MARKETPLACE - TECHNICAL FOUNDATION

Overview:

This document outlines the core technical framework for building a marketplace platform. It covers key routes, API integrations, and the end-to-end flow of order processing, providing a robust and scalable structure for the system.

Tech Stack:

1. **Frontend**:
   - Built with **Next.js**, leveraging server-side rendering for fast page loads and optimized user interfaces.
   - Ensures smooth navigation and dynamic content updates for a seamless shopping experience.

2. **Backend**:
   - Powered by **Sanity CMS** to efficiently manage content like products, orders, and customer data.
   - Provides flexibility for future enhancements and integrations.

3. **Third-Party APIs**:
   - Integrates external APIs for functionalities like shipment tracking, payment processing, and enhanced features.

Operational Workflow:

1. **Home Page (/)**:
   - Fetches and displays a list of available products through an external API.
   - Allows users to explore product-specific pages for more details.

2. **Product Detail Page (/products/{product_id}):**
   - Shows detailed product information such as descriptions, prices, and stock availability.
   - Enables users to add products to their shopping cart with a single click.

3. **Shopping Cart (/cart):**
   o Displays items added to the cart, along with the total cost.
   o Provides options to update, remove, or proceed with items.
   o Cart data is stored and managed in the Cart Schema within Sanity CMS.

4. **Checkout Page (/checkout):**
   o Collects user shipping details and reviews the cart before finalizing the purchase.
   o Displays the total order price and confirms the order placement.

5. **Order Management:**
   o Once an order is confirmed, a Shipping ID is generated.
   o Order status progresses through:
      1. **Processing**: Order is being prepared.
      2. **Shipped**: Order has left the warehouse, and a Tracking ID is issued.
      3. **Delivered**: Order is delivered to the customer.
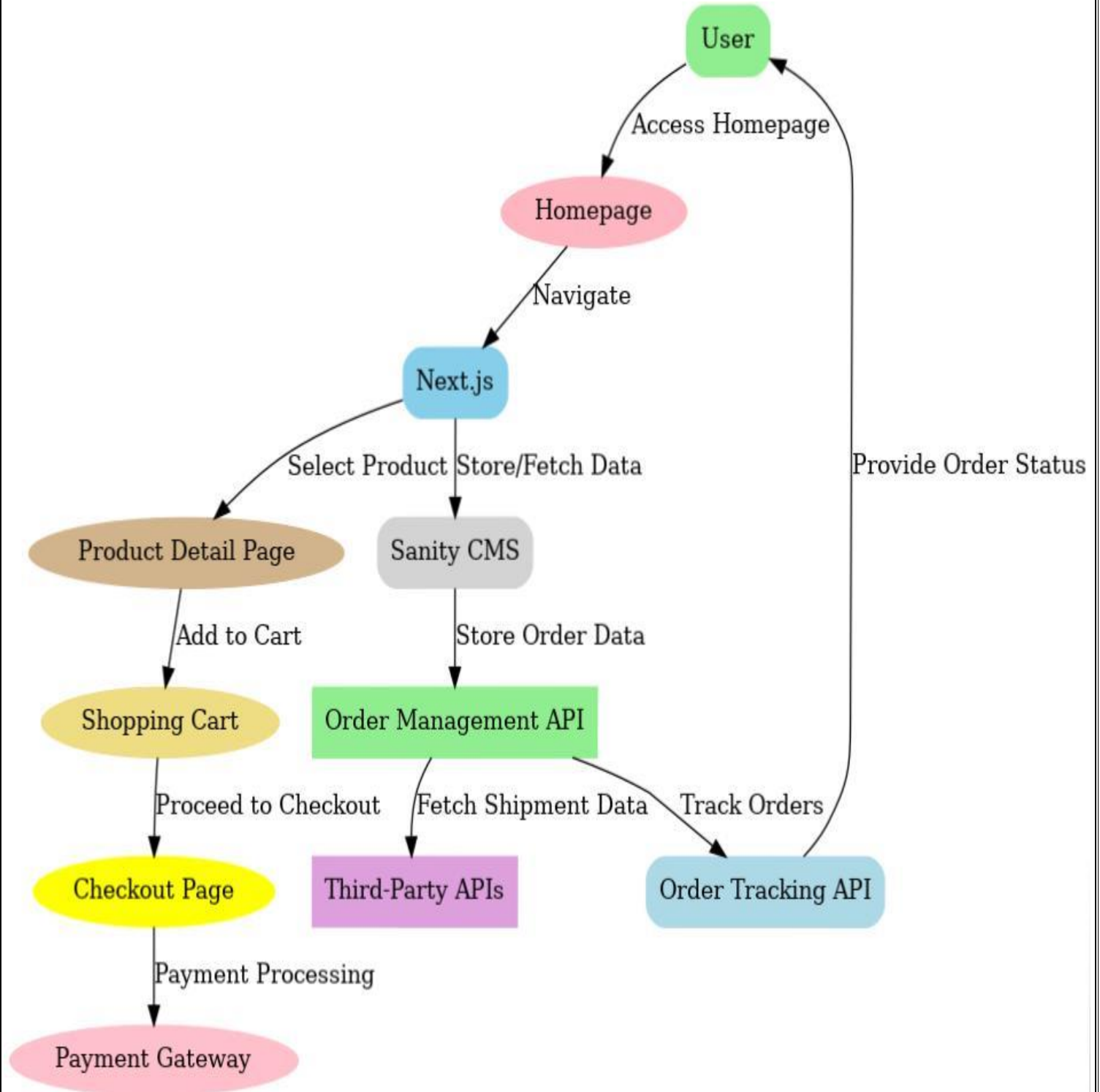
6. **Order Tracking (/order/{order_id}):**
   o Users can track their orders using the Tracking ID.
   o Order details and current status are retrieved from the Order Schema in Sanity CMS.

API Endpoints:

1. **Products (GET):** Retrieves product data from the Sanity database.
2. **Orders (POST):** Creates new orders within the system.
3. **Shipment (GET):** Fetches the current status of an order using a third-party shipping API.

Diagram:

A high-level flow diagram illustrates the routing and workflow for the marketplace, connecting the homepage, product pages, cart, checkout, and order management. The diagram outlines how data flows between the frontend, backend, and APIs, ensuring a smooth process for both users and admins.

API Endpoints with Sanity Schema Examples:

API: Products (GET)

- **Endpoint:** `/api/products`
- **Description:** Retrieve a list of products from the Sanity CMS.
- **Sanity Schema Example:**

```
export const productSchema = {
  name: 'product',
  type: 'document',
  fields: [
    { name: 'name', type: 'string', title: 'Product Name' },
    { name: 'price', type: 'number', title: 'Price' },
    { name: 'description', type: 'text', title: 'Description' },
    { name: 'stock', type: 'number', title: 'Stock' },
    { name: 'image', type: 'image', title: 'Product Image' },
  ],
};
```

API: Orders (POST)

- **Endpoint:** `/api/orders`
- **Description:** Create a new order in the system.
- **Sanity Schema Example:**

```
export const orderSchema = {
  name: 'order',
  type: 'document',
  fields: [
    { name: 'orderId', type: 'string', title: 'Order ID' },
    { name: 'products', type: 'array', of: [{ type: 'reference',
to: { type: 'product' } }], title: 'Products' },
    { name: 'total', type: 'number', title: 'Total Price' },
    { name: 'status', type: 'string', title: 'Order Status',
options: { list: ['Processing', 'Shipped', 'Delivered'] } },
    { name: 'customer', type: 'string', title: 'Customer Name'
},
  ],
};
```

API: Shipment (GET)

- **Endpoint:** /api/shipment
- **Description:** Track the shipment status of an order using a third-party API.
- **Sanity Schema Example:**

```
export const shipmentSchema = {
  name: 'shipment',
  type: 'document',
  fields: [
    { name: 'trackingId', type: 'string', title: 'Tracking ID'
},
    { name: 'orderId', type: 'reference', to: { type: 'order' },
title: 'Order ID' },
    { name: 'status', type: 'string', title: 'Shipment Status',
options: { list: ['In Transit', 'Out for Delivery', 'Delivered']
} },
    { name: 'estimatedDelivery', type: 'datetime', title:
'Estimated Delivery' },
  ],
};
```