PLANNING THE TECHNICAL FOUNDATION

Website Concept: ***Summer New Collection***

Brand Name: Bandage
Theme: A focus on modern, vibrant, and lightweight fashion, emphasizing summer comfort and style.

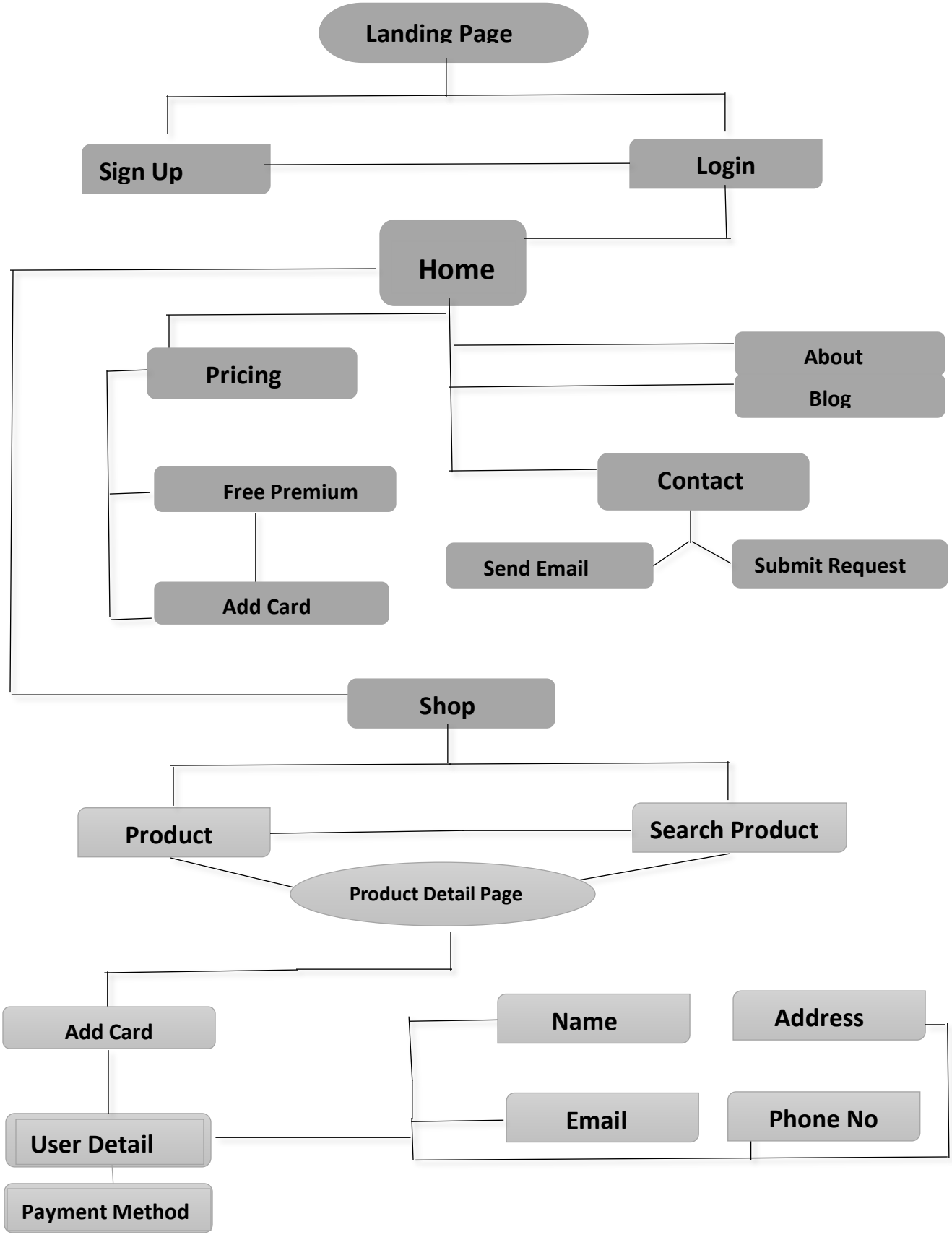
Day 2 Hackathon Tasks:

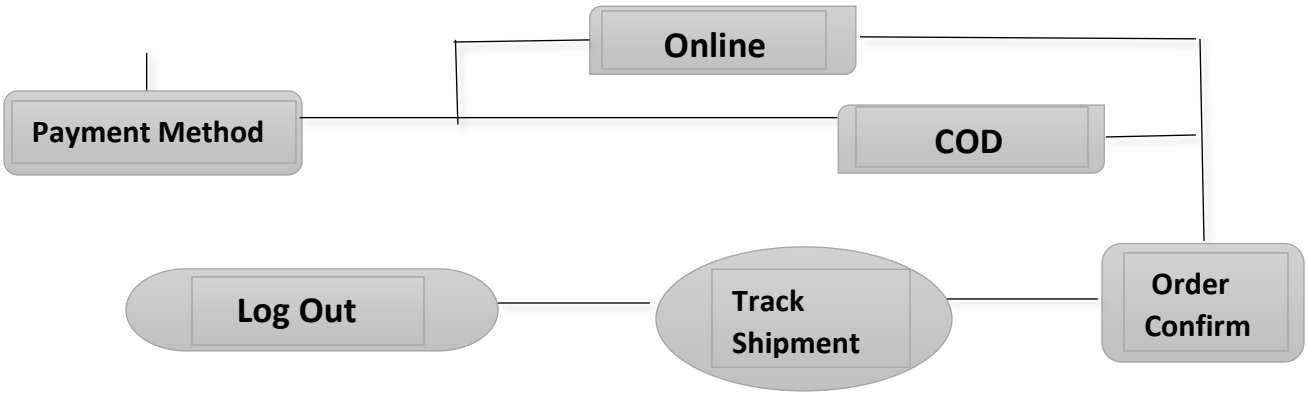
Bandage is an e-commerce platform offering trendy, affordable summer fashion for individuals aged 18-35. Focused on lightweight apparel and accessories, it combines comfort, style, and sustainability with unique branding and eco-friendly packaging. Products include dresses, shirts, shorts, sunglasses, and more, with collections refreshed monthly. The platform's structured data schema covers Products, Orders, Customers, and Delivery Zones to ensure a smooth shopping experience.

FrontEnd:

The marketplace frontend will feature a visually appealing, user-friendly design using Next.js for speed and SEO, with responsive layouts powered by Tailwind CSS or Bootstrap. Key pages include a **Home** page for promotions, **Product Listing** and **Details** pages for browsing and detailed views, a dynamic **Cart** with real-time updates, a secure **Checkout** for payments, and an **Order Confirmation** summarizing the purchase with an order ID.

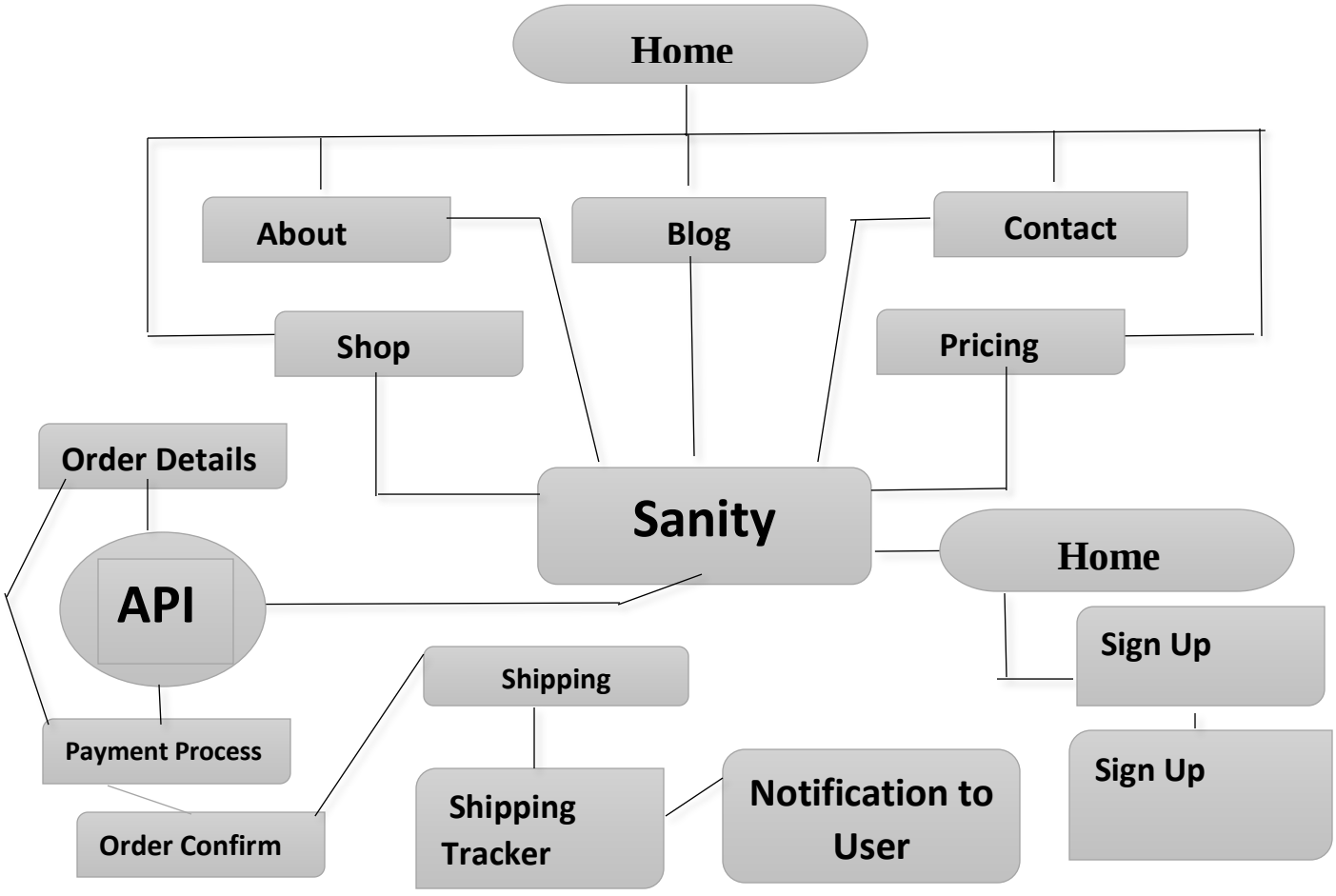
System Architecture:


Landing Page
Sign Up
Login
Home
Pricing
About
Blog
Free Premium
Contact
Add Card
Send Email
Submit Request
Shop
Product
Search Product
Product Detail Page
Add Card
Name
Address
User Detail
Email
Phone No
Payment Method



BackEnd:

The backend, using Sanity CMS, will manage product, customer, and order data with schemas for essential fields like name, price, and stock. Sanity Studio will facilitate content management with validation rules. Third-party APIs will be integrated for shipment tracking (Shippo or AfterShip) and secure payments (HBL). Email notifications will be handled through Send Grid for customer updates.



API Endpoints:

	Product	Order	Shipment
Endpoint	/products	/orders	/shipment
Method	GET	POST	GET
Purpose	Fetch all available products.	Create a new order.	Fetch order tracking details.
Response	{ "id": 1, "name": "Product A", "price": 100, "stock": 20 }	{ "customer": { "name": "John" }, "items": [...], "total": 150 }	{ "order ID": 123, "status": "Shipped", "ETA": "2 days" }

Data Schema Design:

```
export default {
  name: 'product',
  type: 'document',
  fields: [
    { name: 'name', type: 'string', title: 'Product Name' },
    { name: 'price', type: 'number', title: 'Price' },
    { name: 'stock', type: 'number', title: 'Stock Level' }
  ],
};
```

Conclusion:

In summary, the technical planning for the marketplace involves defining clear frontend and backend requirements, integrating third-party APIs, and designing a robust system architecture. Key steps include creating API endpoints, designing data schemas, and documenting workflows.