

Objective:

Ensuring Hiply's preparation for real world deployment by testing it thoroughly, all its components, performance optimizations, and customer-traffic handling.

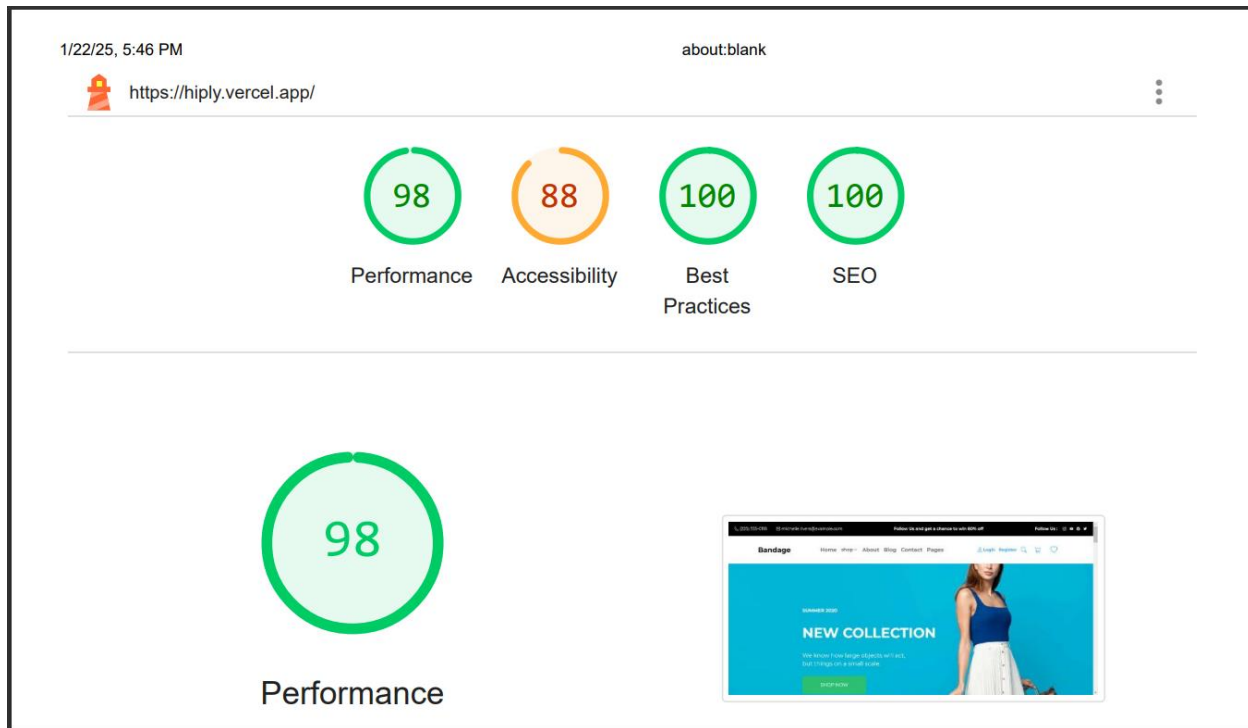
1. Functional Testing:

- a. **Product Listing:** Product listing or the "Shop" route is working perfectly fine, fetching all component data from sanity and showing it, responsive across all screens.
- b. **Product Details page:** Product details page or the dynamic product details page works completely fine, showing product title, description, available colors, sizes, stock, add to cart button, Wishlist button etc.
- c. **Cart operations:** Cart works perfectly fine, but it has this one issue or error that when adding certain quantity of products, it isn't calculating the price on the cart component, except when you click on the increment or decrement of quantity in the cart component then it calculates the whole price.

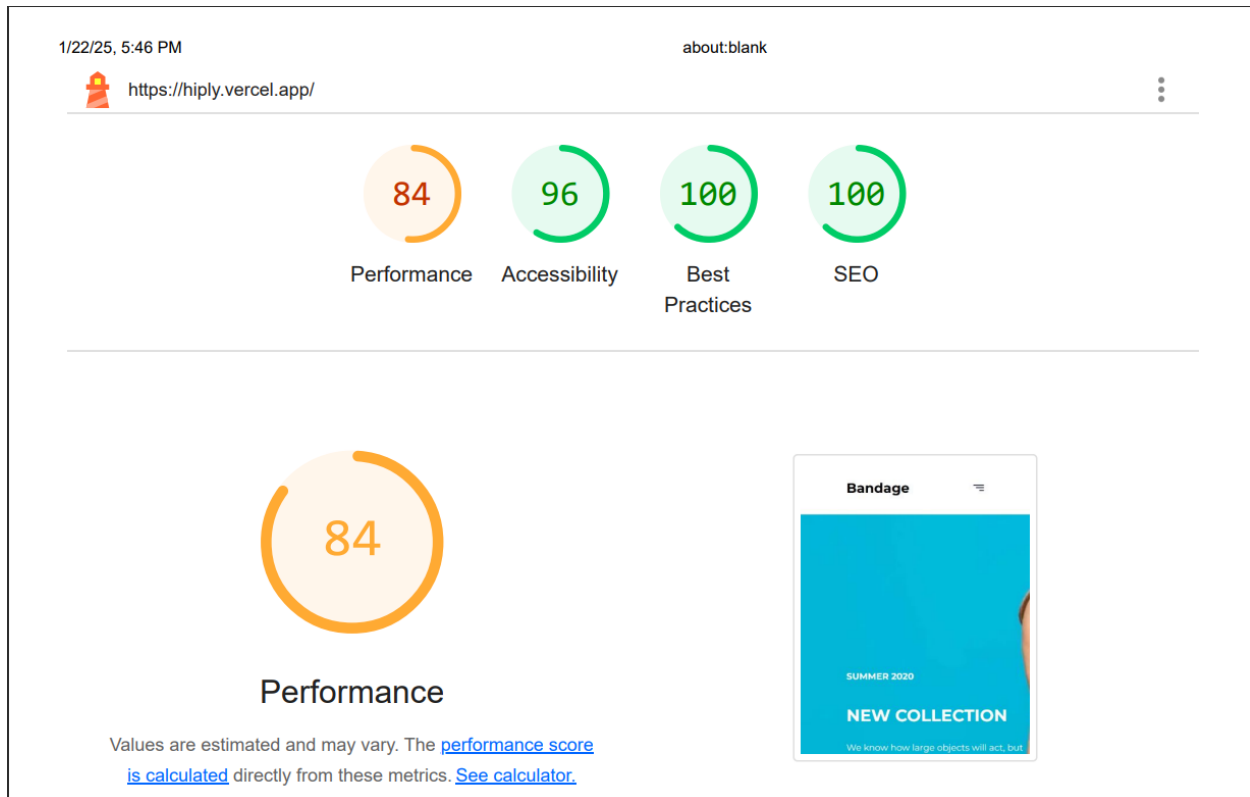
2. **Error handling:** Added respective error handling logics in the data fetching and api etc, also added fall back UIs for not found, or no products available and for all network errors.

3. Performance Testing

- a. **Light house:** On desktop test it gave the following ratings.
 - i. Performance 98.
 - ii. Accessibility 88.
 - iii. Best Practices 100.
 - iv. SEO 100.
 - v. Accessibility issues with buttons not having accessible names and background and foreground colors do not have sufficient contrast ratios.



- b. Light house:** For mobile devices it gave the following ratings.
- Performance 84
 - Accessibility 96
 - Best Practices 100
 - SEO 100
 - First contentful paint and largest contentful paints are taking more time reducing performance score.



4. Browser Testing:

- Used **Browser stack** to test different browsers like Fire Fox, Chrome, Edge and Safari, layout is consistent across every browser.
- And functionalities are properly working in all browsers.

5. Device Testing:

- Done using **Browser Stack**, the layout has responsiveness issues in the “Home” page.

6. Security Testing: Haven't done it working on it and exploring OWSAP Zap.

- Api calls are made over HTTPS, so they are secure. In addition, all the sensitive information is hold in environment variables.

7. User Acceptance Testing:

- a. Asked peers to test whole functionalities of the website. Which resulted in everything working fine except a bug in the cart functionality, mentioned above as well.

Test Case ID	Test Case Description	Test Steps	Expected Results	Actual Result	Status	Severity Level	Assigned To	Remarks
TC001	validating product listing	opened products > verify products	Product should be listed properly	Products listed perfectly	Passed	low	-	No issues
TC002	Checking Filters and Search	opened products (shops route) > tried different filters and searching products	Filters should be applied based on category, tags and price and products should be searched	Filters applied and Searches were correct	Passed	low	-	No issues
TC003	Checking Cart Operation	Click Add to cart (in product card or product details page)	product gets added in the card	products added successfully	passed	low	-	Cart not working more q... product... click on... decrem...
TC004	Validating Dynamic Route	Click product card > Verify products Details	Product details listed	Product details listed	passed	low	-	No issues
TC005	Ensuring Responsiveness across Mobile	Browser Stack and Browser dev tools window dimension adjustment	Responsive across every mobile layout	Except Home page, every page was responsive	failed	medium	-	Home p... respons...
TC006	Testing Error handled gracefully with fall back Uis	searched non existing routes, added intentional mistakes in the api calls etc	Fall back Uis being shown for errors	Fall back Uis shown	failed	low	-	Haven't... back Uis... them.
TC007	Performance	Used Light House and GTMetrix	High performance, accesibility, SEO and best practices	high performance on desktop,	passed	low	-	mobile perform... LIGHTH...

			score with loading times less than 2 seconds	but a bit low result on mobile devices				improvement in rendering content
--	--	--	--	--	--	--	--	----------------------------------