**🌟 Hackathon Day 2 Technical Foundation**

**🛋️ Décor and Furniture Website**

**Furniture and Decor E-commerce Platform Architecture**

Our furniture and decor e-commerce platform, **AVION**, is built using cutting-edge technologies, including:

1. 🚀 **Next.js 14** for a fast and scalable frontend
2. 🔧 **TypeScript** for robust and maintainable code
3. 📝 **Sanity CMS** for seamless content management
4. 🎨 **Tailwind CSS** for a customized and responsive design
5. 💻 **Node.js** for a powerful and efficient backend

**Key Components**

* 🌐 **Website pages and components** for a smooth user experience
* 📊 **Admin functionalities** for efficient management
* 🛠️ **Sanity CMS** for seamless data management
* 💳 **Secure payment gateway integration**
* 🔍 **Robust search functionality** for easy product discovery

**🏗️ System Architecture**

**Frontend (Next.js)**

Utilizes **Next.js 14** with **TypeScript** for:

* ⚡ **Server-side rendering (SSR)**
* 🛣️ **Fast and dynamic routes**

📍 **Routes**:

* 🏠 **Home**
* 📜 **About**
* 🛒 **Products**
* 📝 **Products/[slug]** (Dynamic Route for Product Details)
* 🛍️ **Cart**
* ✅ **Checkout**

**Components**

**🛋️ Product Components**:

* 🖼️ ProductCard.tsx
* 📖 ProductDetails.tsx

**🔄 Reusable Components**:

1. 🔝 UpNavbar
2. 🔻 DownNavbar
3. 🎯 HeroSection
4. 🔗 NavbarLinks
5. 🛠️ BrandCard
6. 🖌️ BrandSection
7. 🌟 FeatureCeramics
8. 🛒 Cart
9. 🔐 SignUp\_Login
10. ✅ CheckOut
11. 💳 Payment
12. 🚚 Shipment
13. 👤 UserDetail
14. 📝 PaymentForm

(All files have a .tsx extension.)

**Sanity (Headless CMS)**

**Sanity Studio**

Manages content and data, including:

**Data Models**

1. 🛒 **Products**: Names, prices, images, categories, and stock
2. 👥 **Users**: Login credentials and order history
3. 📝 **Orders**: Items purchased, quantities, and shipping information
4. 🚚 **Shipments**: Tracks shipments using the **Shippo API**
5. 📊 **Analytics**: Total sales, revenue, and product analytics

**Data Management**

* 🗂️ **Schemas**: Define data structures for Orders, Products, Users, Shipments, Inventory, etc.
* 🔎 **GROQ Queries**: Fetch real-time data from Sanity's content lake

**🌐 API Management**

**Third-Party APIs**

Purpose: Integrate external services for enhanced functionality

**APIs**:

1. 💳 **Stripe API**: Secure payment processing
2. 🚚 **Shippo API**: Real-time shipment tracking

**API Integration**

* 📡 Utilizes **Axios** for seamless communication with third-party services

**💳 Payment Gateway**

**Stripe Integration**

Purpose: Handle payments securely

**Features**:

1. 🔐 Utilizes **Stripe Elements** for secure payment detail collection
2. 💳 Processes payments securely via **Stripe API**

**🚚 Shipment Tracking**

**Shippo API Integration**

Purpose: Track orders and shipments

**Features**:

1. 🧭 Displays shipment tracking information on the user's order history page
2. 🔄 Provides real-time tracking updates using **Shippo API**

**🔄 Workflow**

**User Workflow**

1️⃣ **Visit Home Page**

* 🛍️ Users browse products fetched dynamically from Sanity via API

2️⃣ **Add to Cart**

* ➕ Users click "Add to Cart" and proceed to the checkout page

3️⃣ **Checkout**

* ✅ Users provide necessary details (e.g., shipping and payment information)

4️⃣ **Payment Processing**

* 💳 Payment information is sent to **Stripe API** via PaymentForm

5️⃣ **Shipment Tracking**

* 🚚 Shipment tracking details are fetched using **Shippo API** and displayed in Shipment.tsx

6️⃣ **User Portal**

* 👤 Users can log in to access their portal and view their order/shipment history via UserDetail.tsx

**Conclusion**

This detailed architecture ensures that the platform is well-structured for scalability and maintainability. The combination of reusable components, dynamic pages, and the Sanity CMS makes the platform flexible and easy to manage.

The integration of third-party APIs enhances the frontend-backend interaction, while user-friendly workflows provide an intuitive and seamless experience. This user-centric design approach ensures the platform remains robust and adaptable for various applications.