Marketplace Project Progress Report

Day 1: Marketplace Business Documentation

Marketplace Type:

- General e-commerce platform catering to various product categories.

Business Goals:

- Address common pain points in e-commerce, such as lack of product interaction, insufficient details, and non-intuitive platforms.
- The solution includes clear product descriptions, high-quality images, and a user-friendly interface.

Target Audience:

- Homeowners, interior designers, office managers, tech enthusiasts, fashion-conscious consumers, and beauty aficionados.

Products & Services:

- Featured categories include:
    - Furniture
    - Clothing
    - Electronics
    - Beauty products
    - Sports & fitness

Unique Selling Points (USPs):

- Focus on ease of use, comprehensive product pages, seamless checkout, and efficient delivery.
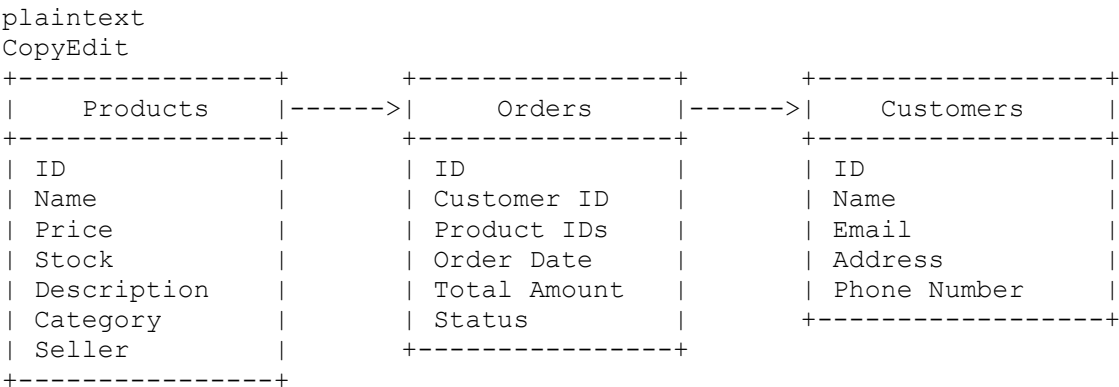
Market Research Insights:

- Competitor analysis shows that many e-commerce platforms lack clear product interaction features and struggle with slow load times.
- Consumer feedback indicates a demand for platforms with high-quality product images and intuitive navigation.
- Target audience research indicates a strong preference for fast and secure checkout processes and mobile-friendly designs.

Data Schema Diagram:

- Defined key entities (Products, Orders, Customers) with relationships between them.

Diagram:

```plaintext
CopyEdit
+---------------+        +---------------+        +-----------------+
|   Products    |------>|     Orders     |------>|    Customers    |
+---------------+        +---------------+        +-----------------+
| ID            |        | ID            |        | ID              |
| Name          |        | Customer ID   |        | Name            |
| Price         |        | Product IDs   |        | Email           |
| Stock         |        | Order Date    |        | Address         |
| Description   |        | Total Amount  |        | Phone Number    |
| Category      |        | Status        |        +-----------------+
| Seller        |        +---------------+
+---------------+
```

Additional Sections:

- **Technical Architecture:** Outlined a high-level system design and flow.
- **Security Measures:** Discussed the implementation of secure authentication and payment processing.
- **Performance Optimization:** Proposed methods for faster loading times and smooth user interactions.
- **Testing Strategy:** Identified testing methods such as unit testing, integration testing, and user acceptance testing (UAT).
- **UI Design Guidelines:** Established principles for a clean, user-friendly design.

Day 2: Marketplace Technical Documentation

Frontend Requirements:

- **User-Friendly Interface:** Key features include:
    - Search bar, filters, product pages with images, reviews, and a streamlined cart and checkout process.
- **Responsive Design:** Fluid layout across devices, with mobile-friendly touch interactions.
- **Essential Pages:**
    - Home, product listing, product details, cart, checkout, and order confirmation pages.

Backend: Sanity CMS:

- **Product Data:** Stored in Sanity CMS, including details such as name, description, price, category, and images.
- **Customer & Order Data:** Stored in Sanity CMS with references to orders and customer details.

- **Sanity Schema Definitions:** Drafted schemas for products, orders, and customers.
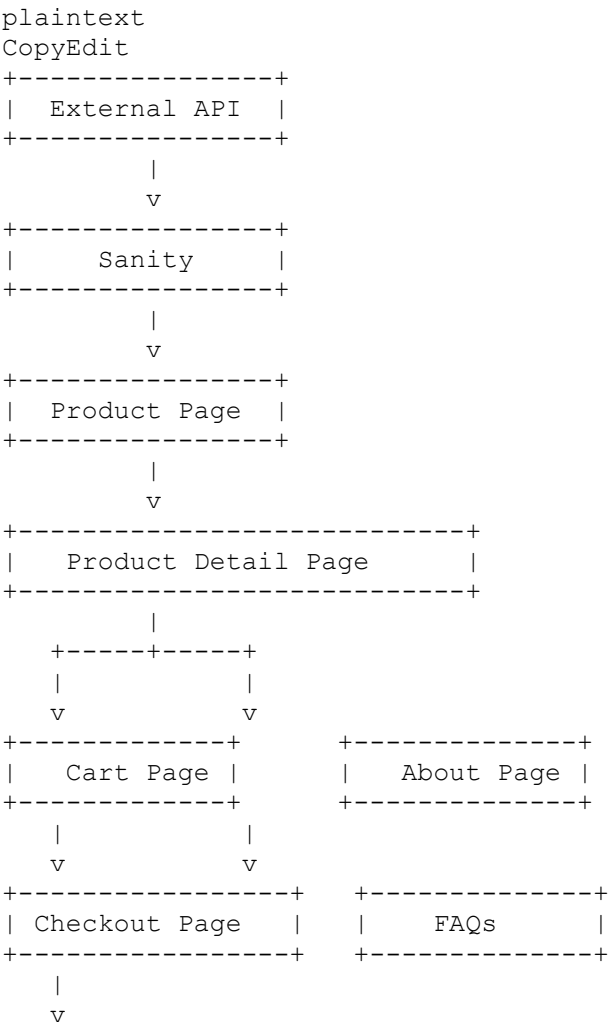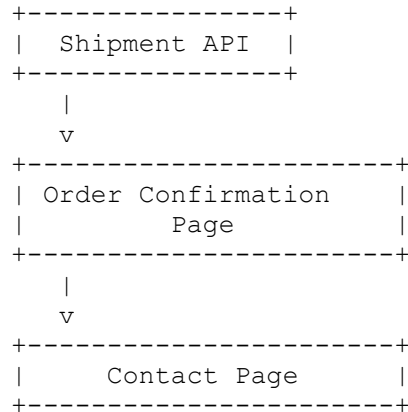
Third-Party APIs:

- **Shipment Tracking API:** Integration with ShipEngine for real-time tracking.
- **Payment Gateway API:** Stripe or PayPal for secure payments.
- **Authentication:** NextAuth.js for user management.
- **Shipping Rates API:** Real-time shipping rates integration.

System Architecture:

- **High-Level System Architecture:**
  - Decoupled frontend (Next.js), backend (Sanity CMS), and third-party services (payment and shipment tracking APIs).
- **Key Workflows:**
  - User registration, product browsing, order placement, and shipment tracking.

Workflow Block Diagram:

```
plaintext
CopyEdit
+---------------+
|  External API |
+---------------+
        |
        v
+---------------+
|    Sanity     |
+---------------+
        |
        v
+---------------+
| Product Page  |
+---------------+
        |
        v
+--------------------------+
|   Product Detail Page    |
+--------------------------+
             |
       +-----+-----+
       |           |
       v           v
+------------+      +-------------+
| Cart Page  |      | About Page  |
+------------+      +-------------+
     |                   |
     v                   v
+----------------+   +-------------+
| Checkout Page  |   |    FAQs     |
+----------------+   +-------------+
     |
     v
```

```
+---------------+
|  Shipment API |
+---------------+
    |
    v
+----------------------+
| Order Confirmation   |
|        Page          |
+----------------------+
    |
    v
+----------------------+
|     Contact Page     |
+----------------------+
```

API Requirements:

- **Key Endpoints:**
  - Defined endpoints for fetching product details, creating orders, and tracking shipments.
- **Example Request/Response:**
  - Provided example API calls for:
    - Product fetching
    - Order creation
    - Shipment tracking

Technical Documentation:

- **System Architecture Document:** Detailed explanation of the architecture, components, and integrations.
- **API Specification Document:** Thorough API documentation for endpoints, request/response formats.
- **Technical Roadmap:** A phased development plan, including setting up Sanity CMS, frontend pages, API integrations, and deployment.
- **Collaboration & Refinement:** Peer reviews and version control using GitHub.

Key Outcomes:

- **Technical Plan:** Aligned with business goals.
- **System Architecture & Workflow:** Clear diagrams explaining the system flow.
- **API & Sanity Schemas:** Drafted for product, order, and customer data management.
- **Collaboration:** Incorporating feedback from peers and mentors.

Industry Best Practices:

- Planning, using the right tools, focusing on user experience, and collaboration.

Submission Guidelines:

- Organized documentation with clear formatting.

- Submit the final version in PDF format, ensuring all diagrams and tables are labeled and explained.

System Architecture Diagram:

```plaintext
CopyEdit
[Frontend (Next.js)] <--> [Sanity CMS] <--> [Product Data API]
                              |
                              v
                    [Shipment Tracking API] <--> [Payment Gateway]
```

Next Steps:

- Continue developing the system architecture and start implementing the frontend and backend based on the specifications.
- Proceed with API integrations (payment, shipment tracking) in the upcoming phases.
- Begin testing and refine the user experience as you move towards deployment.