Marketplace Development Journey

Day 1:
Defined the core concept of the marketplace, which is for "Business Groups."
The marketplace aims to provide eco-friendly and customized furniture solutions.
Target audience includes people who care about their home and environment, designers, architects, and small business owners.
A data schema was created, identifying key marketplace entities and their relationships.

Day 2:
Faced a technical problem where the npm script "importdata" was missing, causing issues with data import.

Day 3:
Integrated APIs and migrated data.
Successfully uploaded and imported various products, such as "Rustic Vase Set," "Bed," "Wood Chair," and "The Lucky Lamp."

Day 4:
Implemented data fetching and dynamic routing.
Developed a function to fetch products by slug using the Sanity API.
Created a single product page displaying product details, price, discount, and stock status.

Day 5 - Error Handling:
The team was working on fixing errors in the code, specifically in a file named page.tsx.
The error seems related to passing the slug property incorrectly when rendering the CategoryProducts component.
Debugging efforts were likely focused on ensuring proper data flow and fixing TypeScript-related issues.

Day 6 - Deployment Stage:
During the deployment process, the application failed to compile due to TypeScript linting rules.
The error message suggests that the {} (empty object) type was not correctly defined, and TypeScript required it to be either object or unknown.
There was also a React Hook warning about missing dependencies in useEffect(), which could lead to unexpected behavior.
However, the logs show successful API requests, meaning some functionalities were working

correctly, indicating progress toward a stable deployment.

Day 7 - Live Website:

The application was successfully deployed and made live. The website showcases a furniture store with product categories like Chairs, Plant Pots, Cutlery, and Tables.

A GTmetrix performance report was generated, showing:

- 89% Performance Score (good speed and optimization)
- 94% Structure Score (well-structured code and layout)
- LCP (Largest Contentful Paint) of 1.8s (indicating page loads quickly)
- TBT (Total Blocking Time) of 21ms (low delay in interactivity)
- CLS (Cumulative Layout Shift) of 0.01 (stable page layout with minimal unexpected shifts)

Conclusion:

The process involved handwriting data, terminal problem, import sanity-data, dynamic-routing debugging errors, handling deployment challenges, and optimizing performance before the website was successfully launched. The final outcome was a live website with strong performance metrics.