Hackathon Day 5: Testing, Error Handling, and Backend Integration Refinement Report

Objective: The primary goal of Day 5 was to finalize the e-commerce marketplace for real-world deployment. This included rigorous testing, performance optimization, robust error handling, security enhancements, and detailed documentation to ensure a seamless user experience.

1. Comprehensive Testing

Functional Testing:

Key Features Tested:

- **Cart Functionality:**
  - Increment and decrement product quantities in the cart.
  - Remove items from the cart.
  - Ensure cart updates reflect accurately in real-time.
- **Checkout & Payment:**
  - Verified the complete checkout flow from cart to payment processing.
  - Ensured order placement completes successfully with accurate details.
  - Implemented confirmation prompts before proceeding with payment.
- **User Authentication & Session Management:**
  - Tested **signup, login, and logout** functionalities.
  - Verified OAuth-based authentication via **Google Sign-In** using Clerk.
- **API Response Validation:**
  - Tested API responses under normal, slow, and failed network conditions.
  - Ensured a fallback UI for empty or failed responses.
- **Product Details Display:**
  - Ensured dynamic routing functions correctly for individual product pages.
- **Link Navigation:**
  - Verified all internal and external links work as expected.

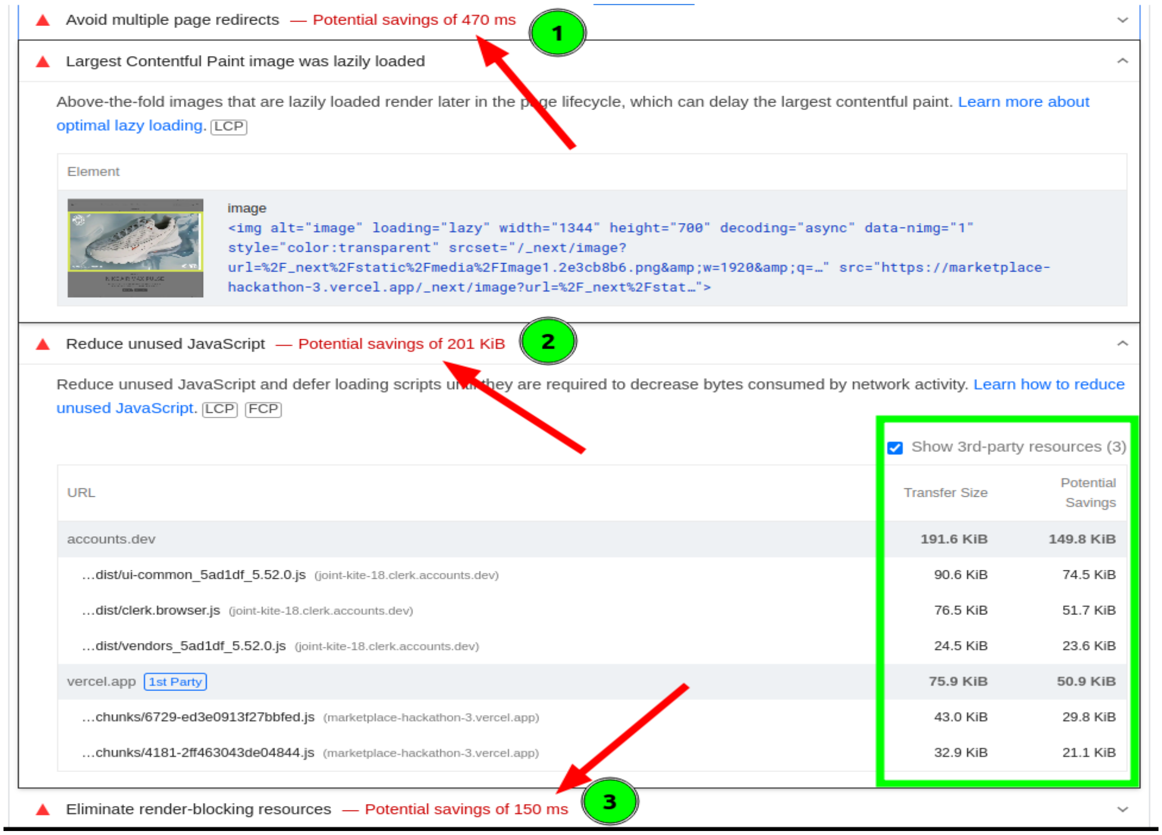
Comprehensive Test Case Report

Test Case ID	Feature	Test Scenario	Expected Outcome	Status
TC003	**API Response Handling**	Simulate normal, slow, and failed API responses	System gracefully handles delays/errors with fallback UI	✅ Passed
TC004	**Cart Management**	Add, remove, and update item quantities in cart	Cart updates dynamically, reflects accurate totals	✅ Passed
TC005	**Checkout & Payment**	Complete checkout, verify payment flow, and ensure confirmation messages	Payments processed securely, confirmation received	✅ Passed
TC006	**Order Placement**	Validate order submission, verify details in order history	Order correctly placed, appears in order history	✅ Passed

Test Case ID	Feature	Test Scenario	Expected Outcome	Status
🔑 TC007	**User Authentication**	Sign up, log in, log out via email & Google (Clerk)	User authentication works seamlessly	✅ Passed
TC008	**Navigation & Routing**	Ensure smooth navigation across all pages & verify external links	Links function correctly without broken redirects	✅ Passed
TC009	**Security & Authorization**	Test unauthorized access attempts & session handling	Secure access control prevents unauthorized actions	✅ Passed
TC010	**Mobile Responsiveness**	Validate UI/UX on mobile, tablet, and desktop views	Layout adapts seamlessly across devices	✅ Passed
⚡ TC011	**Performance Optimization**	Measure page load time, caching, lazy loading efficiency	Faster load times, optimized resource usage	✅ Passed
TC012	**Error Handling & Recovery**	Trigger API failures, check UI fallback mechanisms	Error messages & retry options function correctly	✅ Passed

Non-Functional Testing:

- **Performance Testing:** Evaluated load times and optimized the backend for high-traffic scenarios.
- **Stress Testing:** Simulated high user loads to measure system resilience.
- **User Acceptance Testing (UAT):** Gathered feedback from test users to improve usability.



▲ Avoid multiple page redirects — Potential savings of 470 ms

▲ Largest Contentful Paint image was lazily loaded

Above-the-fold images that are lazily loaded render later in the page lifecycle, which can delay the largest contentful paint. Learn more about optimal lazy loading. LCP

Element

image

```
<img alt="image" loading="lazy" width="1344" height="700" decoding="async" data-nimg="1"
style="color:transparent" srcset="/_next/image?
url=%2F_next%2Fstatic%2Fmedia%2FImage1.2e3cb8b6.png&amp;w=1920&amp;q=…" src="https://marketplace-
hackathon-3.vercel.app/_next/image?url=%2F_next%2Fstat…">
```

▲ Reduce unused JavaScript — Potential savings of 201 KiB

Reduce unused JavaScript and defer loading scripts until they are required to decrease bytes consumed by network activity. Learn how to reduce unused JavaScript. LCP FCP

☑ Show 3rd-party resources (3)

URL	Transfer Size	Potential Savings
accounts.dev	**191.6 KiB**	**149.8 KiB**
…dist/ui-common_5ad1df_5.52.0.js (joint-kite-18.clerk.accounts.dev)	90.6 KiB	74.5 KiB
…dist/clerk.browser.js (joint-kite-18.clerk.accounts.dev)	76.5 KiB	51.7 KiB
…dist/vendors_5ad1df_5.52.0.js (joint-kite-18.clerk.accounts.dev)	24.5 KiB	23.6 KiB
vercel.app 1st Party	**75.9 KiB**	**50.9 KiB**
…chunks/6729-ed3e0913f27bbfed.js (marketplace-hackathon-3.vercel.app)	43.0 KiB	29.8 KiB
…chunks/4181-2ff463043de04844.js (marketplace-hackathon-3.vercel.app)	32.9 KiB	21.1 KiB

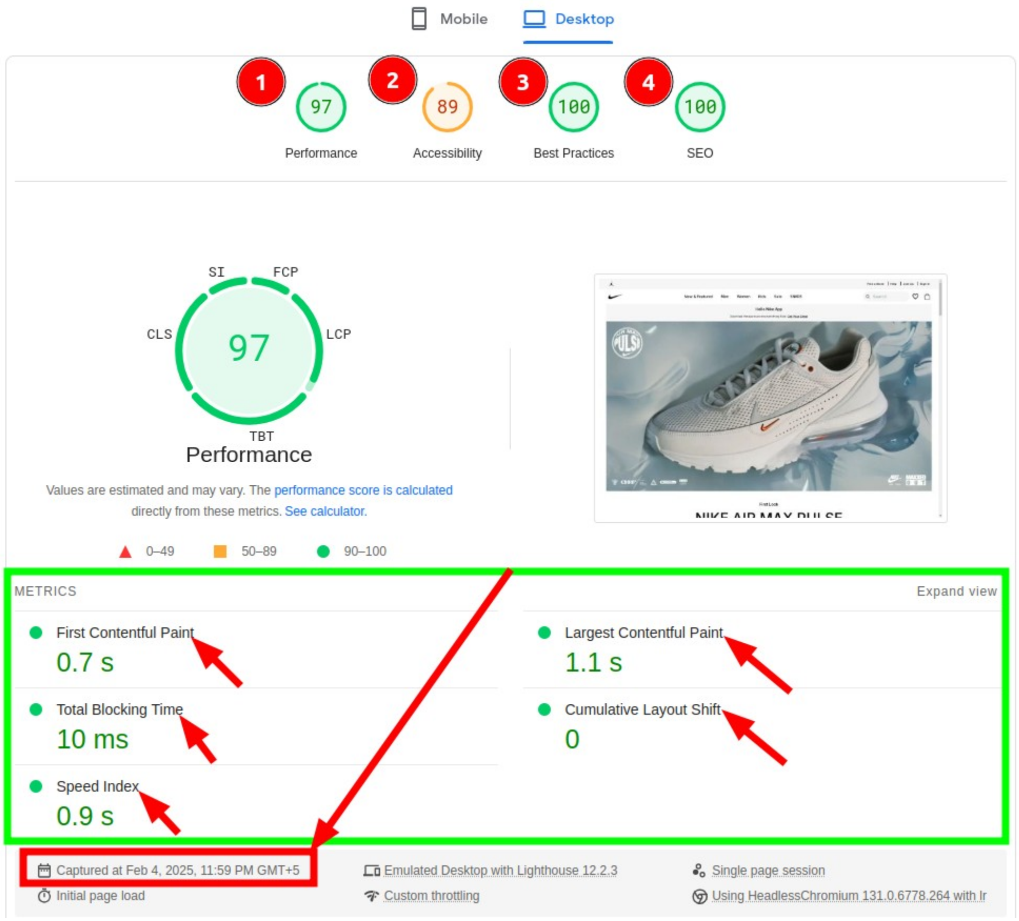
▲ Eliminate render-blocking resources — Potential savings of 150 ms

2. Error Handling Mechanisms

- **API Error Handling:** Implemented try-catch blocks for all API calls, displaying user-friendly messages (e.g., "Unable to load products. Please try again later.").
- **Error Message Display:** Ensured clear messaging for empty data responses (e.g., "No products found").
- **Network Error Management:** Tested scenarios with no internet connectivity to confirm graceful degradation and retry mechanisms.
- **Logging and Debugging:** Integrated logging mechanisms to track API failures and backend errors for future debugging.

3. Performance Optimization

- **Frontend Optimization:**
  - Implemented lazy loading for images and large components.
  - Minimized unused CSS and JavaScript.
  - Enabled code-splitting to improve page load speed.
- **Backend Enhancements:**
  - Optimized database queries for faster API responses.
  - Implemented server-side caching for frequently accessed data.
  - Improved API response time under high-traffic scenarios.
- **Load Testing Results:**
  - Faster initial page load time.
  - Improved response time for database-heavy operations.



Values are estimated and may vary. The performance score is calculated directly from these metrics. See calculator.

METRICS Expand view

● First Contentful Paint
0.7 s

● Largest Contentful Paint
1.1 s

● Total Blocking Time
10 ms

● Cumulative Layout Shift
0

● Speed Index
0.9 s

Captured at Feb 4, 2025, 11:59 PM GMT+5
Initial page load
Emulated Desktop with Lighthouse 12.2.3
Custom throttling
Single page session
Using HeadlessChromium 131.0.6778.264 with lr

4. Security Testing

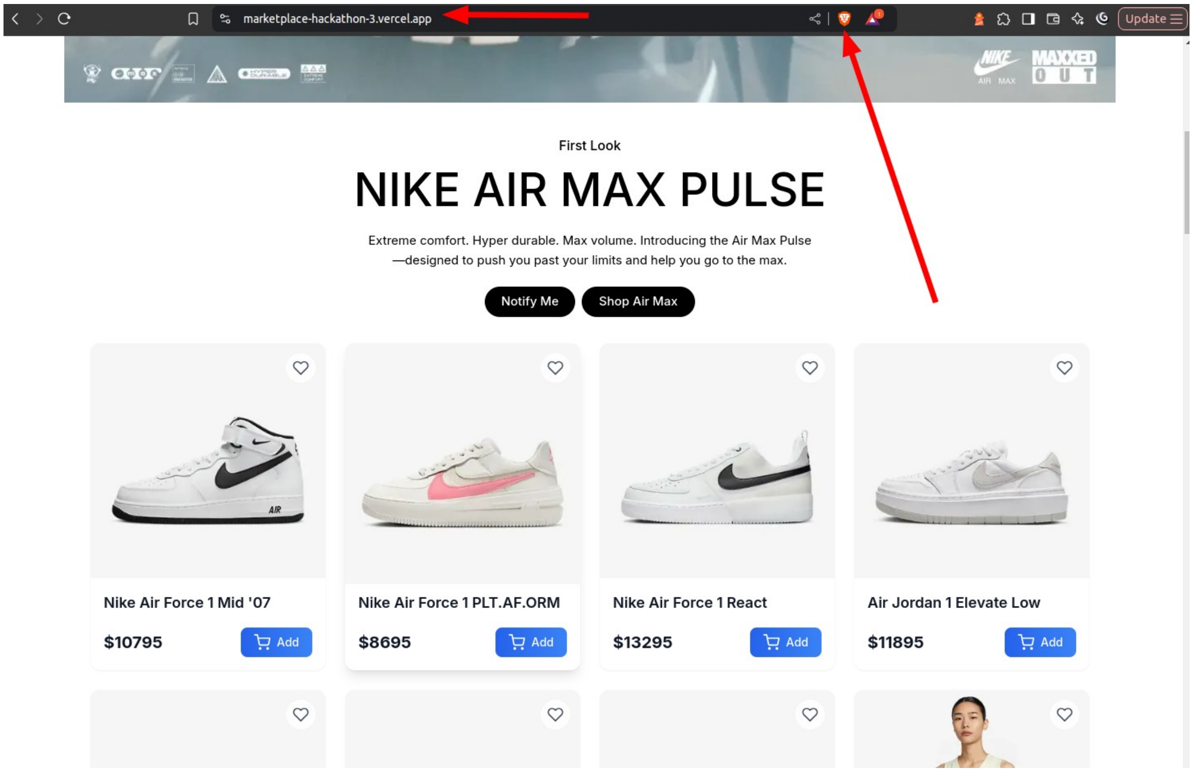
- **Vulnerability Scanning:** Conducted automated scans to identify potential security threats.
- **Data Encryption:** Implemented encryption for sensitive data, including user passwords and payment details.
- **Authentication & Authorization:** Strengthened authentication mechanisms to prevent unauthorized access.
- **Input Validation:**
  - Applied regex validation for form inputs (e.g., email, phone numbers).
- **API Security Measures:**
  - Enforced HTTPS for all API requests.
  - Implemented token-based authentication.

Results: No major security vulnerabilities detected. Security measures effectively implemented.

5. Cross-Browser and Device Compatibility

- **Browsers Tested:** Google Chrome, Mozilla Firefox, Microsoft Edge, and Safari.
- **Devices Tested:** Desktop, tablet, and mobile devices.
- **Testing Methods:**
  - Used Developer Tools for debugging and performance monitoring.
  - Conducted manual testing on physical devices.
  - Verified consistent rendering and responsiveness across various screen sizes.

Results: The platform performed consistently across all tested browsers and devices.



First Look

NIKE AIR MAX PULSE

Extreme comfort. Hyper durable. Max volume. Introducing the Air Max Pulse —designed to push you past your limits and help you go to the max.

Nike Air Force 1 Mid '07	$10795
Nike Air Force 1 PLT.AF.ORM	$8695
Nike Air Force 1 React	$13295
Air Jordan 1 Elevate Low	$11895

6. Documentation and Reporting

- **Test Case Documentation:**
    - Created a CSV-based report documenting test cases, expected results, actual results, and resolutions.
    - Ensured compliance with industry standards for testing documentation.
- **Fallback UI Elements:**
    - Implemented retry buttons, placeholder content, and informative modals for API failures.

7. Challenges and Resolutions

Challenge	Resolution
Handling API errors without disrupting UX	Implemented fallback UI and retry mechanisms.
Ensuring smooth performance under high traffic	Optimized SSR processes and enabled caching.
Documenting test results professionally	Organized test cases into a CSV format for clear reporting.

Conclusion: By the end of Day 5, the marketplace was rigorously tested, optimized, and secured for deployment. With a focus on performance, security, and user experience, the platform is now ready for real-world use, ensuring a seamless and secure shopping experience for users.

Date: February 5, 2025