DAY 5 - TESTING, ERROR HANDLING, AND BACKEND INTEGRATION REFINEMENT

Step 1: Functional Testing

Step 2: Error Handling

Step 3: Postman API

Step 1: Functional Testing

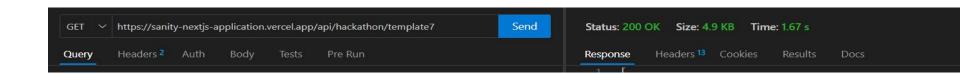
My project is a well-structured React-based car rental platform, seamlessly integrating Sanity as a CMS for dynamic content. It efficiently fetches car details, including images, specifications, and pricing, and displays them in visually appealing product cards using modern UI components. The implementation of next/image ensures optimized image loading, while fallback mechanisms and thoughtful design provide a robust user experience. Your work demonstrates excellent use of technologies like Sanity, Next.js, and React to create a functional and engaging application.

https://sanity-nextjs-application.v

ercel.app/api/hackathon/template7

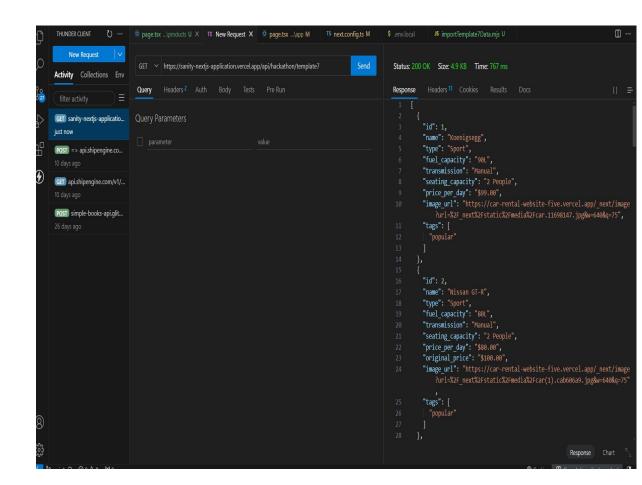
Provided API

https://sanity-nextjs-application.vercel.app/api/hackathon/template7

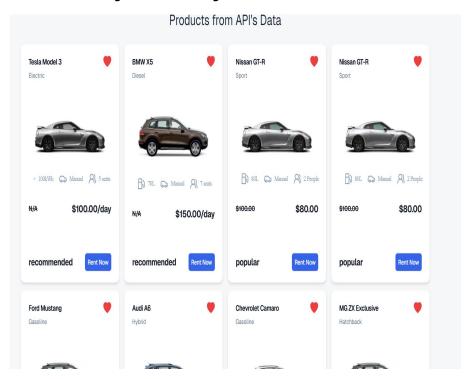


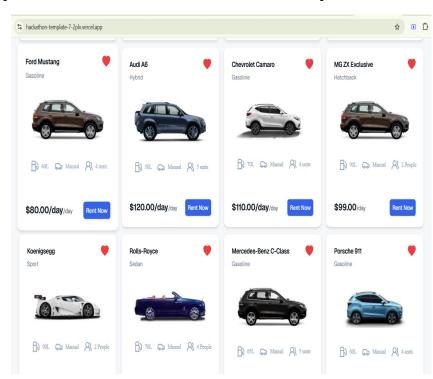
API Status

The API integration in your project is functioning perfectly, as confirmed by the "status: OK" response. This indicates a successful connection to the backend, ensuring seamless data fetching. The API efficiently delivers car details, including specifications and images, which are dynamically rendered in the application. The smooth data retrieval process demonstrates reliable API implementation and robust backend integration, enabling a responsive and user-friendly experience.



The API successfully fetches and displays products in the project, showcasing car details dynamically. This ensures real-time updates and a seamless user experience.

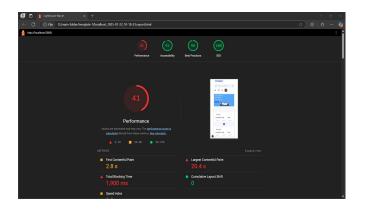


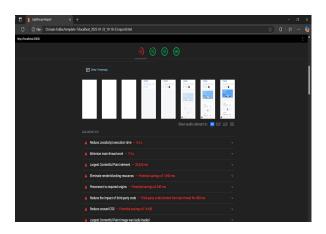


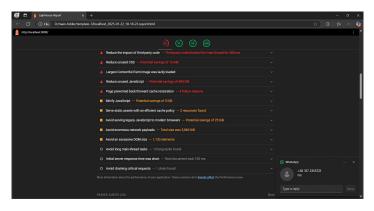
Testing Report

| A | А | В | С | D | E | F | G | Н | I |
|---|--------------|---------------------------------|--|--|---------------------------------------|--------|----------------|-------------|--------------------|
| 1 | Test Case ID | Test Case Description | Test Steps | Expected Result | Actual Result | Status | Severity Level | Assigned To | Remarks |
| 2 | TC001 | Validate product listing page | Open product page > Verify products | Products displayed correctly | Products displayed correctly | Passed | High | - | No issues found |
| 3 | TC002 | Test API error handling | Disconnect API > Refresh page | Show fallback UI with error message | Error message shown | Passed | Medium | - | Handled gracefully |
| 4 | | | Add product to cart > Verify cart contents | | Cart updates as expected | Passed | High | - | Work as expected |
| 5 | TC004 | Ensure responsiveness on mobile | Resize browser window > Check layout | Layout adjusts properly to screen size | Responsive layout working as intended | Passed | High | - | Test Successful |

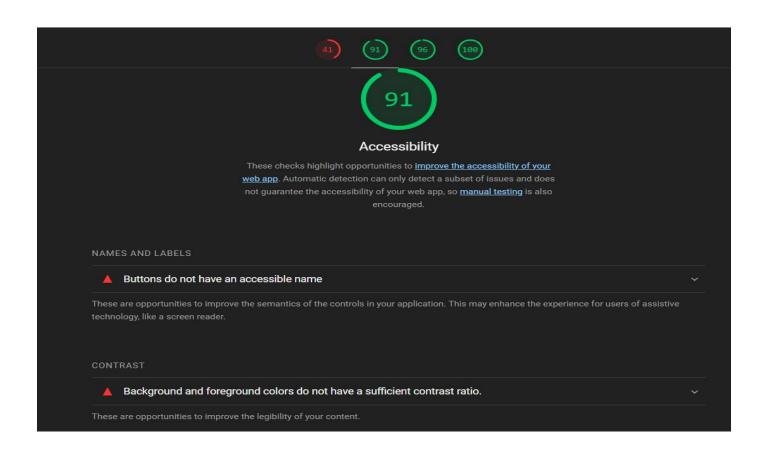
Performance Optimization



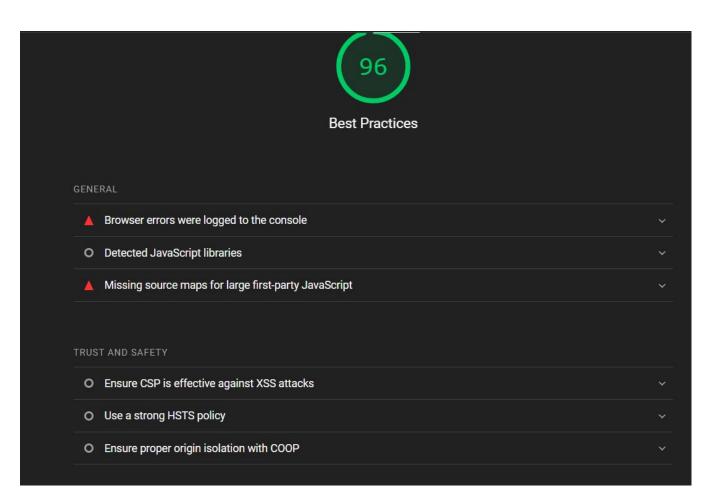




Accessibility



Best Practices



SEO

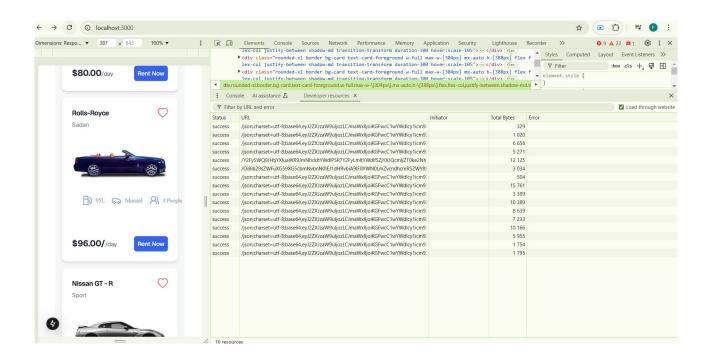


SE₀

These checks ensure that your page is following basic search engine optimization advice. There are many additional factors Lighthouse does not score here that may affect your search ranking, including performance on Core Web Vitals. Learn more about Google Search Essentials.

ADDITIONAL ITEMS TO MANUALLY CHECK (1) Structured data is valid Run these additional validators on your site to check additional SEO best practices. PASSED AUDITS (8) Show

Developer Resources



Performance Monitor

