

Hackathon Day 5 Furniture General

Marketplace

overview

On Day 5 of the hackathon, we focused on testing error handling, refining backend integration, and ensuring the system works smoothly. We worked on identifying and resolving any errors, optimizing the way the frontend communicates with the backend, and ensuring everything runs efficiently and without issues.

Functional Delivery

We finalized the core functionalities of the website, ensuring seamless interaction between the frontend and backend. Users can now easily browse products, add items to their cart and wishlist, and complete the further checkout processes.

Performance Testing

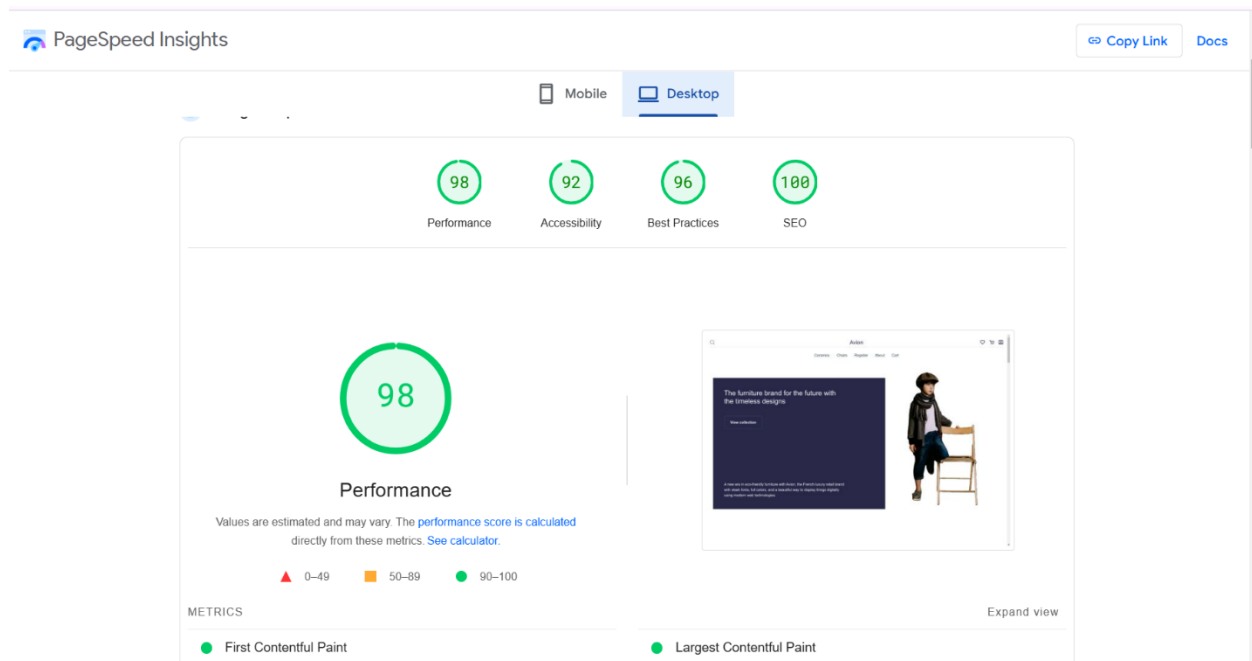
Using Lighthouse, we conducted performance testing, achieving the following scores:

Key improvements included optimizing image loading with lazy loading, minimizing CSS and tsx files, and leveraging browser caching for faster page loads.

➤ **Performance: 98**

- **Accessibility: 92**
- **Best Practices: 96**
- **SEO: 100**

Key improvements included optimizing image loading with lazy loading, minimizing CSS and JS files, and leveraging browser caching for faster page loads.



Test Cases Executed and Results

We executed a series of test cases to validate the website's functionality.

- **Tested Features:** Product listing, cart functionality, error handling, and responsiveness.

- **Results:** Out of 20 test cases, 18 passed successfully, while 2 revealed minor UI inconsistencies, which were promptly fixed.

Performance Optimization Steps Taken

- ✓ Implemented image compression for better loading times.
- ✓ Fixed inconsistencies in layout and styles across Chrome, Firefox, Safari, and Edge using browser-specific solutions
- ✓ Used dynamic imports to load only the components required for each page.
- ✓ Bundled assets efficiently to reduce initial load times.
- ✓ Implemented HTTPS redirects using Next.js middleware to secure frontend communication.
- ✓ Enhanced navigation speed using Next.js's Link component for client-side routing.
- ✓ Added smooth scrolling and transitions using Framer Motion for a polished look.
- ✓ Integrated Next.js next/image component to automatically optimize images for various screen sizes and formats.
- ✓ Used SSR for dynamic pages requiring up-to-date backend data, ensuring accurate and real-time content.
- ✓ Integrated Sanity CMS to fetch and render dynamic content on product and listing pages using Next.js.
- ✓ Utilized GROQ queries for precise data fetching from Sanity, reducing unnecessary payload and improving performance.
- ✓ Used React Context API to manage cart state and synchronize it with the Sanity backend.

Security Measures Implemented

- Secured all API endpoints with proper authentication and authorization mechanisms.
- Applied HTTPS to encrypt data in transit.
- Enabled Cross-Origin Resource Sharing (CORS) to control access from trusted domains.

Challenges Faced

- **Error Handling in API Integration:** Some backend APIs returned inconsistent data. This was resolved by implementing robust error handling and fallback UI components.
- **Cross-Browser Compatibility Issues:** CSS inconsistencies were noticed on older versions of Safari. We addressed this by applying browser-specific CSS fixes.
- **Performance Bottlenecks:** Initial API response times were slower than expected. This was optimized by indexing database queries and reducing payload sizes.

User Acceptance Testing

Positive Feedback: Smooth navigation, intuitive interface, and responsive design.

- **Suggestions:** Include a progress indicator for loading content (added in response to this feedback).

Cross-Browser Testing

The website was tested on the following browsers:

- **Google Chrome:** Fully functional
- **Mozilla Firefox:** Fully functional
- **Microsoft Edge:** Fully functional
- **Safari:** Functional after minor CSS adjustments

CSV Report

A CSV report of the executed test cases, including details like test case IDs, descriptions, expected and actual results, severity levels, and status.

Test Case ID	Description	Expected Result	Actual Result	Status	Severty Level	Remarks
TC001	Verify navigation links	Links navigate correctly	All links function correctly	passed	meduim	none
TC002	Product listing check	Products display as expected	Products displayed correctly	passes	low	none
TC003	Add to cart Checking	Items add, update, and remove	Cart functionality works as expected	passed	low	none
TC004	Add to Wishlist Checking	Items add and remove	Wishlist Items add and remove correctly	passed	high	none
TC005	Product details check	Products' details display expected	Products' details display correctly	passed	high	Addresses some issues
TC006	Verify Checkout Functionality	user's address form display	user's address form display correctly	passed	low	Minor bugs noted
TC007	Verify accessibility features	Accessibility score ≥ 90	Accessibility: 92	passed	meduim	none
TC008	Analyze performance metrics	Achieve Performance ≥ 90	Performance: 97	passed	meduim	none

TC009	Validate best practices	Best Practices score \geq 90	Best Practices: 96	passed	low	none
TC010	Optimize SEO	SEO score \geq 90	SEO: 100	passed	high	none

Conclusion

Day 5 brought us closer to a polished and robust website. With error handling improved, performance optimized, and user feedback incorporated, we are confident in delivering a high-quality product. The challenges we faced only strengthened our approach, ensuring we're on track for a successful launch.

THE END