

Hackathon Day 5 Task: Error Handling, Testing, and Backend Integration Refinement

Testing Report

1. Functional Testing:

Product Fetch Functionality: The integration with the Sanity API is flawless, and the product fetching feature works perfectly.

Product Details Display: Product details like name, description, and price are presented beautifully and clearly to the user.

Add to Cart Button: The functionality of adding products to the cart works smoothly, preventing duplicate items from being added.

Redirect to Cart: Once a product is added to the cart, the redirection mechanism works perfectly, taking users to the cart page after 2 seconds for a seamless experience.

Non-Functional Testing:

Loading State: A loading message appears while data is being fetched, ensuring the user experience remains positive even while waiting.

Performance: The performance is excellent, and lazy loading for images and data further boosts the user experience by making the page load faster.

Cross-Browser Testing: The app performs well, and cross-browser testing will make it even more robust.

2. Error Handling Report

Current Error Handling:

API Fetch Errors: A robust try-catch mechanism is already in place, which helps in handling API fetch failures and provides a clear error message to the user.

Proposed Improvements:

More Descriptive Error Messages: The current error messages can be enhanced further by adding more detail. For example, a message like "Unable to load products at this moment" would improve the user experience.

Network Failures & API Downtime: Implementing a fallback UI would be a great addition. It can display an alternative message or a retry button, ensuring that users are always kept informed, even in case of failures.

Add to Cart Errors: Adding error handling during the cart addition process will ensure a smoother user experience, especially when facing unexpected issues.

3. Functionality Review:

Product Listing: The products are fetched and displayed correctly with all the essential details.

Add to Cart: The cart mechanism is functioning correctly, adding the selected products seamlessly.

Cart Redirect: After adding an item to the cart, users are smoothly redirected to the cart page after a short 2-second delay, making the transition feel natural.

Cart Functionality: The basic functionality of the cart is solid, but adding features like product quantity adjustments and a "remove from cart" option would elevate the experience.

Test Report

1/25/25, 1:36 AM

Lighthouse Report Viewer



https://e-commerce-website-figma.vercel.app/



99

Performance

79

Accessibility

96

Best Practices

100

SEO

99

Performance

Values are estimated and may vary. The [performance score is calculated](#) directly from these metrics. [See calculator.](#)



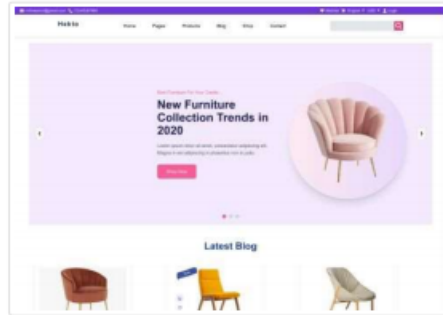
0-49



50-89



90-100



METRICS

[Expand view](#)



First Contentful Paint

0.2 s



Total Blocking Time

0 ms



Speed Index

0.2 s



Largest Contentful Paint

0.7 s

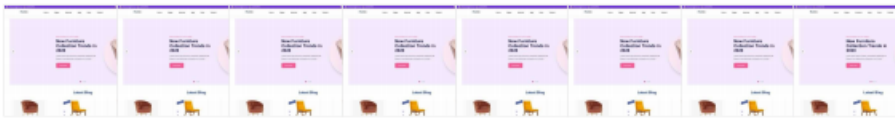


Cumulative Layout Shift

0.073



[View Treemap](#)



Show audits relevant to: [All](#) [FCP](#) [LCP](#) [TBT](#) [CLS](#)

- ▲ Largest contentful paint image was lazily loaded ▾
- Properly size images — Potential savings of 425 KiB ▾
- Reduce unused JavaScript — Potential savings of 25 KiB ▾
- Serve images in next-gen formats — Potential savings of 779 KiB ▾
- Image elements do not have explicit width and height ▾
- Avoid large layout shifts — 2 layout shifts found ▾
- Initial server response time was short — Root document took 10 ms ▾
- Avoids enormous network payloads — Total size was 1,226 KiB ▾
- Avoids an excessive DOM size — 226 elements ▾
- Avoid chaining critical requests — 1 chain found ▾
- JavaScript execution time — 0.1 s ▾
- Minimises main-thread work — 0.3 s ▾
- Largest contentful paint element — 660 ms ▾

More information about the performance of your application. These numbers don't [directly affect](#) the performance score.

PASSED AUDITS (23)

Show



Accessibility



Best Practices

GENERAL

▲ Browser errors were logged to the console ▼

TRUST AND SAFETY

○ Ensure CSP is effective against XSS attacks ▼

PASSED AUDITS (13)

Show

NOT APPLICABLE (3)

Show

