DAY 5 - TESTING, ERROR HANDLING, AND BACKEND INTEGRATION FOR FAME

Prepared by Shayan Siddiqui

Day 4 Recap:

Day 4 was dedicated to laying the foundation for the marketplace's dynamic functionality. Students focused on:

1. Designing and developing core front-end components such as product listings, filters, and search functionality.
2. Integrating basic API responses to display dynamic data.
3. Creating reusable and modular components for scalability.
4. Ensuring components are responsive and accessible across devices and browsers.
5. Writing technical documentation to summarize the work completed.

By the end of Day 4, I had a partially dynamic frontend with well-structured components, preparing them for the deeper integrations and testing on Day 5.

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

Objective:

Day 5 focuses on preparing your marketplace for real-world deployment by ensuring all components are thoroughly tested, optimized for performance, and ready to handle customer-facing tra ic. The emphasis will be on testing backend integrations, implementing error handling, and refining the user experience.

Key Learning Outcomes:

1. Perform comprehensive testing, including functional, non-functional, user acceptance, and security testing.
2. Implement robust error handling mechanisms with clear, user-friendly fallback messages.
3. Optimize the marketplace for speed, responsiveness, and performance metrics.
4. Ensure cross-browser compatibility and device responsiveness for a seamless user experience.
5. Develop and submit professional testing documentation that meets industry standards, including a CSV-based test report.
6. Handle API errors gracefully with fallback UI elements and logs.
7. Optimize the marketplace for speed and responsiveness.

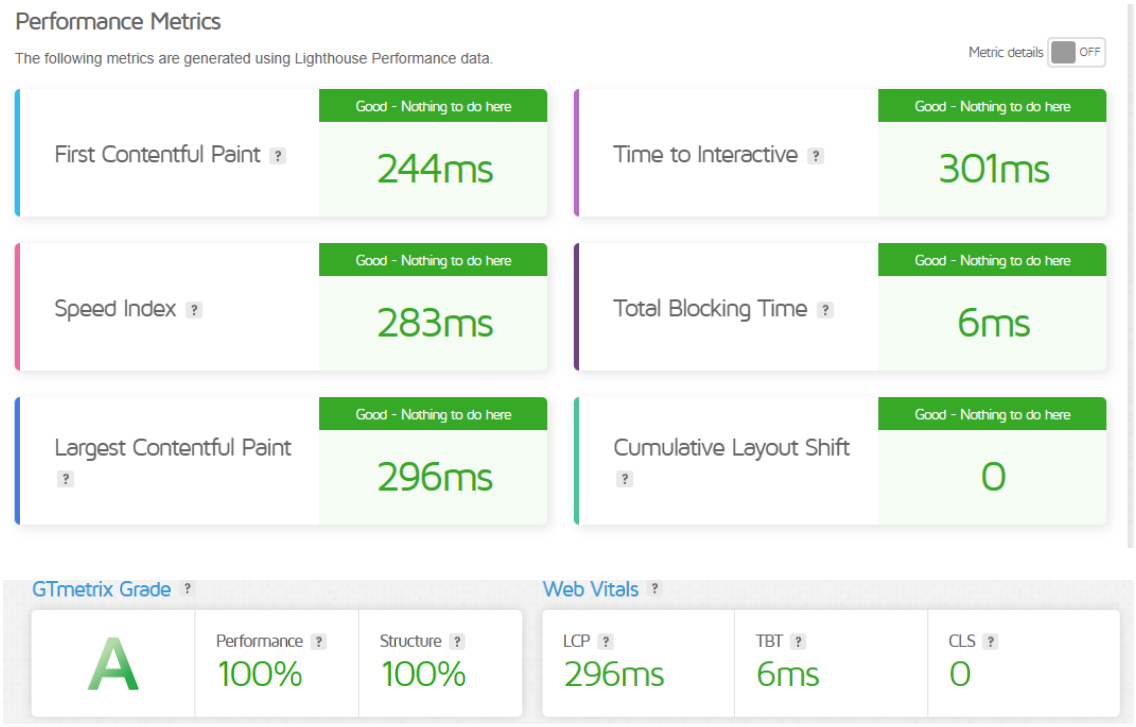
8. Ensure cross-browser and device compatibility.
9. Prepare detailed documentation for testing results and resolutions.

Key Areas of Focus:

1. Functional Testing

- Validate that all marketplace features work as intended.
- Test core functionalities like product listing, detail pages, cart operations and Checkout.

3. Performance Testing



5. Security Testing

- Validate input fields to prevent injection attacks.
- Use HTTPS for secure communication.
- Avoid exposing sensitive API keys in your frontend code.

6. User Acceptance Testing (UAT)

- Simulate real-world scenarios by interacting with your marketplace as a user.
- Verify that workflows like browsing, searching, and checkout are intuitive and error-free.