DAY 5 (HACKATHON)

Technical Report on Marketplace Development

Prepared by: Sufyan Mirza

Project Overview

This report explains how we built the marketplace application, the challenges faced, solutions applied, and the best practices followed during development.

Steps Taken to Build and Integrate Components

1. **Building Components:**
   - Created parts like headers, footers, product cards, and filters using React and Next.js.
   - Linked the product listing page with the Sanity CMS backend.
2. **Adding Filters and Sorting:**
   - Made a category filter so users can find products by type.
   - Added a price range filter for easier searching.

    - Created a "High to Low" price sorting option for users.
3. **Search Bar:**
    - Built a search bar for finding products by keyword.
4. **Responsive Design:**
    - Made sure all parts work well on any device using Tailwind CSS.
    - Tested on different browsers and screen sizes.
5. **Error Handling:**
    - Added simple error messages and backup displays for missing products or API issues.
6. **Testing and Optimizing:**
    - Tested all features to ensure they work properly.
    - Used Next.js tools to load images faster and make the app quicker.
    - Reduced delays by loading parts of the app only when needed.

Challenges Faced and Solutions Implemented

1. **Data Issues:**
    - **Problem:** Data from Sanity backend was incomplete at times.
    - **Solution:** Wrote better queries and handled missing data carefully.
2. **TypeScript Errors:**
    - **Problem:** Type mismatches caused issues while coding.
    - **Solution:** Defined clear data types for components and backend responses.
3. **Slow Loading:**
    - **Problem:** Pages loaded slowly due to large assets.

- **Solution:** Optimized images, used lazy-loading, and ensured efficient rendering.

4. **Responsive Design Problems:**
   - **Problem:** Layouts didn't always look right on small screens.
   - **Solution:** Fixed layouts with Tailwind CSS and tested on different devices.

Best Practices Followed

1. **Reusable Components:**
   - Designed modular parts to make future updates easy.
2. **Type Safety:**
   - Used TypeScript to avoid errors and ensure data consistency.
3. **User-Friendly Design:**
   - Focused on smooth navigation, fast loading, and clear messages for users.
4. **Testing:**
   - Tested parts separately and together to ensure everything worked.
   - Kept a record of test cases, results, and fixes.
5. **Clean Code:**
   - Followed simple coding rules to keep the project organized.
6. **Performance:**
   - Improved data loading and rendering speed for a better experience.

Final Deliverables

1. **Complete Components:**
   - All features tested and working.
2. **Error Messages:**
   - Clear warnings and backup displays added.
3. **Fast Performance:**
   - Pages load quickly with smooth interactions.
4. **Responsive Design:**
   - Works well on different devices and browsers.
5. **Testing Report:**
   - Documented test cases, results, and fixes in a CSV file.
6. **Project Documentation:**
   - Summarized the whole process, challenges, and improvements.