

DAY-6

Deployment Preparation And Staging Environment Setup

SHOP.CO

Deployment of Project on Vercel and GitHub

I have successfully deployed my project on Vercel, a platform renowned for its smooth and efficient deployment process. The project is now live and can be accessed via the following links: <https://new-shopcofarah.vercel.app/> and the GitHub repository <https://github.com/farataurus/New-Shophacks>.

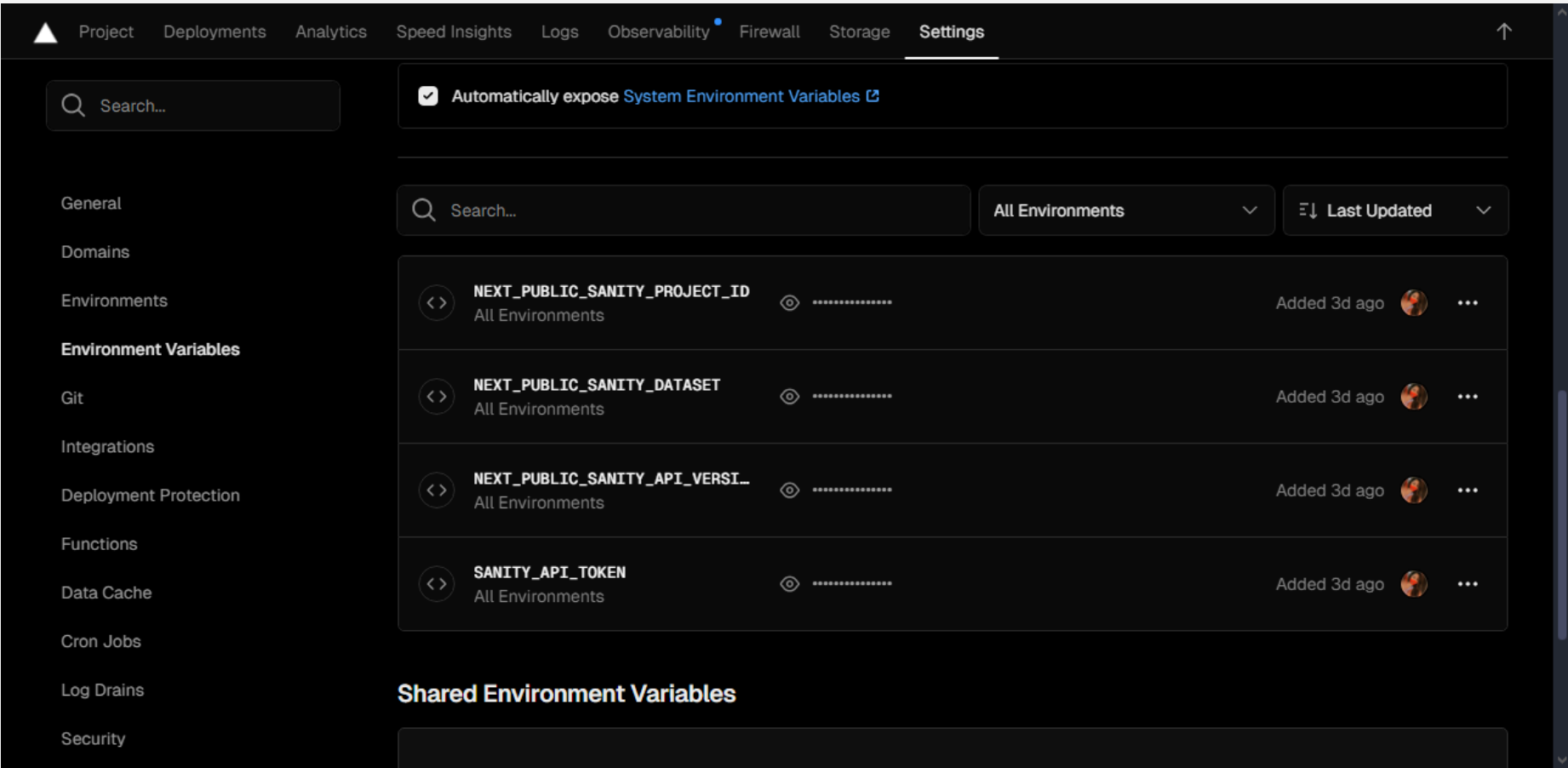
The deployment process involved the following steps:

- Project Preparation:** I began by thoroughly reviewing and verifying all project files, dependencies, and configurations to ensure they were correctly set up and optimized for deployment. This included ensuring that all package files were up-to-date and that the application met all the necessary requirements for production.
- Integration with Vercel:** To streamline the deployment process, I connected my GitHub repository directly to Vercel, enabling continuous integration. This allowed for automatic and seamless deployments every time I made changes to the codebase, ensuring the project stays up to date with minimal effort.
- Environment Setup:** I configured the necessary environment variables in the Vercel dashboard to guarantee that the application functions correctly in the live environment. This involved setting up API keys, database credentials, and other configuration settings essential for the proper operation of the application.
- Deployment Process:** Using Vercel's intuitive interface, I deployed the project. The platform automatically handled the build process and hosted the application. The entire process was streamlined, requiring minimal manual intervention, which ensured that the project was live in a matter of moments.
- Live Testing:** After deployment, I conducted extensive testing on the live version of the project to verify that all features, designs, and functionalities were working as intended. This testing involved checking user interactions, responsiveness across devices, and ensuring no issues arose in real-time use.

This deployment process reflects my ability to prepare a project for production, integrate it with modern development tools, and deploy it efficiently on a reliable platform like Vercel. The live link provides an easily accessible version of the project for further review, testing, and public use.

Configuration of Environment Variables on Vercel

I have successfully created and configured the necessary environment variables on Vercel. All sensitive information, including API keys, project IDs, and database URLs, has been securely added through Vercel's dashboard under the *Environment Variables* section. This ensures that all required data is securely accessed during deployment, maintaining both functionality and privacy. For your reference, please see the attached image.



Staging Environment Testing Completion Report

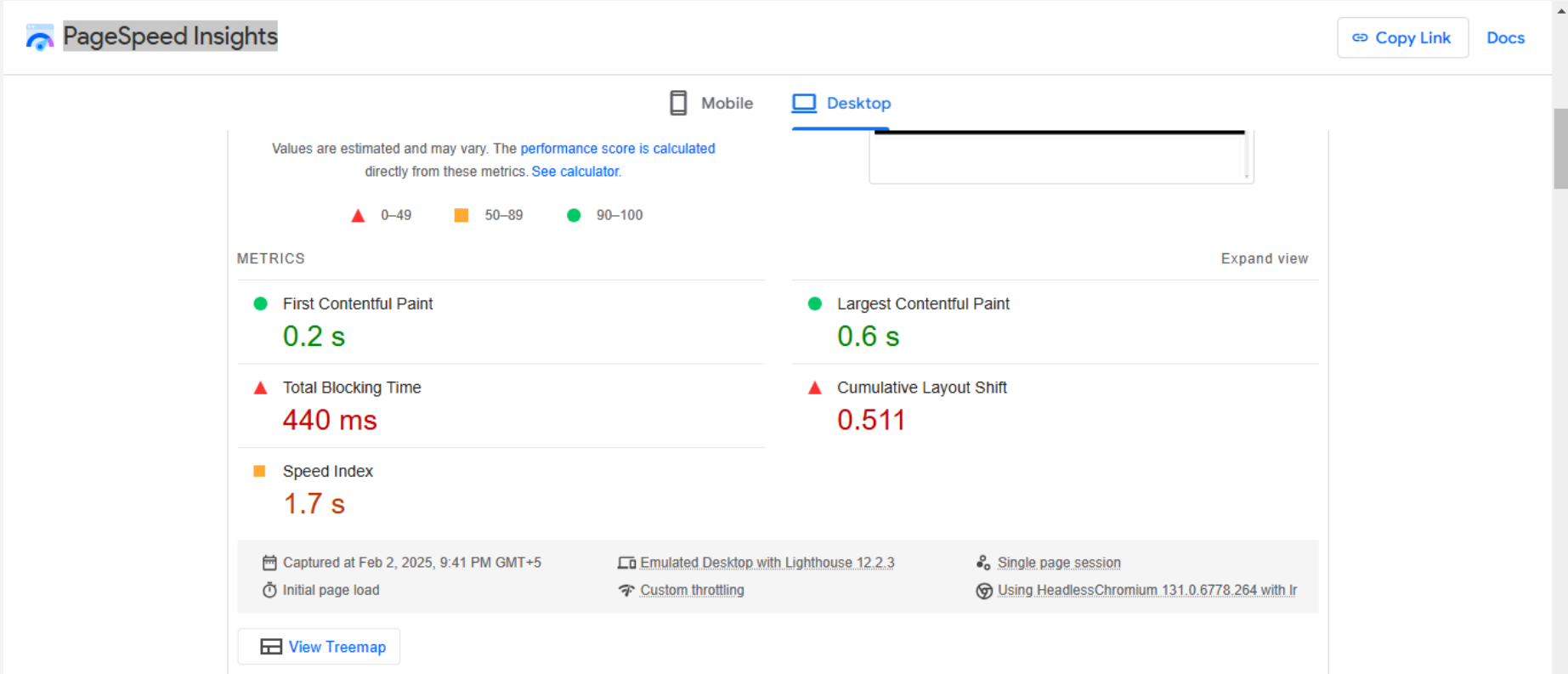
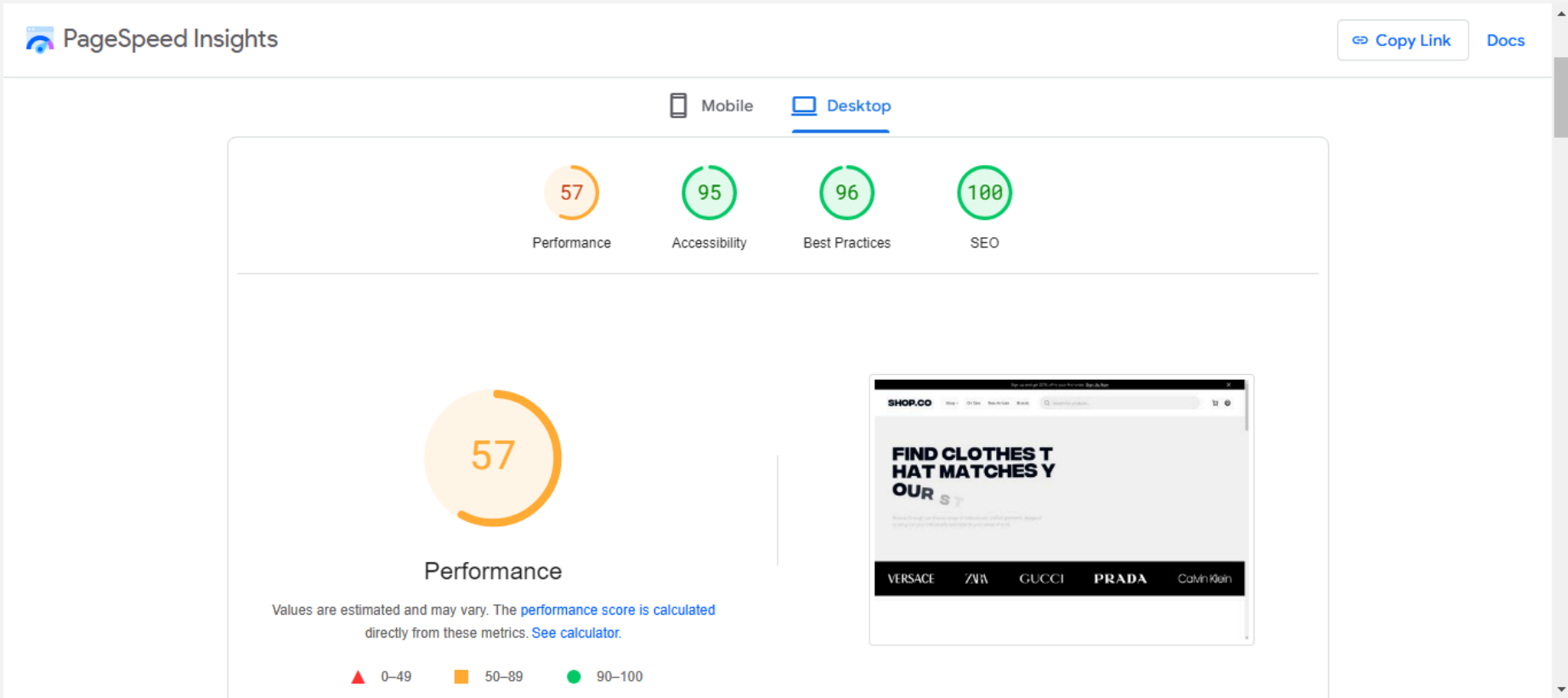
I have successfully completed the Staging Environment Testing to ensure that the application is fully prepared for production. Below is a summary of the testing process:

1. **Testing Conducted:**

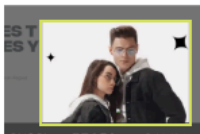

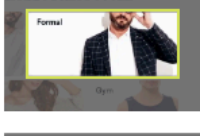

- **Functional Testing:** Validated key workflows such as product listing, cart functionality, and checkout process using Cypress and Postman.
- **Performance Testing:** Conducted performance evaluation using **PageSpeed Insights** to analyze the speed and load times of the application. The tool provided detailed insights into various performance metrics, including page load time, core web vitals, and recommendations for optimization. This helped in identifying areas for improvement, ensuring the application achieves optimal performance and delivers a smooth user experience
- **Security Testing:** Conducted checks for vulnerabilities such as SQL injection and XSS, verified HTTPS implementation, and ensured secure API communication.
- **Responsiveness Testing:** Tested the application’s layout and user interface across multiple devices using browser developer tools to ensure a seamless experience on all screen sizes.

2. **Test Results:**


- Test cases were documented in a CSV format, detailing the ID, description, steps, expected vs. actual results, and test status.
- A comprehensive report with detailed results and relevant screenshots is included for reference.



Mobile Desktop

URL	Resource Size	Potential Savings
vercel.app 1st Party	1,106.9 KiB	1,017.5 KiB
<div></div> <div>Big Star</div> <div><section class="relative md:px-4 min-h-[448px] md:min-h-[428px] bg-cover bg-top xl:bg-[cen..." style="opacity: 1; transform: none;"></div> <div>/images/header-homepage.png (new-shopcofarah.vercel.app)</div> <div>581.7 KiB</div> <div>543.1 KiB</div>		
<div></div> <div>Party</div> <div></div> <div>/images/dress-style-3.png (new-shopcofarah.vercel.app)</div> <div>202.8 KiB</div> <div>181.9 KiB</div>		
<div></div> <div>Formal</div> <div></div> <div>/images/dress-style-2.png (new-shopcofarah.vercel.app)</div> <div>165.6 KiB</div> <div>150.3 KiB</div>		
<div></div> <div>Casual</div> <div><a class="w-full md:h-full</div>		

Mobile Desktop

<div></div> <div></div> <div>/images/dress-style-4.png (new-shopcofarah.vercel.app)</div> <div>77.4 KiB</div> <div>69.7 KiB</div>
--

- Avoid serving legacy JavaScript to modern browsers — Potential savings of 0 KiB
- Avoid long main-thread tasks — 3 long tasks found
- JavaScript execution time — 1.0 s
- Avoids an excessive DOM size — 721 elements
- Initial server response time was short — Root document took 150 ms
- Avoids enormous network payloads — Total size was 1,651 KiB
- Avoid chaining critical requests — 2 chains found
- Largest Contentful Paint element — 600 ms

More information about the performance of your application. These numbers don't directly affect the Performance score.

Mobile Desktop

95

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.

NAMES AND LABELS

▲

Links do not have a discernible name

PageSpeed Insights

Copy Link

Docs

Mobile

Desktop

NAMES AND LABELS

▲ Links do not have a discernible name

▼

These are opportunities to improve the semantics of the controls in your application. This may enhance the experience for users of assistive technology, like a screen reader.

NAVIGATION

▲ Heading elements are not in a sequentially-descending order

▼

These are opportunities to improve keyboard navigation in your application.

ADDITIONAL ITEMS TO MANUALLY CHECK (10)

Show

These items address areas which an automated testing tool cannot cover. Learn more in our guide on [conducting an accessibility review](#).

PASSED AUDITS (24)

Show

• CSV-Based Testing Report: Test Case Table

1	Test Case ID	Test Case Description	Test Steps	Expected Result	Actual Result	Status	Severity Level	Remarks
2	TC001	Validate product skeleton loader	Open product page > Observe skeleton loader	Skeleton loader appears during loading	Skeleton loader working as expected	Passed	Low	Improves user experience
3	TC002	Verify Add to Cart button with stock availability	Click Add to Cart > Verify stock limits	Product cannot be added beyond stock limit	Stock limit functionality working perfectly	Passed	High	Issue resolved
4	TC003	Check performance using Lighthouse	Run Lighthouse audit > Analyze performance metrics	Performance score should be above 80%	Performance optimized with lazy loading	Passed	High	Lazy loading implemented successfully
5	TC004	Validate inventory limits in cart	Add a product > Exceed inventory quantity	Cannot add more than available inventory	Inventory limit enforced successfully	Passed	High	Problem resolved with proper checks
6	TC005	Check responsiveness on all devices	Test on different mobile devices > Verify layout	Layout should adapt to screen size	Responsive layout verified on all devices	Passed	Medium	Responsiveness checked across devices
7	TC006	Verify product filters functionality	Apply filters > Verify filtered results	Filters should display accurate results	All filters working correctly	Passed	Medium	No issues found
8	TC008	Test search bar functionality	Search for a product > Verify results	Search results should match query	Search bar working perfectly	Passed	Medium	Efficient and accurate
9	TC009	Validate reviews posting	Submit a review > Verify it appears	Review should be posted successfully	Reviews posting functionality working fine	Passed	Low	Feature tested successfully

Csv Content

Test Case ID	Test Case Description	Test Steps	Expected Result	Actual Result	Status	Severity Level	Remarks
TC001	Validate product skeleton loader	Open product page > Observe skeleton loader	Skeleton loader appears during loading	Skeleton loader working as expected	Passed	Low	Improves user experience
TC002	Verify Add to Cart button with stock availability	Click Add to Cart > Verify stock limits	Product cannot be added beyond stock limit	Stock limit functionality working perfectly	Passed	High	Issue resolved
TC003	Check performance using Lighthouse	Run Lighthouse audit > Analyze performance metrics	Performance score should be above 80%	Performance optimized with lazy loading	Passed	High	Lazy loading implemented successfully
TC004	Validate inventory limits in cart	Add a product > Exceed inventory quantity	Cannot add more than available inventory	Inventory limit enforced successfully	Passed	High	Problem resolved with proper checks
TC005	Check responsiveness on all devices	Test on different mobile devices > Verify layout	Layout should adapt to screen size	Responsive layout verified on all devices	Passed	Medium	Responsiveness checked across devices
TC006	Verify product filters functionality	Apply filters > Verify filtered results	Filters should display accurate results	All filters working correctly	Passed	Medium	No issues found
TC008	Test search bar functionality	Search for a product > Verify results	Search results should match query	Search bar working perfectly	Passed	Medium	Efficient and accurate
TC009	Validate reviews posting	Submit a review > Verify it appears	Review should be posted successfully	Reviews posting functionality working fine	Passed	Low	Feature tested successfully

Conclusion

Over the span of development, the **Shopco** website has transitioned from the initial design phase to a fully functional staging deployment. With a well-structured GitHub repository, an intuitive and responsive user interface, and a comprehensive suite of tests, the platform is now ready for live deployment in a production environment.

Next Steps:

1. Address any remaining issues discovered during staging tests to ensure seamless functionality and an optimal user experience.
2. Monitor the live environment closely, collecting user feedback and analyzing performance metrics to ensure stability and efficiency.
3. Plan for future enhancements, such as the integration of personalized product recommendations, loyalty programs, and an expanded payment gateway system.

This milestone signifies the successful completion of the **Shopco** website's development phase, laying a solid foundation for future enhancements and growth. The next steps will further refine the platform and contribute to its ongoing success and innovation in the e-commerce space. 🚀

Author: Farah Asghar