.

- Ensures that the application is accessible with optimized performance and scalability.

- Provides CI/CD pipelines for smooth updates.

System Flow:

- The user interacts with the Frontend (Next.js).

- The frontend fetches data from Sanity CMS for product, blog, and other dynamic content.

- User actions like adding products to the cart or checking out interact with custom Next.js APIs.

During checkout:

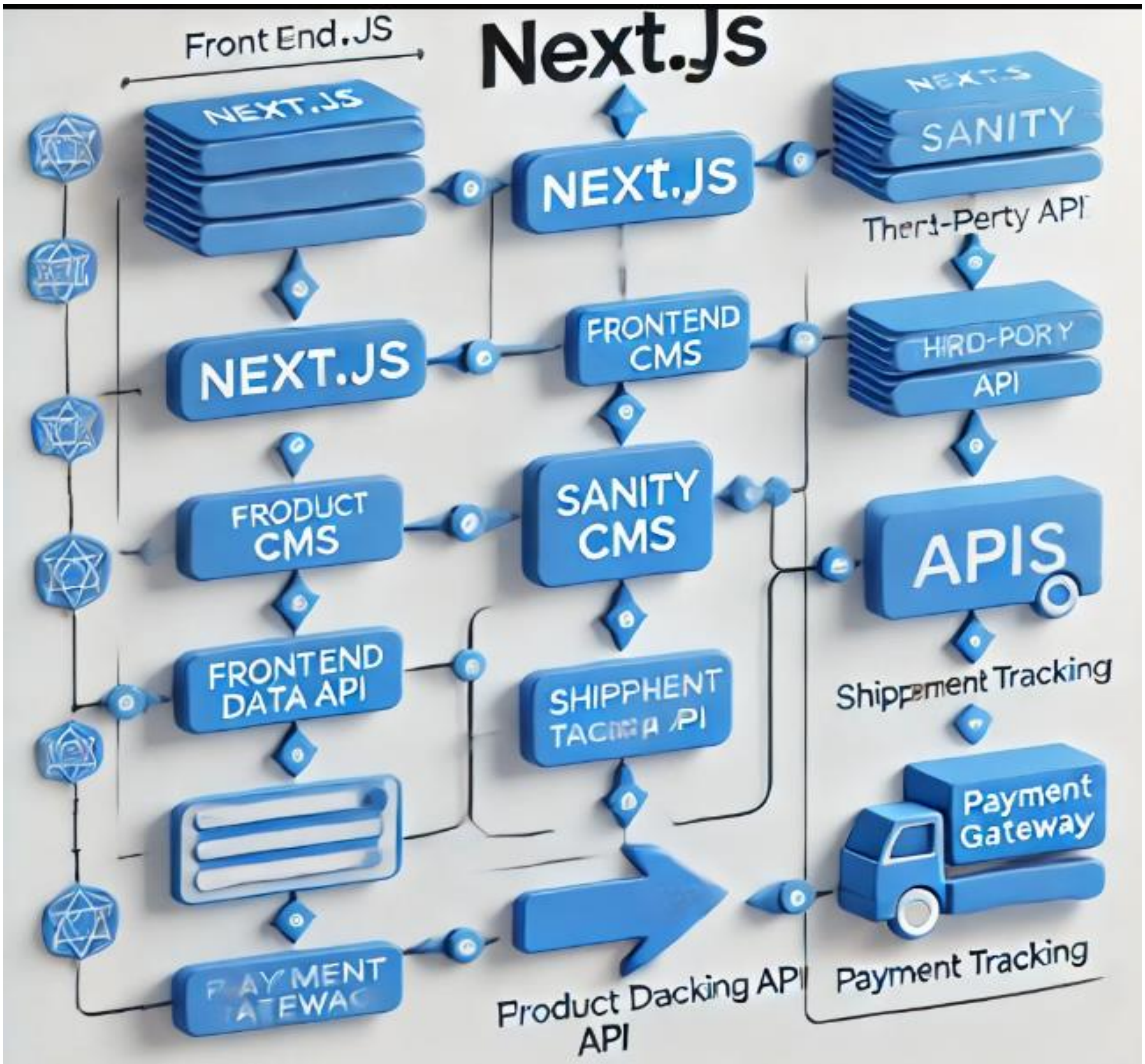
- Product availability is verified via the Product Data API.

- Shipment details are handled by the Shipment Tracking API.

- Payments are processed through a Payment Gateway.

- The system updates the order status and shipment tracking information via the backend and displays it to the user.

Diagram Representation


Front End.JS
Next.Js
NEXT.JS
NEXT.JS
NEXT.JS
SANITY
Thert-Perty API
NEXT.JS
FRONTEND CMS
HIRD-PORY API
FRODUCT CMS
SANITY CMS
APIS
FRONTEND DATA API
SHIPPHENT TACI API
Shippment Tracking
Payment Gateway
AYMENT ATEWAC
Product Dacking API API
Payment Tracking