Planning The Technical Foundation (Day 02)

General E-Commerce Marketplace

Technical Requirements:

1. Frontend Development

Key Technologies

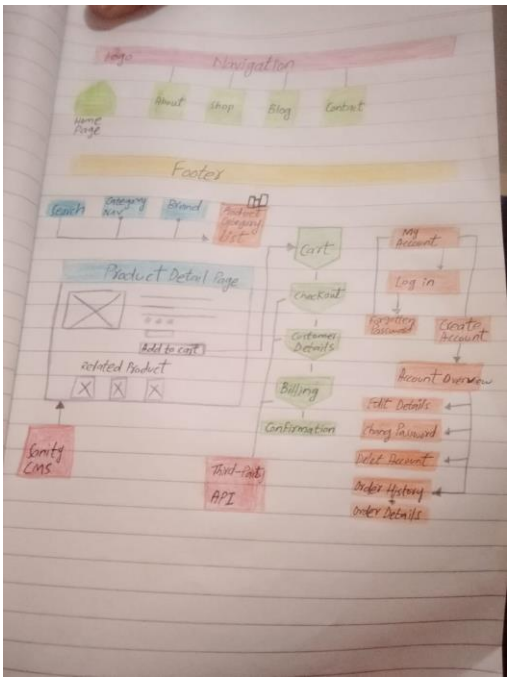
- **React.js / Next.js** (For building the user interface, with Next.js offering server-side rendering for faster load times and SEO benefits)
- **Tailwind CSS / SCSS / Styled-components** (For styling the UI with responsive design)
- **JavaScript / TypeScript** (For dynamic content and user interaction)
- **Responsive Design** (Ensure compatibility across devices: mobile, tablet, desktop)

2. Backend Development

Key Technologies

- **Node.js** (For building API endpoints and handling business logic)
- **Express.js** (For REST API routing and middleware management)

Design System Architecture



Sanity's Role for the Product Detail Page

1. **Content Management System (CMS):**
   - Acts as the backend where you manage product details.
   - Allows admins or sellers to create, update, and delete product information.
2. **Structured Content Delivery:**
   - Stores structured product data such as:
     - Product Name
     - Description
     - Price
     - Categories
     - Tags
     - Variants (e.g., size, color)
     - Media (images, videos)
   - Provides APIs (GROQ or GraphQL) to fetch this data dynamically.
3. **Dynamic Updates:**
   - Any changes made in Sanity (e.g., updating prices or descriptions) reflect immediately on the Product Detail Page without requiring redeployment.
4. **Localization and SEO:**
   - Supports multilingual content for global reach.
   - Manages metadata (title, description) to optimize SEO.

Key Roles of Third-Party APIs in the Checkout Process

1. Payment Gateway APIs

- Examples: **Stripe**, **PayPal**, **Razorpay**, **Square**.
- **Role:**
    - Securely process payments made by buyers using credit cards, debit cards, net banking, UPI, or digital wallets.
    - Handle sensitive payment information (e.g., card details) without storing it on your servers.
    - Support recurring payments for subscriptions if required.

---

2. Tax Calculation APIs

- Examples: **TaxJar**, **Avalara**.
- **Role:**
    - Automatically calculate applicable taxes (e.g., VAT, GST) based on buyer and seller locations.
    - Ensure compliance with local tax regulations, especially in marketplaces with multiple regions.

---

3. Shipping APIs

- Examples: **Shippo**, **EasyPost**, **FedEx API**, **DHL API**.
- **Role:**
    - Provide real-time shipping rates during checkout.
    - Generate shipping labels and tracking IDs for orders.
    - Offer shipping options like standard, express, or same-day delivery.

---

4. Address Validation APIs

- Examples: **Google Maps API**, **SmartyStreets**.
- **Role:**
    - Validate and auto-complete buyer shipping addresses.
    - Reduce errors in address entry, ensuring successful deliveries.

---

5. Fraud Detection APIs

- Examples: **Signifyd**, **FraudLabs Pro**, **Sift**.
- **Role:**
    - Analyze transactions to detect and flag suspicious activities.
    - Protect buyers and sellers from chargebacks and fraudulent orders.

---

6. Currency Conversion APIs

- Examples: **Open Exchange Rates**, **CurrencyLayer**.
- **Role:**
  - Display product prices and checkout totals in the buyer's local currency.
  - Update exchange rates in real-time for accurate conversion.

Benefits of Using Third-Party APIs

- **Speed and Efficiency:** Reduce development time by leveraging existing solutions.
- **Compliance:** Ensure adherence to industry standards (e.g., PCI DSS for payment security).
- **Scalability:** Handle high transaction volumes without performance degradation.
- **Global Reach:** Support multiple payment methods, currencies, and shipping carriers.

API requirements:

---

1. Authentication and Authorization API

API Endpoints:

- `POST /api/auth/login`
- `POST /api/auth/register`
- `POST /api/auth/logout`
- `GET /api/auth/roles`

2. Order and Checkout API

API Endpoints:

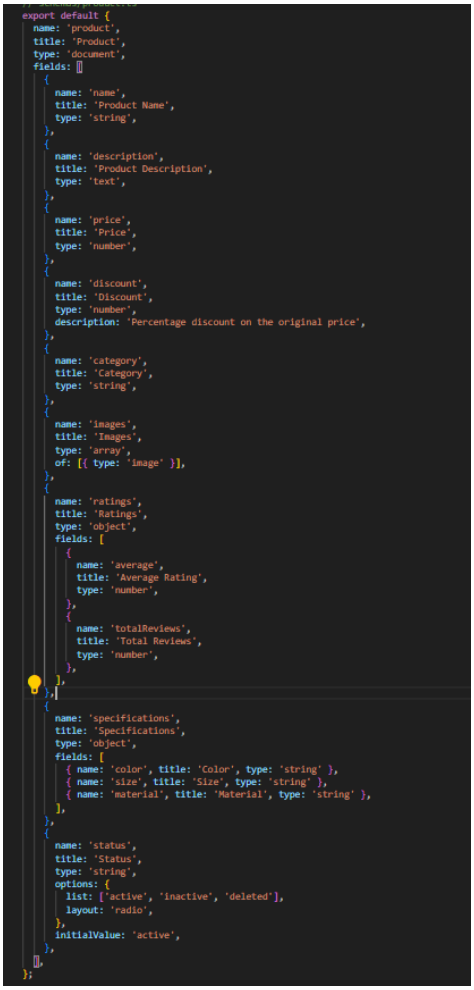
- `POST /api/cart/add`
- `GET /api/cart`
- `POST /api/orders/checkout`
- `GET /api/orders/{order_id}/status`

3. Shipping and Logistics API

API Endpoints:

- `GET /api/shipping/rates`
- `POST /api/shipping/validate-address`
- `GET /api/shipping/track/{tracking_id}`

Sanity Poduct Schema:



```
export default {
  name: 'product',
  title: 'Product',
  type: 'document',
  fields: [
    {
      name: 'name',
      title: 'Product Name',
      type: 'string',
    },
    {
      name: 'description',
      title: 'Product Description',
      type: 'text',
    },
    {
      name: 'price',
      title: 'Price',
      type: 'number',
    },
    {
      name: 'discount',
      title: 'Discount',
      type: 'number',
      description: 'Percentage discount on the original price',
    },
    {
      name: 'category',
      title: 'Category',
      type: 'string',
    },
    {
      name: 'images',
      title: 'Images',
      type: 'array',
      of: [{ type: 'image' }],
    },
    {
      name: 'ratings',
      title: 'Ratings',
      type: 'object',
      fields: [
        {
          name: 'average',
          title: 'Average Rating',
          type: 'number',
        },
        {
          name: 'totalReviews',
          title: 'Total Reviews',
          type: 'number',
        },
      ],
    },
    {
      name: 'specifications',
      title: 'Specifications',
      type: 'object',
      fields: [
        { name: 'color', title: 'Color', type: 'string' },
        { name: 'size', title: 'Size', type: 'string' },
        { name: 'material', title: 'Material', type: 'string' },
      ],
    },
    {
      name: 'status',
      title: 'Status',
      type: 'string',
      options: {
        list: ['active', 'inactive', 'deleted'],
        layout: 'radio',
      },
      initialValue: 'active',
    },
  ],
};
```