Market Place Technical Documentation

General E – Commerce

Overview

This technical documentation provides an in-depth description of the architecture, components, and features of the E-Commerce Marketplace system. The platform is designed to deliver a seamless shopping experience for users while equipping administrators with effective management tools. Key areas such as data flow, system components, API operations, and third-party integrations are covered to offer a complete understanding of the system's functionality.

System Architecture

The architecture is designed to handle frontend user interactions, backend content management, and integration with third-party services efficiently. This ensures a user-friendly interface, secure data management, and reliable communication with external systems. Below is a detailed overview of its main components:

1. Frontend (Next.js)

- **Purpose**: Acts as the user interface, enabling customers to browse, search, and purchase products with ease.
- **Features**:
    - A visually appealing and intuitive design that enhances the shopping experience.
    - Fully responsive layout optimized for both desktop and mobile devices, ensuring accessibility across different screen sizes.
    - Dynamic rendering of product details, real-time order updates, and personalized content through API integrations with the backend.

2. Backend (Sanity CMS)

- **Purpose**: Serves as the backbone for content management, handling product catalogs, customer information, and order records.
- **Capabilities**:
    - A headless CMS designed to store, manage, and deliver content via APIs.
    - Custom data schemas created to align with the unique business requirements of the marketplace.
    - Centralized storage for product listings, user profiles, and order history, ensuring consistent and accurate data flow to the frontend.
    - Flexible data structures that allow future expansion and customization as the business grows.

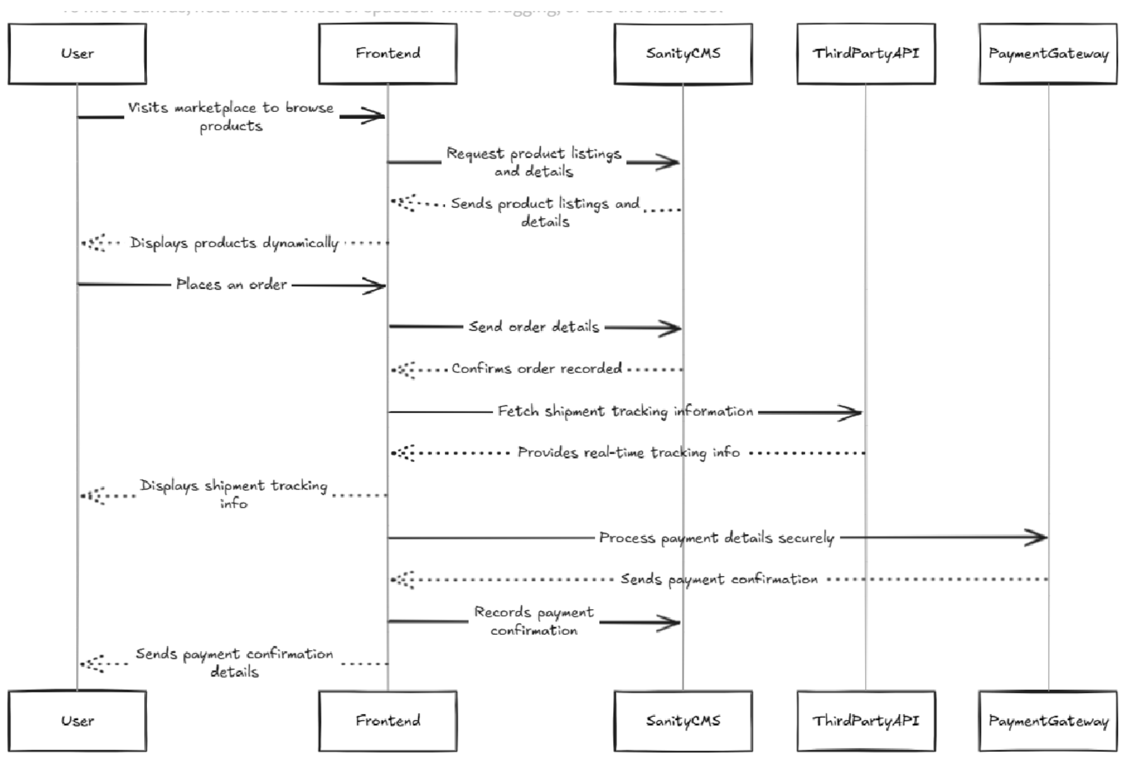
3. Third-Party Service Integrations

- **Payment Gateway**:
  - Enables secure and efficient processing of payments for orders.
  - Ensures data protection through encryption and compliance with industry standards (e.g., PCI DSS).
- **Shipping APIs**:
  - Provides real-time tracking of shipments to keep customers updated on the status of their orders.
  - Automates delivery updates and integrates with various courier services for wider coverage.
- **Analytics Tools**:
  - Monitors user behavior, sales performance, and site activity for actionable insights.
  - Supports data-driven decision-making to improve marketing strategies and user engagement.

Visual Representation

The architecture integrates these components into a unified system, ensuring smooth data flow:

1. The **Frontend** communicates with the backend to fetch product details and interact with APIs.
2. The **Backend (Sanity CMS)** stores and manages data for seamless frontend updates.
3. **Third-Party Services** handle specialized functions like payment processing, shipment tracking, and analytics reporting.


User
Frontend
SanityCMS
ThirdPartyAPI
PaymentGateway
Visits marketplace to browse products
Request product listings and details
Sends product listings and details
Displays products dynamically
Places an order
Send order details
Confirms order recorded
Fetch shipment tracking information
Provides real-time tracking info
Displays shipment tracking info
Process payment details securely
Sends payment confirmation
Records payment confirmation
Sends payment confirmation details
User
Frontend
SanityCMS
ThirdPartyAPI
PaymentGateway

Key Workflows

1. User Registration and Login

- New users can create an account, and existing users can log in to access their profiles.
- User credentials and profile information are securely stored in **Sanity CMS** for future interactions.

2. Product Exploration

- Customers can browse through product categories and view detailed information about items.
- Product data is retrieved dynamically from **Sanity CMS** through API calls to ensure up-to-date content.

3. Placing Orders

- Users can add desired products to their shopping cart and complete the checkout process.
- Details of the order, including items and user information, are recorded in **Sanity CMS** for processing and management.

4. Handling Payments

- Payments are securely processed through integrated third-party payment services.
- The system ensures safe transactions and notifies both the user and the system upon successful payment.

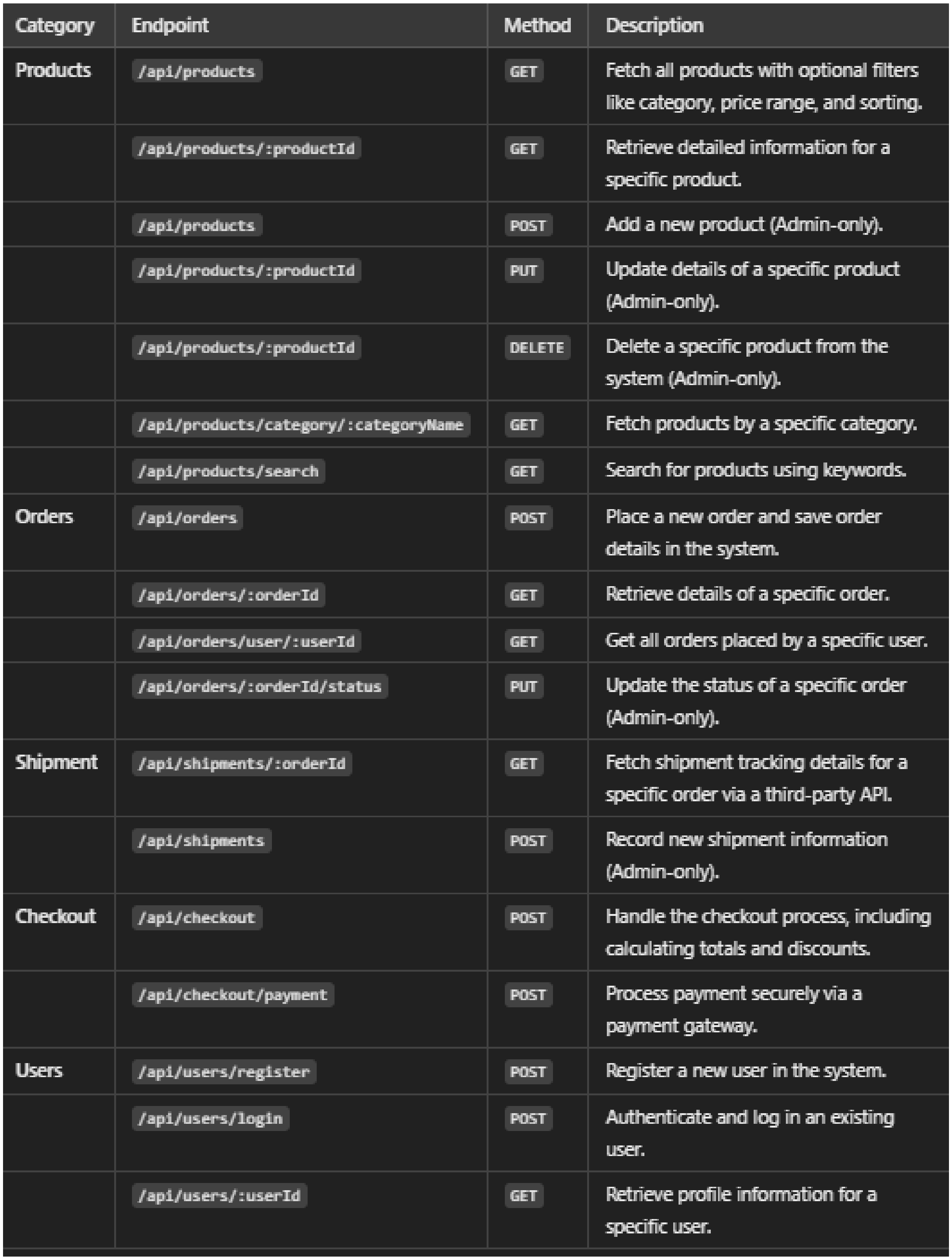
5. Tracking Shipments

- Real-time updates on shipping status are fetched from external APIs connected to courier services.
- Users can view live tracking information to monitor the delivery progress of their orders.

This flow ensures a seamless and efficient experience for both users and administrators while leveraging the strengths of Sanity CMS and third-party integrations.

API Endpoints

Table of Possible Endpoints



Category	Endpoint	Method	Description
Products	`/api/products`	`GET`	Fetch all products with optional filters like category, price range, and sorting.
	`/api/products/:productId`	`GET`	Retrieve detailed information for a specific product.
	`/api/products`	`POST`	Add a new product (Admin-only).
	`/api/products/:productId`	`PUT`	Update details of a specific product (Admin-only).
	`/api/products/:productId`	`DELETE`	Delete a specific product from the system (Admin-only).
	`/api/products/category/:categoryName`	`GET`	Fetch products by a specific category.
	`/api/products/search`	`GET`	Search for products using keywords.
Orders	`/api/orders`	`POST`	Place a new order and save order details in the system.
	`/api/orders/:orderId`	`GET`	Retrieve details of a specific order.
	`/api/orders/user/:userId`	`GET`	Get all orders placed by a specific user.
	`/api/orders/:orderId/status`	`PUT`	Update the status of a specific order (Admin-only).
Shipment	`/api/shipments/:orderId`	`GET`	Fetch shipment tracking details for a specific order via a third-party API.
	`/api/shipments`	`POST`	Record new shipment information (Admin-only).
Checkout	`/api/checkout`	`POST`	Handle the checkout process, including calculating totals and discounts.
	`/api/checkout/payment`	`POST`	Process payment securely via a payment gateway.
Users	`/api/users/register`	`POST`	Register a new user in the system.
	`/api/users/login`	`POST`	Authenticate and log in an existing user.
	`/api/users/:userId`	`GET`	Retrieve profile information for a specific user.

Sanity CMS Schema Examples Note:

The schemas below are dummy examples and are not final. You should align your schemas to match the data structure provided by the APIs you will be using to populate Sanity CMS. These examples serve as placeholders to help visualize the structure. Product Schema

Product Schema

```
export default {
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 Level' },
{ name: 'category', type: 'string', title: 'Category' },
{ name: 'image', type: 'image', title: 'Product Image' }
 ] };
```

Order Schema

```
export default
 { name: 'order',
type: 'document', fields: [
{ name: 'orderId', type: 'string', title: 'Order ID' },
 { name: 'customer', type: 'reference', to: [{ type: 'customer' }],
 title: 'Customer' },
 { name: 'items', type: 'array', of: [{ type: 'reference', to: [{ type: 'product' }] }],
title: 'Items' },
{ name: 'totalAmount', type: 'number', title: 'Total Amount' },
{ name: 'status', type: 'string', title: 'Order Status' }
 ] };
```

Conclusion

This document provides a comprehensive guide for building and deploying your furniture eCommerce platform. By following the outlined system architecture, workflows, and roadmap, you can create a robust and scalable application tailored to your business needs. Thank you for taking the time to explore this blueprint. Let’s bring your vision to life and deliver an exceptional shopping experience to your users!