Marketplace Project Documentation

1. Marketplace Type: General E-commerce Platform

The project aims to create a general e-commerce platform offering a diverse range of products, such as furniture, clothing, electronics, and more. The platform will cater to customers seeking to purchase products for personal or professional use, with an intuitive shopping experience, easy navigation, and detailed product information.

2. Business Goals

Problem Statement:

Consumers often face several challenges while shopping online:

- **Inability to physically interact with products**
- **Insufficient product details**
- **Non-intuitive, complex shopping experience**

The platform aims to address these challenges by:

- Clear product descriptions, high-quality images, and an intuitive shopping experience.

Target Audience:

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

Products and Services:

- Furniture (Sofas, Tables, Office Furniture)
- Clothing (T-shirts, Jeans, Shoes)
- Electronics (Smartphones, Laptops, Gadgets)
- Beauty (Skincare, Haircare, Makeup)
- Sports & Fitness (Apparel, Equipment)

Unique Selling Points (USPs):

- User-friendly interface, comprehensive product pages, seamless checkout, and personalized recommendations.

3. Data Schema Diagram

Entities:

- **Products**: ID, Name, Price, Stock, Description, Category, Images, Seller

- **Orders**: ID, Customer ID, Product IDs, Order Date, Total Amount, Status
- **Customers**: ID, Name, Email, Address, Phone Number

Connections:

- Products → Orders
- Orders → Customers

---

4. Day 1 - Marketplace Business Documentation

Overview:

- Created a **general e-commerce platform** focused on diverse product categories.
- Designed **business goals** addressing challenges like insufficient product information and complex shopping experiences.
- Defined the **target audience** and **product categories**, with an emphasis on high-quality product details and images.

---

5. Day 2 - Technical Overview

Frontend Requirements:

- User-friendly interface with product filtering and intuitive navigation.
- Responsive design and key pages like **Home**, **Product Listing**, **Product Details**, **Cart**, and **Checkout**.

Backend with Sanity CMS:

- Product and order data management using **Sanity CMS**.
- API integration with third-party services (e.g., **Stripe**, **ShipEngine**).

System Architecture:

- **Frontend (Next.js)** communicates with **Sanity CMS** and **Product Data API**, integrating **Shipment Tracking API** and **Payment Gateway**.

API Requirements:

- Endpoints to fetch product details, create orders, and track shipments.

---

6. Day 3 - Data Integration and API Setup

Key Tasks Completed:

- **API Integration**: Product data is fetched dynamically.
- **Data Migration**: Migrated data to **Sanity CMS**.
- **Testing API Responses**: Ensured proper data mapping and error handling.
- **Data Handling**: Managed API responses for product listings and categories.

Challenges:

- **API Response Errors**: Resolved issues with misconfigured API requests and large product datasets.
- **Sanity CMS Integration**: Adjusted the schema to align with product and order data structure.

Solutions:

- **Error Handling**: Added try-catch blocks and custom error messages.
- **Performance Optimization**: Implemented pagination and lazy loading.

---

7. Day 4 - Dynamic Frontend Components

Key Components Built:

1. **Product Listing Component**: Displays dynamic product grid.
2. **Product Detail Component**: Uses dynamic routing for product details.
3. **Category Component**: Filters products by category.
4. **Search Bar**: Filters products based on name or tags.
5. **Cart Component**: Displays products added to the cart.
6. **Pagination**: Handles large product lists.
7. **Header/Footer Components**: Consistent navigation.

Current Status:

- Finalizing UI consistency, ensuring responsiveness across devices.
- Resolving minor bugs related to routing, data fetching, and pagination.

Next Steps:

- Fix mobile responsiveness, finalize bug fixes, and test user flows.

---

8. Day 5 - Testing, Error Handling, and Backend Integration Refinement

Functional Testing:

- Tested **Product Listing**, **Cart Operations**, **Search** and **User Profiles** using tools like **Postman**, **Cypress**, and **React Testing Library**.

Error Handling:

- Implemented error messages for API call failures, missing data, and server issues.

Performance Optimization:

- Reduced image sizes, implemented lazy loading, and minimized JavaScript/CSS files.

Cross-Browser & Device Testing:

- Tested on multiple browsers (Chrome, Firefox, Safari, Edge) and devices (Desktop, Tablet, Mobile).

Security Testing:

- Implemented HTTPS, input validation, and secured API keys.

---

9. Day 6 - Deployment Preparation and Staging

Objective:

- Prepare the marketplace for deployment by setting up the **staging environment** and **configuring hosting platforms** (e.g., Vercel, Netlify).

Key Learning Outcomes:

1. **Staging Environment Setup**:
    - Connected the GitHub repository to the hosting platform.
    - Configured deployment settings and environment variables securely.
2. **Staging Testing**:
    - Conducted **functional**, **performance**, and **security testing**.
3. **Professional Documentation**:
    - Created deployment reports for performance and test cases.
    - Organized the project files and maintained a clear structure.

Steps Taken:

- Deployed to **Vercel** for testing and validation.
- Verified application functionality and performance on staging.

10. Conclusion

The **Marketplace Project** has been successfully developed, tested, and deployed. The system is fully functional, and the dynamic data fetching from **Sanity CMS** ensures that the platform is scalable. Following industry best practices, the project is now prepared for production deployment.