

SHOP.CO - Technical

System Architecture

- Frontend
- Backend
- Database
- Hosting
- Authentication

→ Frontend :

- * Use Next.js because fast and SEO-friendly.
- * Use Tailwind CSS styling to responsive design.
- * For state management the Redux or React context API can be use.

→ Backend :

- * Sanity CMS will allow sellers to easily manage products.
- * For firebase firestore or Supabase databases, to update data in real time.

→ Hosting :

- * Deploy to vercel because its perfect for Next.js.
- * Sanity CMS cloud-based, So don't need extra hosting.

Workflows

- User Registration & Login
 - Firebase Auth for email/password & social logins
- Products Listing
 - Admin can add, update, or remove clothes using Sanity CMS.
- Shopping Cart
 - Users can add/remove items, view total price.
- Checkout & Payment
 - Stripe API integration for secure payments
- Orders Management
 - Users can track their orders, sellers can update order status.

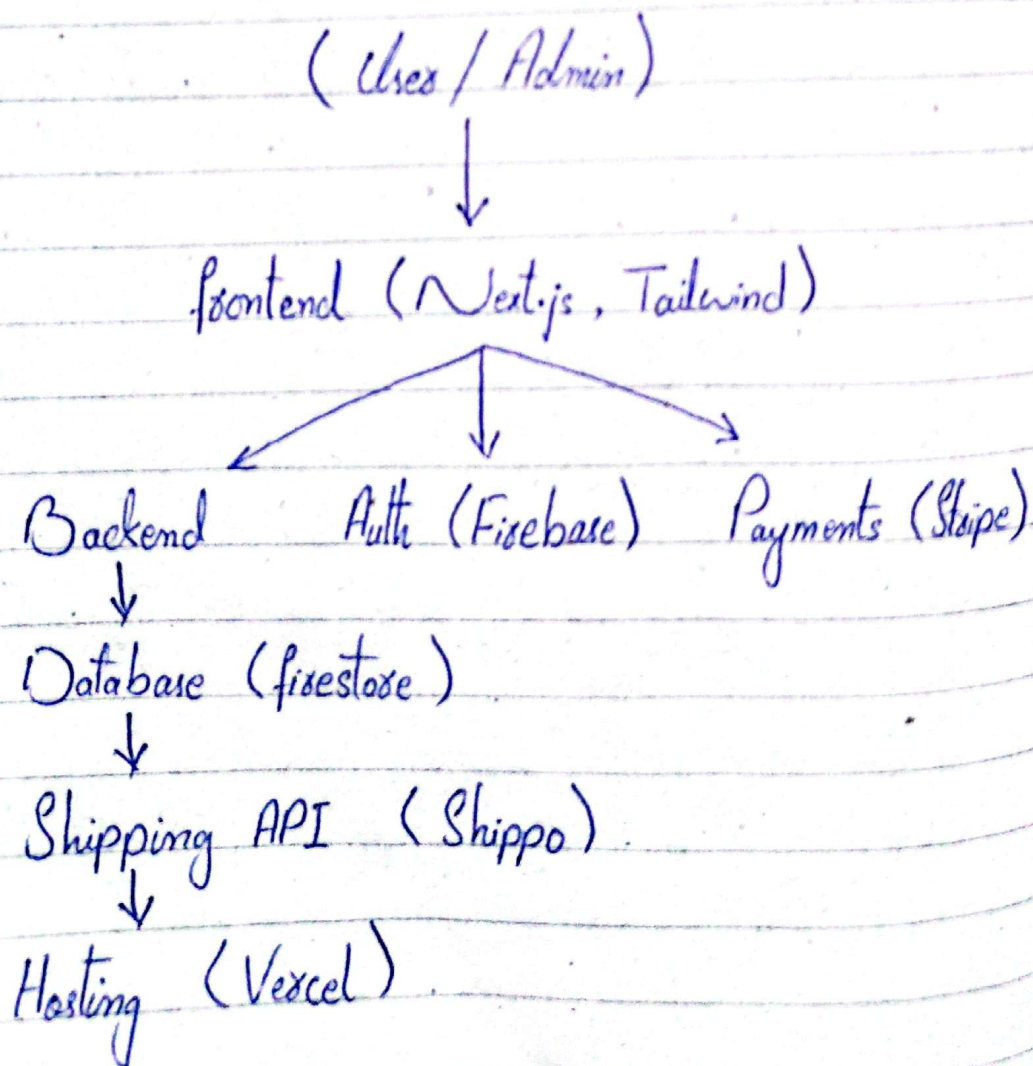
API Requirements

- Sanity CMS API
 - for managing products & categories
- Stripe API
 - for handling payments securely
- Firebase Auth API
 - for user authentication
- Shipping API (e.g: Shippo or EasyPost)
 - for tracking deliveries

Next Step.

- * finalize the architecture & tool choices.
- * Create a simple wireframe of how the website will function.
- * Start setting up the tech stack
(Next.js + Sanity + Stripe)

SHOP.CO technical foundation flow chart.



Next Step.

- * finalize the architecture & tool choices.
- * Create a simple wireframe of how the website will function.
- * Start setting up the tech stack
(Next.js + Sanity + Stripe)

SHOP.CO technical foundation flow chart.

