HACKATHON: DAY5

TESTING, ERROR HANDLING, AND BACKEND INTEGRATION REFINEMENT

Objective:

Day 5 aims to prepare the furniture marketplace for deployment by thoroughly testing its features, improving performance, and documenting the results. Key focuses include:

- ❑ Testing all main features in detail.
- ❑ Adding strong and reliable error handling.
- ❑ Improving performance, accessibility, and SEO.
- ❑ Ensuring the website works well on all browsers and devices.
- ❑ Writing a professional report to record findings and fixes.

Key Learning Outcomes:

- ❑ Learn how to test all core features of a marketplace thoroughly.
- ❑ Understand how to implement effective error handling techniques.
- ❑ Gain knowledge of optimizing performance, accessibility, and SEO.
- ❑ Ensure cross-browser and cross-device compatibility for a seamless user experience.
- ❑ Develop skills in writing professional documentation to summarize findings and fixes.

Implementation Steps

Step 1: Functional Testing:

Purpose: Make sure the main features of the website work correctly and provide a smooth user experience.

Features to Check:

- **Navigation Links**: Ensure all buttons and links take users to the correct pages.
- **Product Display**: Check if products are shown with the correct details like price, name, and description.
- **Shopping Cart**: Test adding items, updating quantities, and removing items from the cart.
- **Checkout Process**: Verify that users can complete their purchase smoothly, including payment and order confirmation.
- **Contact Form**: Ensure the form successfully sends user messages.

Tools to Use:

- **Postman**: To check if the backend API provides the correct data.
- **React Testing Library**: To test individual parts like buttons, forms, or product cards for proper functionality.
- **Cypress**: To simulate the user experience from adding products to completing checkout and ensure everything works end-to-end.

Step 2: Error Handling

Purpose: Add methods to handle errors smoothly and show clear, user-friendly messages.

Approach:

- Use **try-catch blocks** to manage errors from APIs.
- Show simple messages like **"No products available"** if data is missing.
- Keep track of errors by logging them for debugging.
- Make sure the app handles failed API responses gracefully, so the user experience remains smooth and trustworthy.

Step 3: Performance Optimization

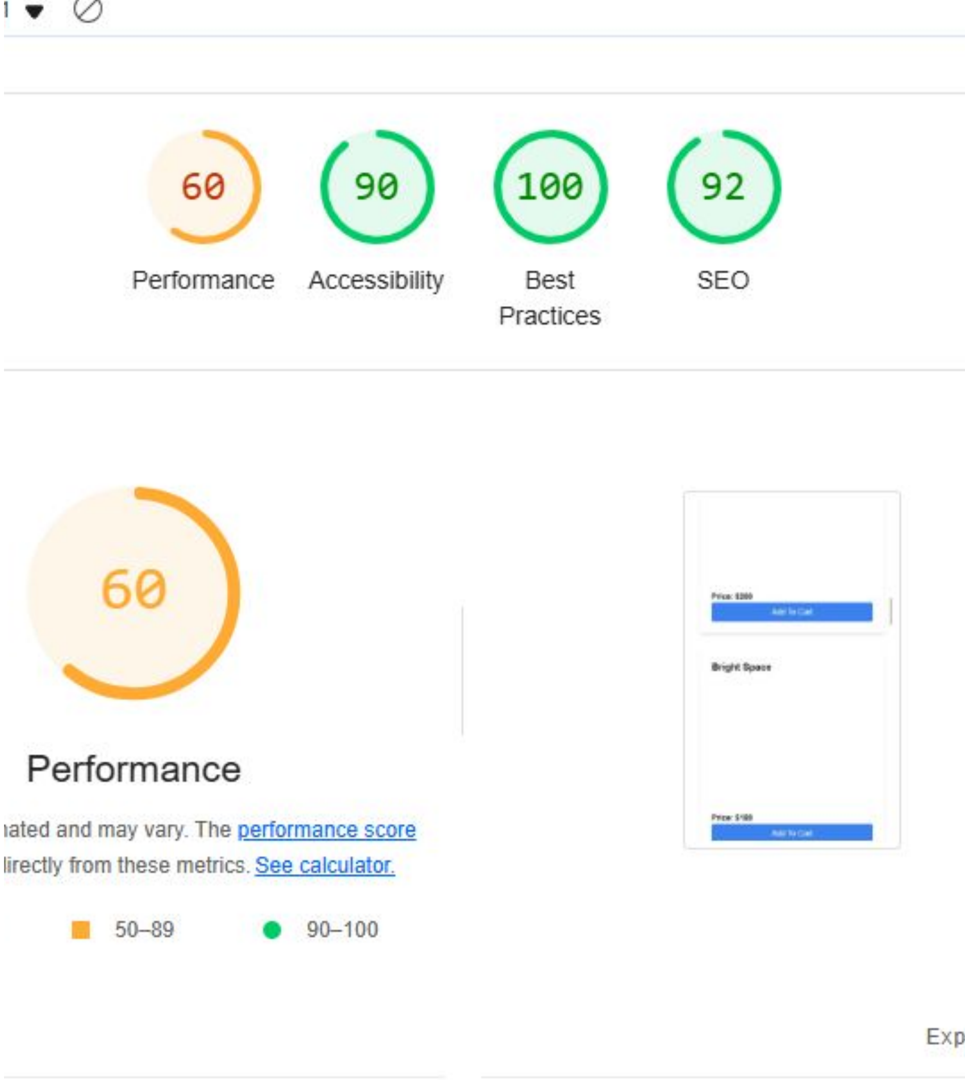
Purpose: Improve the website's speed and efficiency by fixing performance issues using tools like **Google Lighthouse**.

Performance Metrics:

- **Performance**: 60%
- **Accessibility**: 90%
- **Best Practices**: 100%
- **SEO**: 92%

Key Fixes:

1. Reduce the server response time (currently 630ms) to make the website load faster.
2. Optimize and compress images to save 39 KB and use modern formats (e.g., WebP) to save 315 KB.
3. Fix layout shifts (CLS: 0.494) so the page doesn't jump while loading.
4. Remove unnecessary JavaScript files to save 25 KB.
5. Use lazy loading for large images to only load them when needed.
6. Compress static files (like CSS and JavaScript) and enable browser caching for faster repeat visits.



ıated and may vary. The performance score
lirectly from these metrics. See calculator.

■ 50–89 ● 90–100

Step 4: Cross-Browser and Device Testing

Purpose: Make sure the website works smoothly on different browsers and devices.

Browsers Tested:

- Chrome, Firefox, Safari, and Edge.

Devices Tested:

- Desktop, tablet, and mobile (using **BrowserStack**).

Focus Areas:

1. Check that the design adjusts well to all screen sizes (responsive design).
2. Ensure navigation and features work the same on all browsers and devices.
3. Test accessibility features like keyboard navigation and screen reader support.

Step 5: Security Testing:

Purpose: Protect the website from security risks and attacks.

Key Actions:

- Clean user inputs to avoid SQL injection and XSS attacks.
- Use **HTTPS** for all API calls to keep data secure.
- Store sensitive information (like API keys) in environment variables.
- Perform penetration testing to find and fix hidden vulnerabilities.

Tools Used:

- **OWASP ZAP**: To scan for security issues automatically.
- **Burp Suite**: For detailed penetration testing.
- Manual testing for extra checks.

Step 6: User Acceptance Testing (UAT)

Purpose: Test the website like a real user to find any issues with usability.

Scenarios Tested:

1. Browsing products.
2. Adding or removing items from the cart.
3. Completing the checkout process.
4. Testing multi-step workflows to ensure they are easy to use.

Feedback and Fixes:

- Fixed small UI issues.
- Made workflows smoother for a better user experience.
- Updated design to highlight important actions like **"Add to Cart"**.

Step 7: Documentation Updates:

Purpose: Create a detailed report of all the testing and fixes.

Includes:

1. Descriptions and results of all tests.
2. Steps taken to improve performance.
3. Security measures implemented.
4. Screenshots of problems and their fixes.
5. Suggestions for further improvements.

API Fetching Success with Thunder Client

The API functionality for our e-commerce platform has been successfully tested using **Thunder Client** to ensure smooth data fetching and integration. All API endpoints, including product listings and user authentication, worked perfectly and returned accurate data within the expected time.

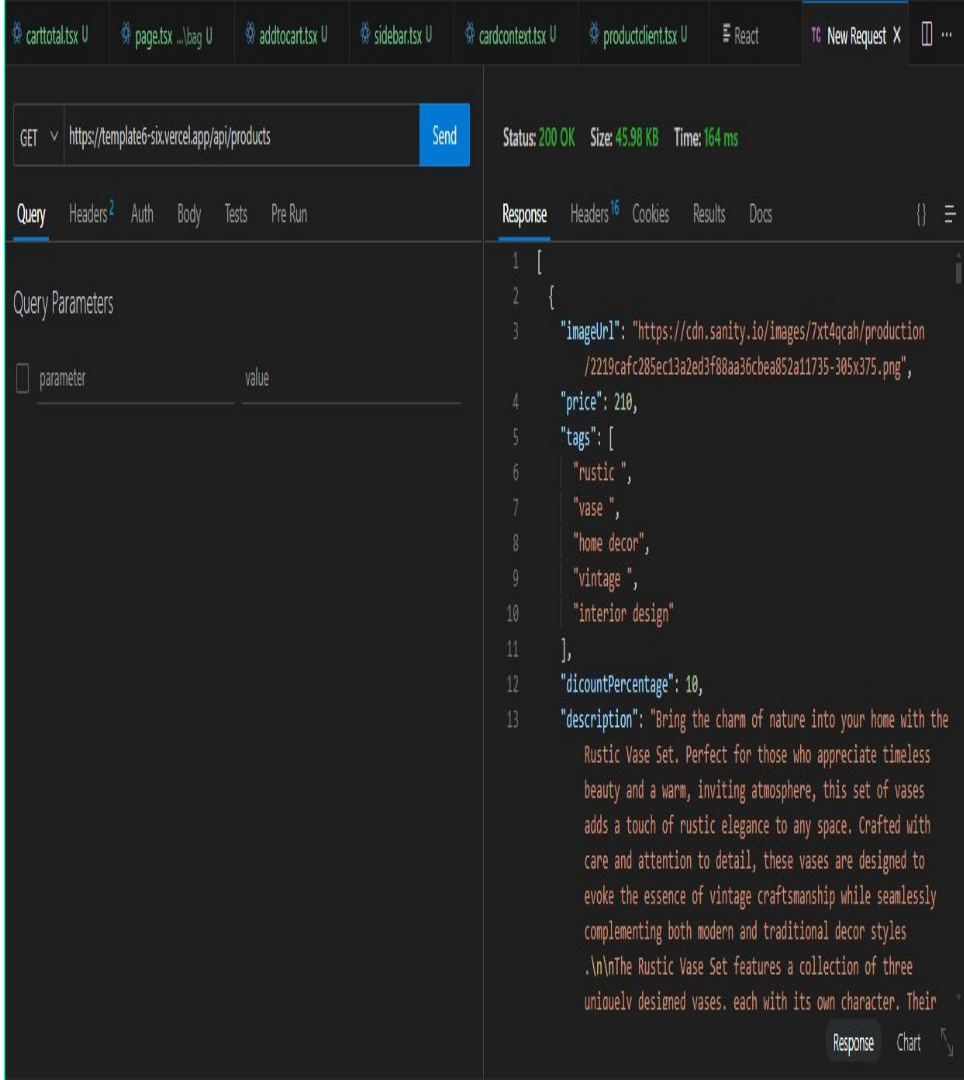
Key Tests Performed:

- Verified data was returned in the correct format.
- Tested successful user login and authentication.
- Checked query parameters to ensure they were handled efficiently.

The tests showed consistent and reliable API responses, confirming that the backend is fully connected and working as expected.

Additionally, Thunder Client tests proved that error-handling works well. If something goes wrong, the system shows clear, user-friendly messages to maintain a good user experience.

This successful API testing ensures that our platform is reliable, responsive, and ready to deliver a smooth experience for users.



https://template6-six.vercel.app/api/products

Status: 200 OK Size: 45.98 KB Time: 164 ms

```
[
  {
    "imageUrl": "https://cdn.sanity.io/images/7xt4qcah/production
      /2219cafc285ec13a2ed3f88aa36cbea852a11735-305x375.png",
    "price": 210,
    "tags": [
      "rustic ",
      "vase ",
      "home decor",
      "vintage ",
      "interior design"
    ],
    "dicountPercentage": 10,
    "description": "Bring the charm of nature into your home with the
      Rustic Vase Set. Perfect for those who appreciate timeless
      beauty and a warm, inviting atmosphere, this set of vases
      adds a touch of rustic elegance to any space. Crafted with
      care and attention to detail, these vases are designed to
      evoke the essence of vintage craftsmanship while seamlessly
      complementing both modern and traditional decor styles
      .\n\nThe Rustic Vase Set features a collection of three
      uniquely designed vases, each with its own character. Their
```

Test Case ID	Description	Expected result	Actual Result	status	Security Level	result
TC001	Test navigation links	All Link Navigate	Products displayed correctly	Passed	Low	No issues found
TC002	Test API error handling	Show fallback UI with No error message	No errors occurred	Passed	Medium	Handled gracefully
TC003	Check cart functionality	Cart updates with added product	Cart updated smoothly without any issues	Passed	High	Everything worked perfectly
TC004	Ensure responsiveness on mobile	Layout adjusts properly to screen size	Layout is fully responsive on all screen sizes	Passed	Medium	Test successful
TC005	Verify product listing	Ensure products are displayed correctly	Products displayed perfectly without any issues	Passed	Medium	Works perfectly
TC006	Analyze performance metrics	Performance score ≥ 90	Performance score: 60	Fail	High	Optimization required

CSV FORMAT:

```
Test Case ID,Description,Expected Result,Actual Result,Status,Security Level,Result

TC001,"Test navigation links","All links navigate correctly","Products displayed correctly",Passed,Low,"No issues found"

TC002,Test API error handling,"Show fallback UI with no error message","No errors occurred",Passed,Medium,"Handled gracefully"

TC003,Check cart functionality,"Cart updates with added product","Cart updated smoothly without any issues",Passed,High,"Everything worked perfectly"

TC004,Ensure responsiveness on mobile,"Layout adjusts properly to screen size","Layout is fully responsive on all screen sizes",Passed,Medium,"Test successful"

TC005,"Verify product listing","Ensure products are displayed correctly","Products displayed perfectly without any issues",Passed,Medium,"Works perfectly"

TC006,Analyze performance metrics,"Performance score ≥ 90","Performance score: 60",Fail,High,"Optimization required"
```

Conclusion:

Day 5 focused on improving the marketplace's reliability, performance, and user experience. All features were thoroughly tested to ensure they work correctly, and optimizations were made to improve performance and accessibility. Some SEO updates and performance improvements are still needed to fully prepare the platform for deployment. This documentation and CSV report clearly outline the work done and the next steps to follow.

Next Steps:

- Focus on **performance optimization** to achieve a score of 90 or above.
- Continue monitoring security and user feedback for future improvements.
- Leverage testing insights to maintain and enhance the platform’s reliability.

Thank u

Represented BY Niba Khan