Planning The Technical Foundation

As you know that yesterday I have laid the foundation of my marketplace journey.
Lets move on day 2.

Technical Requirements

Frontend:

Technical Requirements

Frontend:

1. User-Friendly Interface:
   - Framework: I will use Next.js for server-side rendering (SSR) and I will follow (WCAG)(Web Content Accessibility Guidelines) to ensure inclustivity

2. Responsive Design:
   CSS Framework: I will use Tailwind CSS for rapid styling and responsive utilities. I am going to use mobile first Approach to prioritize mobile users. I will use breakpoints for responsiveness like sm, md, lg, xl, 2xl for each and every screen sizes.

promotions.

2. Product Listing Page: I am going to create Grid view of products with filters. and I will add search functionality.

3. Product Details Page: I would like to add detailed product description, images of every angle, reviews, delivery charges and "Add to cart" button.

4. Cart Page: I will display selected items, quantity adjustment and pricing breakdown.

5. Checkout Page: I will add Collect shipping, billing and payment details securely.

Order Confirmation Page:
In this page I will show order confirmation massege with thank you "greeting".

Sanity CMS As backend

Install Sanity CLI and Initialized project for backend
Firstly, I will install the Sanity CLI globally.

```
npm install -g @sanity/cli
```

then, i would create new Sanity Pooject:

```
sanity init
```

Design Schemas:
I will design schema to struct

First I will creat Product Schema
I would track product details like name, price, category and Images
Category Schema: I would categorized products for easier navigation.
Customer Schema: I would create customer Schema to manage customer details.
Order Schema: I would like to create Order Schema to track customer orders.

Connect Sanity with Next.js:
I have to install Sanity client using this command

```
npm install @sanity/client
```

I will configure the client in Next.js project

4) Query Data with GROQ:
I will use Sanity's Query language (GROQ) to fetch data for my pages

5) Preview and Publish:
I will Run the Sanity Studio locally:

sanity start

After Setting up schemas, preview them in the studio and start adding data

Third Party API

- Shipment Tracking API:
  Enable customers to track their orders in real time
  I will use Popular API which name is EasyPost
  I would install Rest API
  npm install @easypost/api
  then, I will configure API for Frontend Integration for display the tracking status dynamically on the order page detail.
- Payment Gateway API:
  I will securely handle transactions for customer payments
- Testing API:
  I would like to use tools like Postman to test API endpoints during development

Components

★ Frontend (Next.js)
- Handles user interactions and display dynamic content
- Provides a responsive and user-friendly interface

Key Actions:
- Fetch product data from Sanity CMS
- Display products, categories and order details
- Communicate with API for shipping and payments.

★ Backend (Sanity CMS)
- Manages product data, customer details and orders
- Serves as the database for all structured content

Key Action:

- Hosts Schemas for products, categories, customers and orders.
- Sends data to Frontend via API requests.

★ Third-Party API:

- Provides tracking updates for custome orders.
- Facility secure transactions.

★ Product Data API (Sanity):

- Acts as the interface for the frontend to query product and order data stored in sanity.

Step-by-Step:

* Browsing Products:
  - User visit the marketplace frontend built with Next.js.
  - The frontend sends a request to the Product Data API powered by Sanity CMS to fetch product listing.
  - The fetched data is displayed dynamically on the site.

* Product Details Page:
  - When a user clicks on a product the frontend fetches detailed information from Sanity CMS (e.g, description, image, price)

* Adding to Cart:
  - Items added to a cart are stored temporarily on the client

★ Placing An Order:

- On checkout, the order details (e.g products, quantity, customer info) are sent to Sanity CMS to be stored.
- Simultaneously the frontend interacts with the Payment Gateway API to process the transaction.
- After succesful payment, the order is marked as "Confirmed" in Sanity CMS.

★ Order Tracking:

- The order ID is used to interact with the Shipment Tracking API.
- Tracking updates are fetched and displayed on the user's order history page.

1. Products:
   - Endpoint name: /products
   - Method: GET
   - Description: Fetch all available
   - Request: No parameters needed

2. Product Details:
   - Endpoint name: /products/:id
   - Method: GET
   - Description: Fetch details of a single product by its ID
   - Request:
     URL: /products/product123

5. Orders:
Endpoint name: /orders
Method: POST
Description: Create a new order in Sanity CMS
Request: Payload and Response

6. Shipment Tracking:
Endpoint name: /shipment
Method: GET
Description: Track order shipment via third-party API
Request: Query Parameters:
?shipmentId = tracking123

General Notes for Implementation
Align with Sanity CMS API:
Use GROQ queries for fetching data from Sanity CMS.

Secure Endpoints:
Use authentication tokens or API keys to protect sensitive endpoints