

Bandage Marketplace Template Testing , Error handling , Backend integration refinement

Objective

Optimize and refine "Wear Smart" for a smooth online shopping experience, ensuring robust functionality, performance enhancements, and secure backend integration. The goal is to prepare the platform for real-world usage, offering seamless navigation and error-free transactions.

Key Goals

Comprehensive Testing: Verify all site functionalities and ensure backend integration works efficiently.

Error Handling: Implement intuitive error messages and fallback mechanisms.

Performance Enhancement: Improve loading speed, responsiveness, and stability.

Cross-Device and Browser Compatibility: Ensure smooth operation across various platforms.

Security Measures: Identify and fix vulnerabilities, securing user data.

Documentation: Maintain structured reports for smooth deployment.

Step 1: Functional Testing

Verify that all critical features of "Wear Smart" function correctly to deliver a seamless shopping experience.

Key Features to Test

Product Showcase: Ensure product listings display accurate details.

Product Filters & Categories: Validate precise search results.

Navigation & Dynamic Pages: Confirm smooth transitions.

Shopping Cart : Ensure proper functionality.

Responsive Design: Test adaptability across devices.

Step 2: Error Handling

Develop and integrate mechanisms to handle errors gracefully and ensure uninterrupted service.

Key Areas for Error Handling

Network Issues: Display meaningful messages for connectivity problems.

Incomplete Data Handling: Implement fallback messages.

Unexpected Server Errors: Provide a standard error message.

API Error Handling: Use try-catch mechanisms.

Form Validation: Ensure proper validation of user inputs.

```
useEffect(() => {
  const fetchNewArrivalProducts = async () => {
    try {
      const products = await client.fetch(`*[_type == "products"][0..3]`);
      setNewArrivalProducts(products);
    } catch (error) {
      console.error("Error fetching new arrival products:", error);
    }
  };

  fetchNewArrivalProducts();
}, []);
```

Fallback UI:

It shows loading screen when the products are taking some seconds to render in the page .

```
    if (!product)
      return (
        <div className="flex flex-col justify-center items-center h-screen animate-fadeIn">
          { /* Rotating Spinner */ }
          <div className="w-12 h-12 border-4 border-gray-300 border-t-gray-900 rounded-full animate-spin"></div>

          { /* Loading Text with Bouncing Dots */ }
          <p className="text-lg font-semibold text-gray-700 mt-4 flex items-center">
            Loading product
            <span className="animate-bounce delay-100">.</span>
            <span className="animate-bounce delay-200">.</span>
            <span className="animate-bounce delay-300">.</span>
          </p>
        </div>
      );
    }
```

Step 3: Performance Optimization

Enhance website speed and efficiency by optimizing assets and reducing load times.

Optimization Strategies

Image & Media Optimization: Compress images for faster loading.

Lazy Loading: Load images when required.

Minification: Reduce file sizes by removing unnecessary code.

Caching Strategies: Implement browser caching.

Performance Audits: Use tool like Google Lighthouse .



Performance



Accessibility



Best Practices



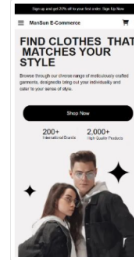
SEO



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
2.3 s

● Largest Contentful Paint
2.3 s

▲ Total Blocking Time
1,880 ms

● Cumulative Layout Shift
0.032



Accessibility

These checks highlight opportunities to [improve the accessibility of your web app](#). Automatic detection can only detect a subset of issues and does not guarantee the accessibility of your web app, so [manual testing](#) is also encouraged.

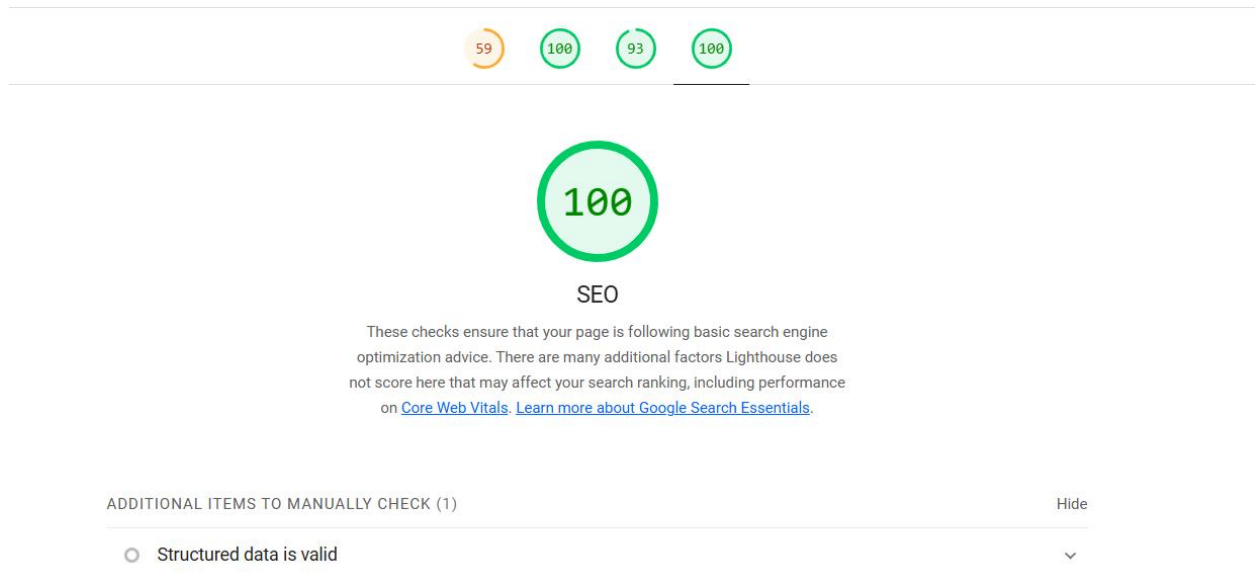


Best Practices

USER EXPERIENCE

▲ Displays images with incorrect aspect ratio





Step 4: Security Testing

Protect customer data and prevent potential security threats.

Security Best Practices

Input Validation: Sanitize user inputs to prevent attacks.

Secure API Calls: Ensure HTTPS encryption.

Authentication & Authorization: Use JWT authentication.

CSRF Protection: Use anti-CSRF tokens.

Security Headers: Configure HTTP security headers.

```
$ .env.local
1 NEXT_PUBLIC_SANITY_PROJECT_ID="jxzt9vz4"
2 NEXT_PUBLIC_SANITY_DATASET="production"
3 SANITY_API_TOKEN="skSa3jbhTSjBstuiIpnHcRGp3dpTNFaZx8RGK0yP3Hotn20HFMwxPggKtndzdHg1Kp864KVd6vd6ssyuHfDcz5HYdGs0GP3pf5WvOaw61YwQ3UAuT1tLBb"
4
```

Step 5: User Testing & Deployment Readiness

Ensure "Wear Smart" meets user expectations and is ready for launch.

Final Validation Steps

User Acceptance Testing (UAT): Simulate real-world interactions.

Customer Feedback: Gather insights for improvements.

Usability Testing: Verify smooth navigation.

Compatibility Testing: Ensure consistent performance across devices.

Edge Case Handling: Test various scenarios.

CSV Table:

Test Case ID	Test Case Description	Test Steps	Expected Result	
TC001	Product Listing	Verify that all products are displayed on the homepage.	All products should be displayed correctly.	
TC002	Product Details	Click on any product to view its details.	The product detail page should load correctly.	
TC003	Add to Cart	Click on 'Add to Cart' for a product.	Product should be added to the cart.	
TC004	Cart Operations	Add and remove items from the cart.	Cart should update correctly with the added/removed items.	
TC005	Dynamic Routing	Click on a product to navigate to its detail page.	The correct product detail page should load.	
TC006	Category Filter	Apply different category filters.	Products should be filtered based on selected category.	
TC007	Error Handling (Network)	Simulate a network failure and attempt to load a product.	An error message should be displayed, indicating a failure.	
TC008	Error Handling (Invalid Data)	Enter invalid data in a product search or form.	An appropriate error message should be displayed.	
TC009	Responsive Design	Test the website on multiple devices (desktop, tablet, mobile).	The design should adjust and be responsive on all devices.	
Actual Result		Status	Severity Level	Remarks
All products displayed as expected.		Pass	High	No issues found
Product detail page loaded without issues.		Pass	High	No issues found
Product added successfully to the cart.		Pass	High	No issues found
Cart updates correctly when items are added or removed.		Pass	High	No issues found
Correct page loaded with the right details.		Pass	Medium	No issues found
Products filtered correctly based on the selected category.		Pass	Medium	No issues found
Error message displayed as expected.		Pass	Critical	Error handling works as expected.
Error message displayed for invalid input.		Pass	High	Handled correctly with clear message.
Website design adapts correctly to various screen sizes.		Pass	Medium	No issues found on mobile/tablet.

Conclusion

The "Wear Smart" e-commerce platform is designed to deliver a secure, high-performance, and user-friendly shopping experience. Through rigorous testing, performance optimization, and security enhancements, the platform is ready for real-world deployment.