TECHNICAL PLANNING

Introduction :

This document presents a comprehensive technical plan for creating an e-commerce marketplace designed to empower small businesses and individuals. The platform will enable sellers to list products and customers to shop seamlessly.

Core Technologies :

- Frontend Framework: Next.js
- Content Management: Sanity CMS
- Order Tracking: ShipEngine API
- Hosting: Vercel
- Payment Gateways: Stripe.

System Architecture :

Frontend (Next.js)

- Combines server-side and client-side rendering for a balance of optimization and fast interactivity.
- Pulls dynamic content from Sanity CMS for real-time updates.

Content Management (Sanity CMS) :

- Used to manage marketing materials, product highlights, and homepage banners with a user-friendly interface.
- Ensures integration with external services like payment gateways and shipping APIs.
- Managing product details, order information, order history, client data, and payment records efficiently.

Order Tracking :

- Real-time shipment updates are powered by ShipEngine, offering accurate delivery information to users.

Payments :

- Secure and flexible payment processing through Stripe and local payment options.

System Workflows and Components

1. User Authentication:

    - APIs manage user registration, login, and verification.

2. Dynamic Content Management:

    - Admins use Sanity CMS to manage product catalogs, banners, and marketing content.

    - The frontend fetches and displays updated content via groq queries.

3. Product Catalog and Checkout Process:

    - Server-side rendered product pages enhance user experience and SEO.

    - Sanity support browsing, filtering, and adding items to the cart.

4. Order Management and Tracking:

    - Orders are processed through backend APIs, with status updates available in real time.

    - ShipEngine integration ensures accurate delivery tracking.

5. Payments:

    - Payment gateways like jazz cash enable secure online transactions.

    - Supports multiple options, including COD, for user convenience.

API Endpoints

User Management

- Method:(POST) /api/auth/register - Register a new user.

- Method:(POST) /api/auth/login - Authenticate a user.

- Method:(GET) /api/users/profile - Retrieve user profile (authentication required).

- Method:(PUT) /api/users/update - Update user details.

Products

- Method:(GET) /api/products - Retrieve all products.

- Method:(GET) /api/products/:id - Get details of a specific product.

- Method:(POST) /api/products - Add a product (requires admin/seller access).

Orders

- Method:(POST) /api/orders - Create a new order.

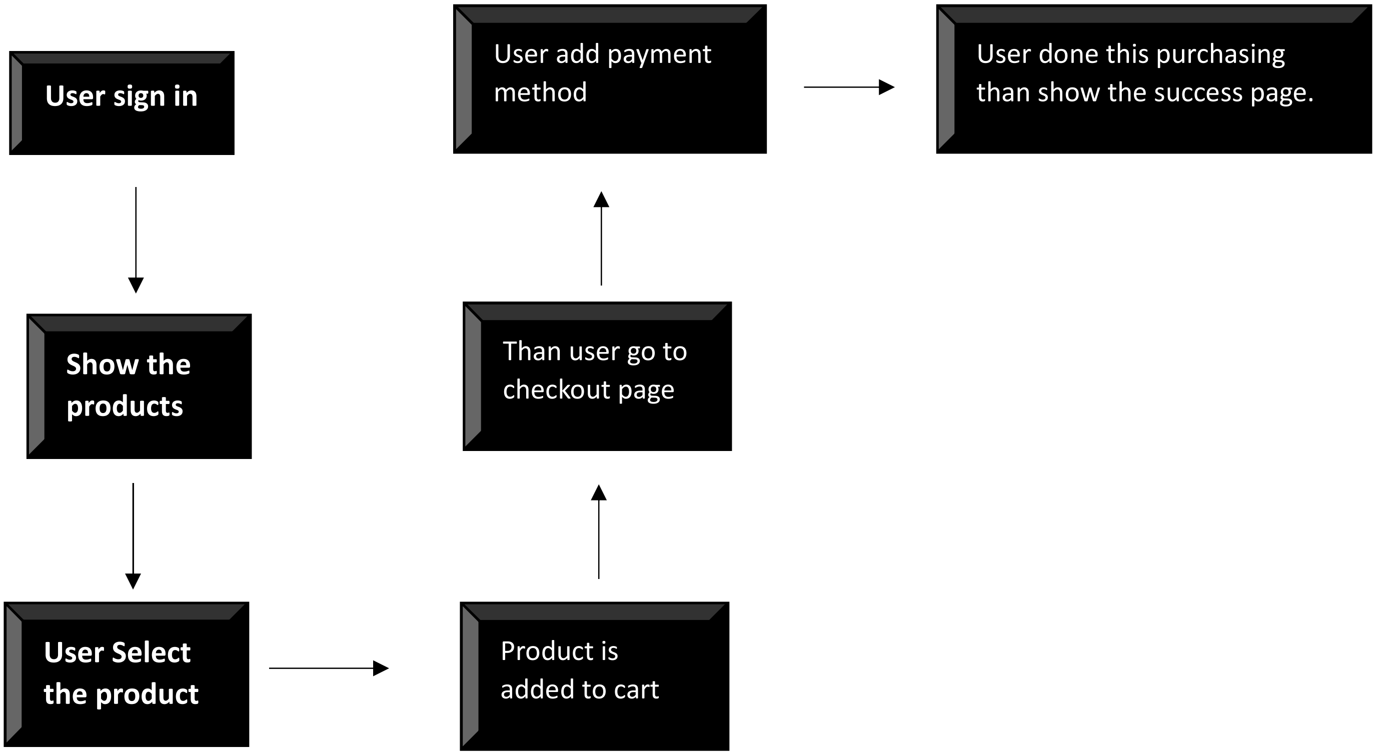
Payments

- Method:(POST) /api/payments - Process a payment. • Method:(GET) /api/payments/status - Retrieve the status of a payment.

Shipments

- Method:(POST) /api/shipments - Create a shipment. • Method:(GET) /api/shipments/track - Track shipment delivery.

User interactions with the frontend



Integration Overview

1. Sanity CMS:

a. Manages homepage banners, product highlights, and blog content.

b. groq queries provide dynamic updates to the frontend.

2. ShipEngine:

a. Handles shipping label generate and live tracking updates.

3. Stripe:

a. Supports secure payment processing, refunds, and subscriptions.

Deployment Strategy

1. Deployment:

• Hosted on Vercel with automated build

Summary

This technical plan outlines an **e-commerce marketplace** using **Next.js, Sanity CMS, Stripe, and ShipEngine** for seamless shopping, content management, payments, and order tracking. The platform features **user authentication, dynamic content updates, secure transactions, and real-time shipping tracking.** Hosted on **Vercel** for scalability and efficiency.