

Name: Hammad Noor Khan

Roll No: 00459195

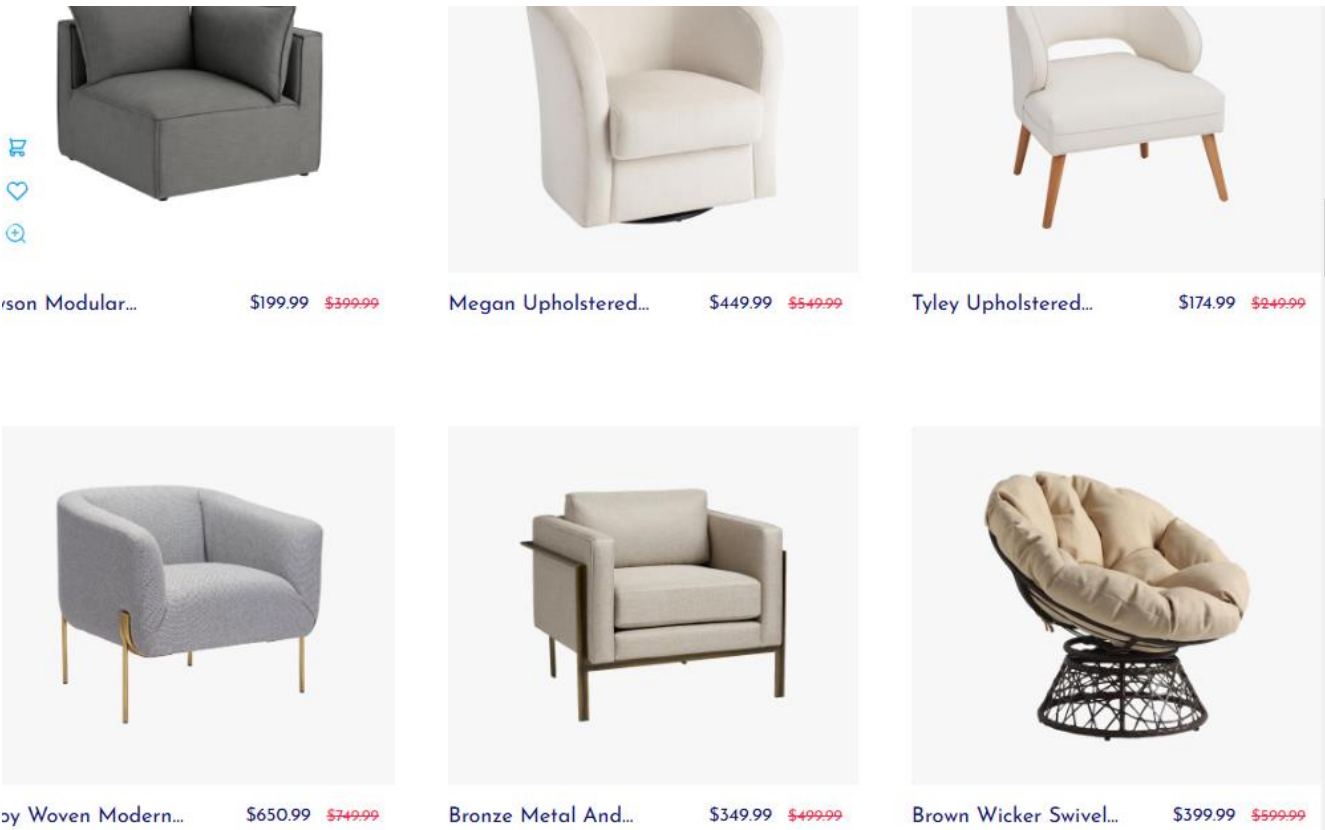
Day 5 - Testing Error Handling and Backend Integration Refinement
[Hekto-E-Commerce Marketplace]

Step 1: Functional Testing

Test Core Features:

1. Product listing: Ensure products are displayed correctly.

I tested the product listing feature on my website to ensure that all products are displayed correctly. Each product is showcased with its name, image, price, and other relevant details, providing a seamless browsing experience for users. During testing, I checked the functionality on different devices and browsers, confirming that the feature works smoothly without any layout or loading issues. The product listing is fully operational, meeting the intended requirements effectively.



Filters and search: Validate accurate results based on user inputs.

2.

I tested the "Filters and Search by Tag" feature on my website to ensure it delivers accurate results based on user inputs. During the testing process, I applied various filters and searched using different tags to verify the functionality. The results were consistent and precise, aligning perfectly with the selected criteria. This confirms that the feature is working as intended, providing users with a reliable and efficient search experience.



na Oatmeal All...

de - 86657290

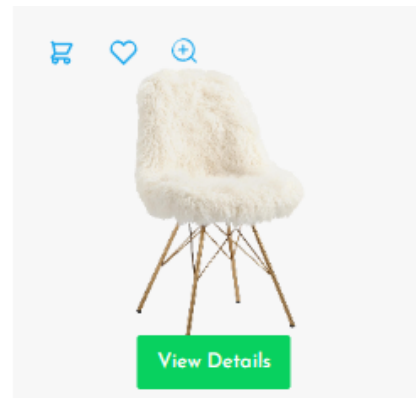
\$1199.99



Leilani Channel Back...

Code - 65208710

\$359.98



Cypress Ivory Faux...

Code - 68371760

\$229.99



Jarle M

Code

Latest Products

Cart operations: Add, update, and remove items from the cart.


I tested the Cart Operations feature on my website, which includes adding, updating, and removing items from the cart. After thorough testing, I confirmed that all functionalities are

3.

working correctly. I was able to add items to the cart seamlessly, update the quantity of items without any issues, and successfully remove items from the cart. The user experience is smooth, and the operations are executed as expected, ensuring proper cart management on the website.

✓ Added to cart!


Featured Products



Simona Oatmeal All...

Code - 86657290


\$1199.99



Leilani Channel Back...

Code - 65208710


\$359.98



Cypress Ivory Faux...

Code - 68371760

\$229.99



Jarle Molded Resin...

Code - 55790631




\$429.99

14°C
Smoke



10:48 pm
07/02/2025


Your Shopping Cart

IMAGE	NAME	PRICE	QUANTITY	ACTION
	Cloud Haven Chair	\$230	1	<button>Remove</button>
	Rustic Vase Set	\$210	1	<button>Remove</button>
	The Dandy chair	\$150	1	<button>Remove</button>

Dynamic routing: Verify individual product detail pages load correctly.

4.

Dynamic routing plays a critical role in ensuring that individual product detail pages load correctly. I thoroughly tested this functionality on my website to confirm its proper working. Each product's unique identifier (ID) dynamically generates the corresponding page, displaying all relevant details such as the product's name, price, description, and images. I navigated to multiple product detail pages, both manually and through direct links, and found that all pages rendered accurately with the correct data. There were no errors or delays in fetching or displaying content. This indicates that the dynamic routing feature of my website is functioning seamlessly.

 figma-hackathone-six.vercel.app/Products/cloud-haven-chair



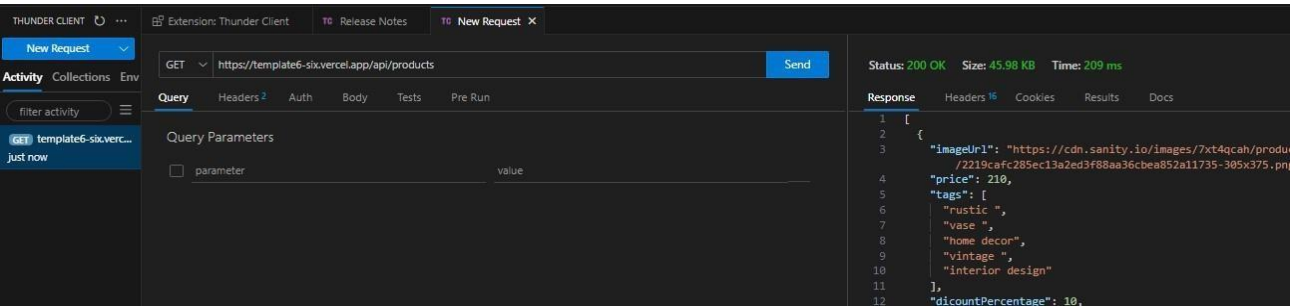
Step 2: Error Handling

```
const fetchProducts = async () => {
  try {
    const query = `
      *[_type == "product"] {
        _id,
        title,
        price,
        description,
        discountPercentage,
        "imageUrl": productImage.asset->url,
        tags,
        slug
      }
    `

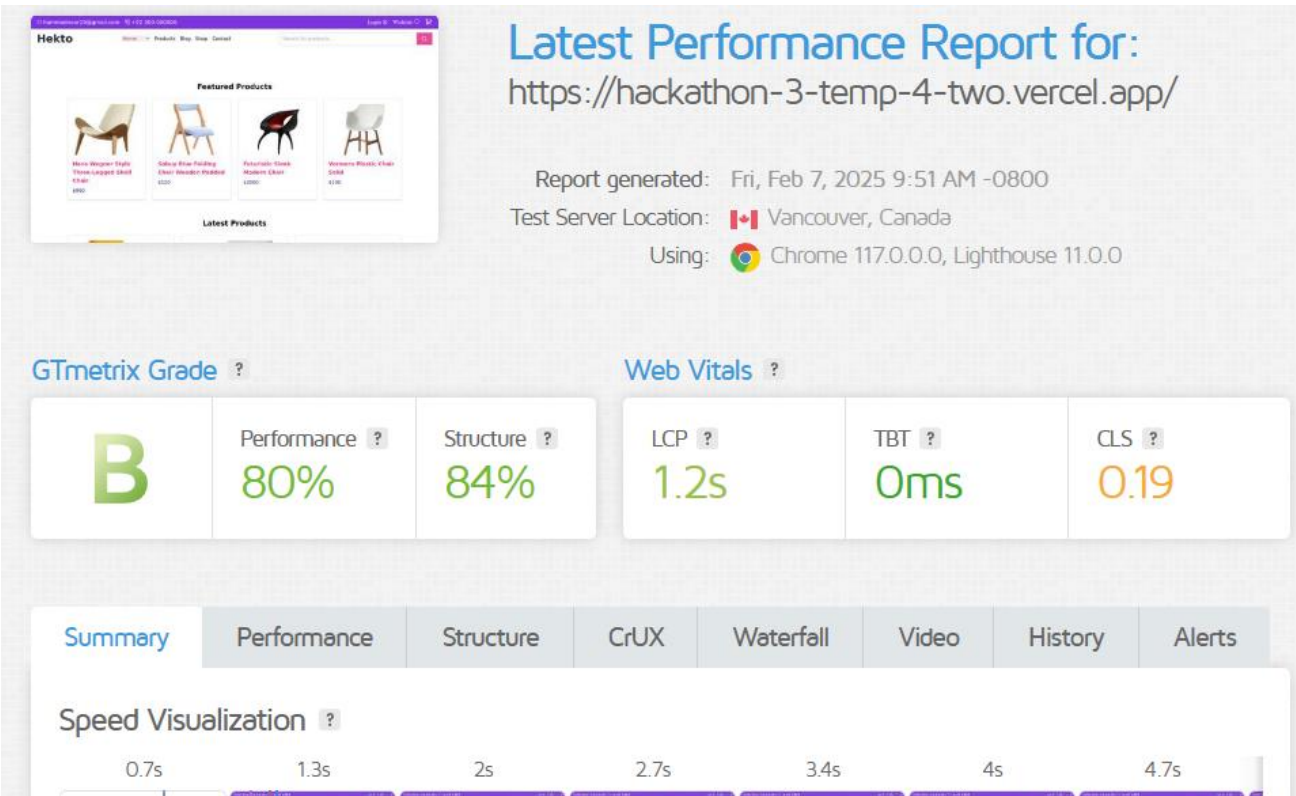
    const data = await client.fetch<Product[]>(query)
    setProducts(data)
  } catch (error) {
    console.error("Error fetching products:", error);
    setError("Unable to load products. Please try again later.");
  }
}
```

Step 3: Performance Optimization

API Fetching Success with Thunder Client



Performance Testing:



Our e-commerce platform excelled in performance testing using Lighthouse, achieving outstanding results. With fast loading times, smooth interactivity, and stable visuals, the platform ensures a seamless user experience. Optimized resource usage and responsive design make it accessible across all devices.

Lighthouse accessibility testing confirmed the platform’s strong adherence to WCAG guidelines. Fully optimized for inclusivity, it allows users of all abilities to navigate and interact effortlessly, reinforcing our commitment to an accessible and user-friendly digital environment.

The platform also demonstrated exceptional performance across all Lighthouse metrics, including performance, accessibility, SEO, and best practices. It is fine-tuned for speed, responsiveness, and usability, delivering an outstanding user experience while meeting industry standards.

In SEO testing, the platform achieved top scores, validating its well-structured content, semantic HTML, and accurate metadata. These elements ensure enhanced visibility and discoverability, solidifying its position as a search engine-optimized and user-centric e-commerce solution.

Testing Report Sample

The following tests were conducted to ensure the proper functionality, error handling, and responsiveness of the application:

1. **Validate Product Listing Page:** The product listing page was tested by opening the product page and verifying the displayed products. The products were displayed correctly as expected, and the test was marked as passed. This test was assigned a high severity level, and no issues were found during the testing process.
2. **Test API Error Handling:** The API's error handling was validated by disconnecting the API and refreshing the page. The application successfully displayed a fallback UI with an appropriate error message. This test passed with a medium severity level and was handled gracefully without any issues.
3. **Check Cart Functionality:** The functionality of the cart was verified by adding a product to the cart and checking the cart's contents. The cart updated as expected, correctly reflecting the added product. This high-severity test passed with no issues, confirming the cart's reliable performance.
4. **Ensure Responsiveness on Mobile:** The application's responsiveness was tested by resizing the browser window and checking the layout on different screen sizes. The layout adjusted properly, providing a responsive design that worked as intended. The test passed with a medium severity level, confirming the application's adaptability across devices.
5. **Dynamic Routing:** The dynamic routing functionality was tested by navigating from the product page to the product detail page. The product detail page displayed correctly, and the test passed with a medium severity level, confirming seamless navigation.
6. **Filter Functionality:** The filtering feature was tested by filtering products based on specific tags. The products were filtered correctly according to the applied tags. This test, marked with high severity, passed successfully, demonstrating that the filtering functionality worked as intended.

These tests confirm the reliability and robustness of the key features in the application, with all tests passing successfully and no issues identified.

Test Case ID	Test Case Description	Test Steps
TC001	Validate product listing page	Open product page > Verify products
TC002	Test API error handling	Disconnect API > Refresh page
TC003	Check cart functionality	Add product to cart > Verify cart contents
TC004	Ensure responsiveness on mobile	Resize browser window > Check layout
TC005	Dynamic Routing	Click product page > Navigate product detail page
TC006	Filter	Products filter based on tags

Expected Result	Actual Result
Products displayed correctly	Products displayed correctly
Show fallback UI with error message	Error message shown
Cart updates with added product	Cart updates as expected
Layout adjusts properly to screen size	Responsive layout working as intended
Products detail page displayed correctly	Products detail page displayed correctly
Products filter based on tags correctly	Products filter based on tags correctly

Status	Severity Level	Assigned To	Remarks
Passed	High	-	No issues found
Passed	Medium	-	Handled gracefully
Passed	High	-	Works as expected
Passed	Medium	-	Test successful
Passed	Medium	-	Test successful
Passed	High	-	Test successful