Marketplace Technical Foundation

Frontend Requirements

1. **Overview**
   The e-commerce platform frontend should provide a user-friendly, responsive interface for browsing products and completing orders on both mobile and desktop devices.

2. **User Interface (UI)**

   1. **Navigation**: Intuitive layout, clear categories, filters, and a visible search bar.
   2. **Branding**: Consistent use of brand colors and readable designs.
   3. **Typography**: Legible fonts with consistent sizes and styles.
   4. **Header/Footer**: Includes essential links (e.g., Home, Categories, Cart, Support).

3. **Responsive Design**

   Mobile, tablet, and desktop compatibility with layouts adjusting based on screen size.

   Breakpoints:

   1. Mobile: Single-column layout.
   2. Tablet: Two-column layout.
   3. Desktop: Grid layout for products.

   Touch-friendly buttons and optimized hover effects.

4. **Essential Pages**

   1. **Home Page**: Promotions, categories, trending products, and call-to-action buttons.
   2. **Product Listing Page**: Product grid with filters, sorting, and pagination/infinite scroll.
   3. **Product Details Page**: Detailed view with zoomable images, reviews, and related products.
   4. **Cart Page**: Displays selected items with total price and options to update/remove products.
   5. **Checkout Page**: Forms for user login, shipping, billing, and payment with a summary.
   6. **Order Confirmation Page**: Order details, tracking info, and delivery time.

5. **Functionality**

   Cart Management: Add/remove items, update quantities, and display cart count.

   Checkout Flow: Secure payment integration and real-time validation for forms.

Sanity CMS Backend Integration

Sanity CMS is the backend system used to manage data like products, orders, and customers, providing structured content for dynamic updates.

Responsibilities

1. **Product Management**: Store product details (e.g., name, price, stock, categories).
2. **Customer Data**: Manage user profiles and order histories.
3. **Order Records**: Track order status, shipping, and payment details.

Schemas

1. **Product Schema**: Stores name, price, categories, images, stock, and discounts.
2. **Category Schema**: Organizes products into categories with SEO-friendly slugs.
3. **Customer Schema**: Contains customer profiles, shipping addresses, and orders.
4. **Order Schema**: Tracks customer orders, products, quantities, prices, and shipping details.

```
export default {
  name: 'product',
  title: 'Product',
  type: 'document',
  fields: [
    {
      name: 'name',
      title: 'Product Name',
      type: 'string',
    },
    {
      name: 'slug',
      title: 'Slug',
      type: 'slug',
      options: {
        source: 'name',
        maxLength: 200,
      },
    },
    {
      name: 'description',
      title: 'Description',
      type: 'text',
    },
    {
      name: 'price',
      title: 'Price',
      type: 'number',
    },
    {
      name: 'priceWithoutDiscount',
      title: 'Price Without Discount',
      type: 'number',
    },
    {
      name: 'discountPercentage',
      title: 'Discount Percentage',
      type: 'number',
    },
    {
      name: 'category',
      title: 'Category',
      type: 'reference',
      to: [{ type: 'category' }],
    },
    {
      name: 'images',
      title: 'Product Images',
      type: 'array',
      of: [{ type: 'image' }],
    },
    {
      name: 'stockQuantity',
      title: 'Stock Quantity',
      type: 'number',
    },
    {
      name: 'sku',
      title: 'SKU (Stock Keeping Unit)',
      type: 'string',
    },
    {
      name: 'brand',
      title: 'Brand',
      type: 'string',
    },
    {
      name: 'rating',
      title: 'Rating',
      type: 'number',
    },
    {
      name: 'ratingCount',
      title: 'Rating Count',
      type: 'number',
    },
    {
      name: 'tags',
      title: 'Tags',
      type: 'array',
      of: [{ type: 'string' }],
    },
    {
      name: 'sizes',
      title: 'Available Sizes',
      type: 'array',
      of: [{ type: 'string' }],
    },
    {
      name: 'colour',
      title: 'Available Colours',
      type: 'array',
      of: [{ type: 'string' }],
    },
  ],
};
```

Frontend Integration

Use **GROQ queries** to fetch product, category, and order data dynamically.

Support real-time updates for instant content changes.

Third-Party API Integration

Overview

Integrates APIs for shipment tracking, payment processing, and inventory management to enhance backend functionality.

APIs

1. **Shipment Tracking API**: (e.g., AfterShip) Fetch and display real-time order tracking details.
2. **Payment Gateway API**: (e.g., Stripe, PayPal) Handle secure payment processing with endpoints to create and confirm payments.

API Features

1. Order tracking displays shipment status and delivery estimates.
2. Payment gateways ensure secure payment collection and card/token validation.

API Requirements

Product Management APIs

- **Get All Products**: Fetch all product data (e.g., name, price, stock, categories).
- **Get Product Details**: Retrieve specific product information by ID.

Order Management APIs

- **Create Order**: Save customer orders with product details, shipping address, and total price.

Cart Management APIs

- **Get Cart**: Retrieve the items in a user's cart.
- **Add/Remove Item**: Add or delete products from the cart.

Payment APIs

- **Create Payment Intent**: Initialize payment with order details.
- **Confirm Payment**: Finalize payment and return the transaction status.

Security and Testing

- **Security**: Use tokenization for sensitive data (e.g., card info) and ensure compliance with data protection laws (e.g., GDPR).
- **Testing**: Test all APIs in a sandbox environment with proper error handling for failed transactions or API issues.