

Day 2!

Technical Planning Documentation

Overview

This document outlines the technical plan for developing an E-Commerce Marketplace to empower small businesses and individuals by providing a platform to sell their product online. The technical planning follows the brainstorming from Hackathon Day 1 and incorporates the recommendations from the Day 2 guidelines.

Key Technologies

- **Frontend:** Next.js
- **Content Management System (CMS):** Sanity
- **Order Tracking and Shipment:** ShipEngine
- **Database:** Clerk for authentication
- **Hosting and Deployment:** Vercel
- **Payment Gateway:** Stripe

Technical Architecture

```
[Frontend (Next.js)]
|
[Sanity CMS] -----> [Product Data API]
|
[Third-Party API] -----> [Shipment Tracking API]
|
[Payment Gateway]
```

Description:

1. Frontend (Next.js)

- Client-side rendering for speed and responsiveness.
- Server-side rendering for SEO and project page preloading.
- Integration with Sanity CMS for dynamic content.

2. Sanity (CMS)

- Manage product data, customer details, and order records.

3. Third-Party APIs

- Handles shipment tracking and payment processing.

2. Key Workflows

Product Browsing:

- User views categories or searches for products.
- Data fetched from Sanity CMS via `/products` API endpoint.

Cart and Checkout:

- User adds items to the cart.
- Proceeds to checkout, providing address and payment info.
- Order saved in Sanity CMS and processed.

Shipment Tracking:

- Order status fetched from a third-party shipment API.
- Displayed on the user's order page.

4. API Endpoints:

- Provide a table format:
 - **Endpoint:** `/products`
 - **Method:** GET
 - **Purpose:** Fetches all product details
 - **Response Example:** `{ "id": 1, "name": "Product A", "price": 100 }`

API ENPOINTS DOCUMENTATION

Authentication Endpoints

/api/auth/signup

POST_ Register new user
with clerk

/api/auth/signin

POST_ authentication user
with Clerk

/api/auth/profile

GET_ Fetch user profile
data

Products Endpoint

/api/products

GET_ Fetch products from
sanity CMS

/api/products/{id}

GET_ Get Single product
details

/api/products/search

GET_ Search products with
filter

Orders Endpoint

/api/order/create

POST_ create new order
with stripe

/api/order/history

GET_ get user's
order history

/api/order/track/{id}

GET_ Track order
status

Shipping endpoints

/api/shipping/rates

GET_ Shipping rates

/api/shipping/labels

POST_ Generate shipping
labels

/api/shipping/track

GET_ Track Shipping
status

4. Sanity Schema Example

Product Schema:

```
export default {
  name: 'product',
  type: 'document',
  fields: [
    { name: 'name', type: 'string', title: 'Product Name' },
    { name: 'price', type: 'number', title: 'Price' },
    { name: 'stock', type: 'number', title: 'Stock Level' },
    { name: 'image', type: 'image', title: 'Product Image' }
  ]
};
```

Order Schema:

```
export default {
  name: 'order',
  type: 'document',
  fields: [
    { name: 'customerName', type: 'string',
      title: 'Customer Name' },
    { name: 'products', type: 'array', title: 'Products',
      of: [{ type: 'reference', to: [{ type: 'product' }] } ] },
    { name: 'total', type: 'number', title: 'Total Amount' },
    { name: 'status', type: 'string', title: 'Order Status' }
  ]
};
```

