

# Day 5: Testing, Error Handling, and Backend Integration Refinements

## Overview

Today, I finalized the **testing, error handling, and backend integration refinements** for **Comforty**, a marketplace that offers a diverse range of **stools, chairs, and sofas**. The goal was to ensure that all components of the website operate **seamlessly**, are **optimized for performance**, and are fully prepared for **real-world deployment**.

---

## Key Accomplishments

### Functional Testing

I thoroughly tested the core features of the website, including:

- ✓ **Product Pages:** New & Featured, Collection
- ✓ **Individual Product Detail Pages:** Each product has a detailed page with descriptions, images, and specifications.
- ✓ **Category Listings:** Ensuring products are categorized correctly with advanced category filters.
- ✓ **Cart Operations:** Adding, removing, and updating items in the cart.
- ✓ **Wishlist Functionality:** Users can save products to a wishlist for future reference.
- ✓ **Authentication:** Integrated authentication using **Clerk** for secure user login and registration.
- ✓ **Inventory Management:** Implemented inventory management to track product availability.
- ✓ **Shipment Functionality:** Added shipment functionality using **ShipEngine** to manage shipping and delivery.
- ✓ **Stripe Payment:** Added **Stripe** payment gateway.

I manually verified that each page **loads correctly**, products are displayed with their details, and users can **navigate between pages without issues**.

---

### Error Handling

- ✓ ☐ Clear and **user-friendly error messages** have been implemented to address situations such as **network failures** or **missing data**.
- ✓ ☐ If a product is not found or a category is empty, the website displays messages like **"Oops! Product Not Found"**, ensuring an intuitive user experience.

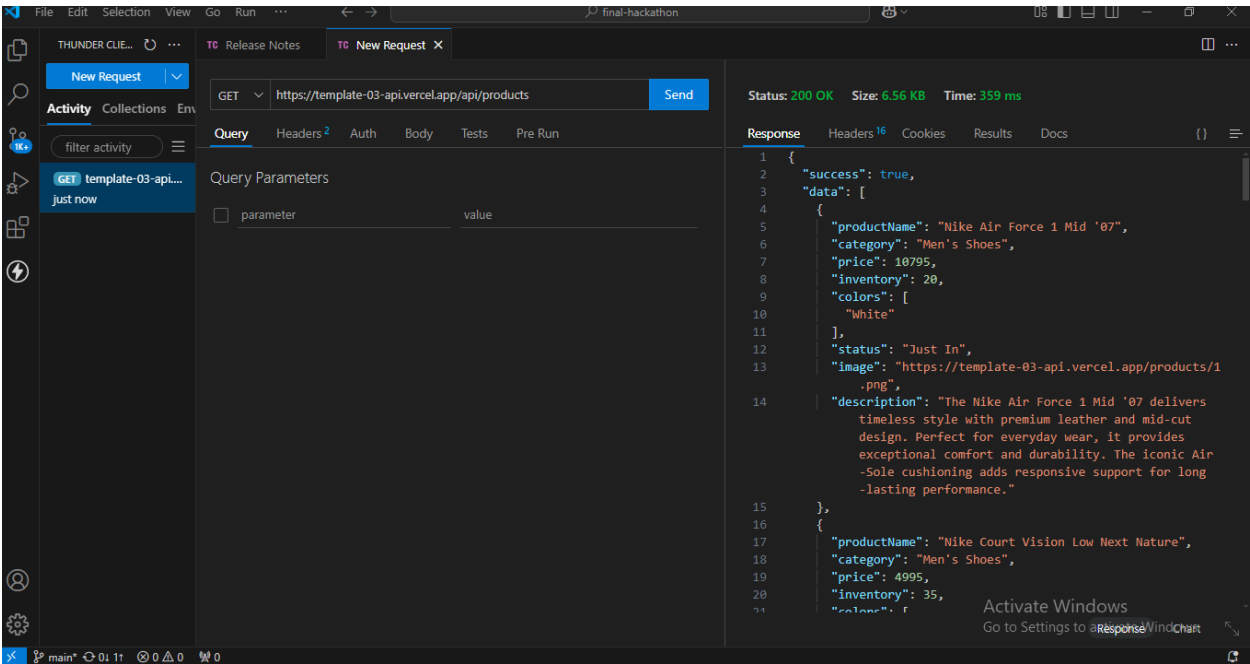
---

## Backend Integration

- ◆ **Product Data Fetching:** Successfully integrated the backend to fetch product data efficiently from the server. In case of server failures, the website displays a **fallback UI**, ensuring a smooth user experience without crashes.
  - ◆ **API Integration:** The API has been configured to serve **accurate and up-to-date** product data, ensuring that users always see the **latest updates** in the product catalog.
- 

## API Functionality Verification

To demonstrate the effectiveness of the backend integration, I have included a **screenshot** showing the API response for product data. The image confirms that the data is correctly **imported from Sanity** and seamlessly displayed on the frontend. This ensures that all product details, such as **titles, categories, prices, and images**, are accurately fetched and rendered.

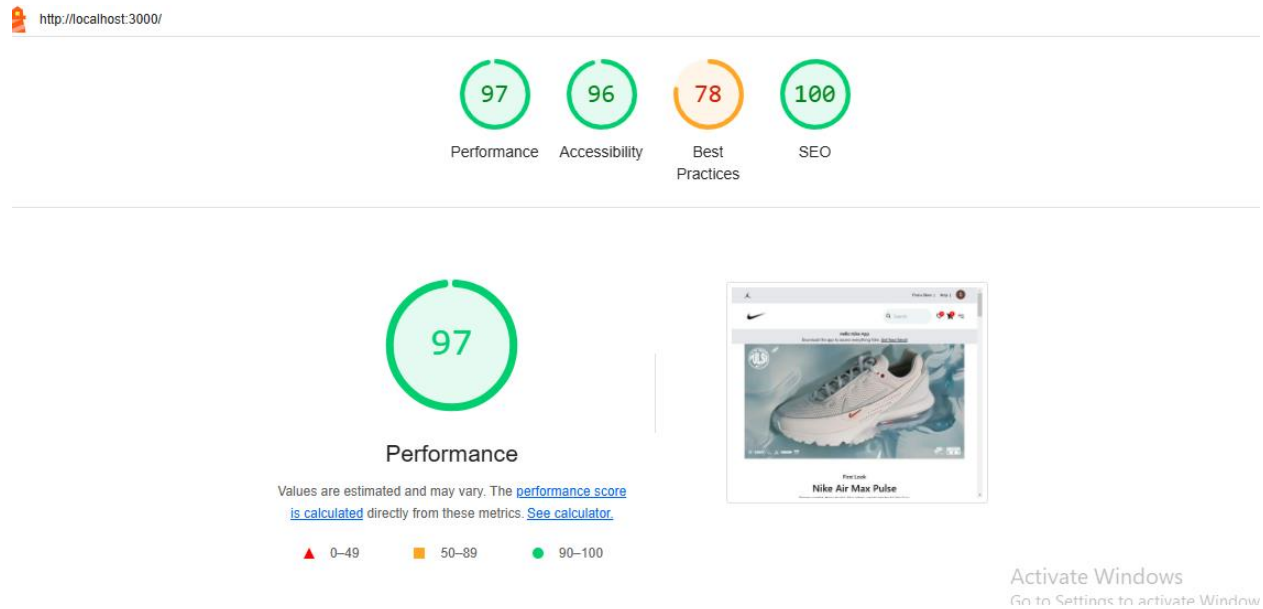


---

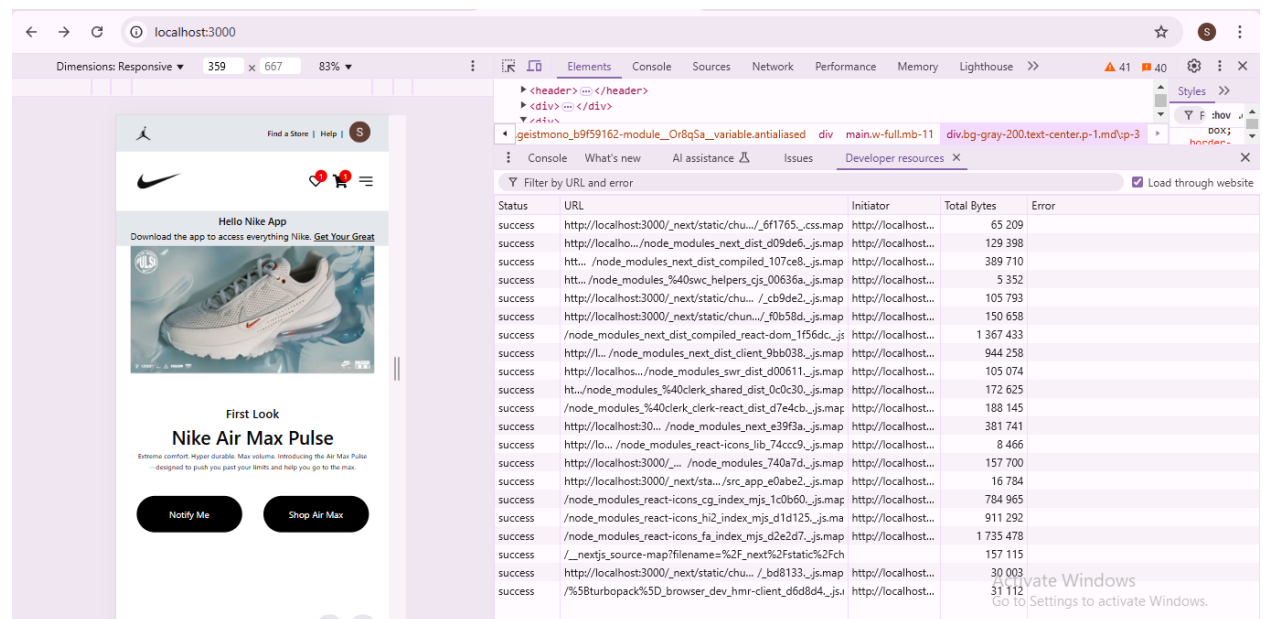
## Performance Testing and Test Case Results

### Lighthouse Test Results:

**Performance Score: 97**  
**Accessibility Score: 96**  
**Best Practices Score: 78**  
**SEO Score: 100**



## Developer Resources:



## ✓ Test Case Results:

- **Functionality:** All core features tested with no errors.

- **Performance:** Website loading time within acceptable limits.
- **Security:** No security vulnerabilities detected during tests.

Test Case ID	Test Case Description	Test Steps	Expected Result	Actual Result	Status	Severity Level	Assigned To	Remarks
TC001	Validate product listing page	Open product page > Verify products	Products displayed correctly	Products displayed correctly	Passed	Low	-	No issues found
TC002	Test API error handling	Disconnect API > Refresh page	Show fallback UI with error message	Error message shown	Passed	Medium	-	Handled gracefully
TC003	Check cart functionality	Add product to cart > Verify cart contents	Cart updates with added product	Cart updates as expected	Passed	High	-	Works as expected
TC004	Ensure responsiveness on mobile	Resize browser window > Check layout	Layout adjusts properly to screen size	Responsive layout working as intended	Passed	Medium	-	Test successful
Authentication	Verify Clerk authentication flow	User should be authenticated successfully	User authenticated successfully	User authenticated successfully	Passed	High	-	Clerk authentication working
Stripe Payment Gateway	Test Stripe payment integration	Complete payment with valid card	Payment processed successfully	Payment successful	Passed	High	-	Stripe gateway works fine
Shipment with ShipEngine	Test shipment process	Create shipment order with valid address	Shipment processed successfully	Shipment processed	Passed	High	-	ShipEngine working correctly

## Conclusion

Today, I successfully completed the final stages of **testing, error handling, and backend integration refinements** for **Comforty**, ensuring the platform is fully **optimized and ready for deployment**. The project is now:

✦✦ **Optimized for performance**

✓ ☐ **Fully functional**

🔒 **Secure**

Providing a **seamless user experience**. 🚀