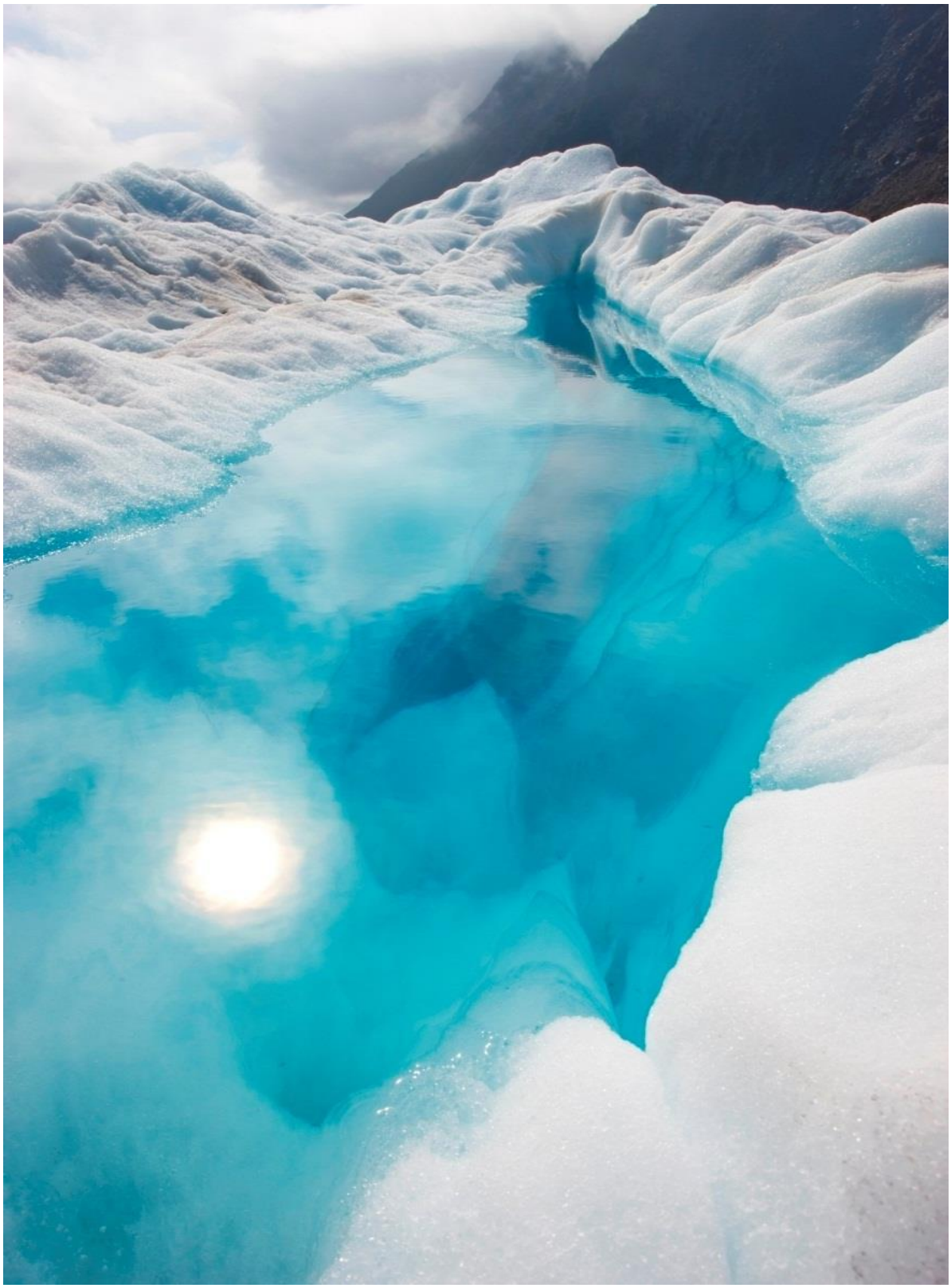
ARJUMAND AFREEN TABINDA ROLL NO:00443645



Hackathon Day 6: Deployment of Project on Netlify

MADE B Y:-ARJUMAND AFREEN TABINDA

Objective:

To successfully deploy the project on Netlify, configure environment variables, set up a staging environment, and conduct various tests to ensure the functionality, performance, and security of the application.

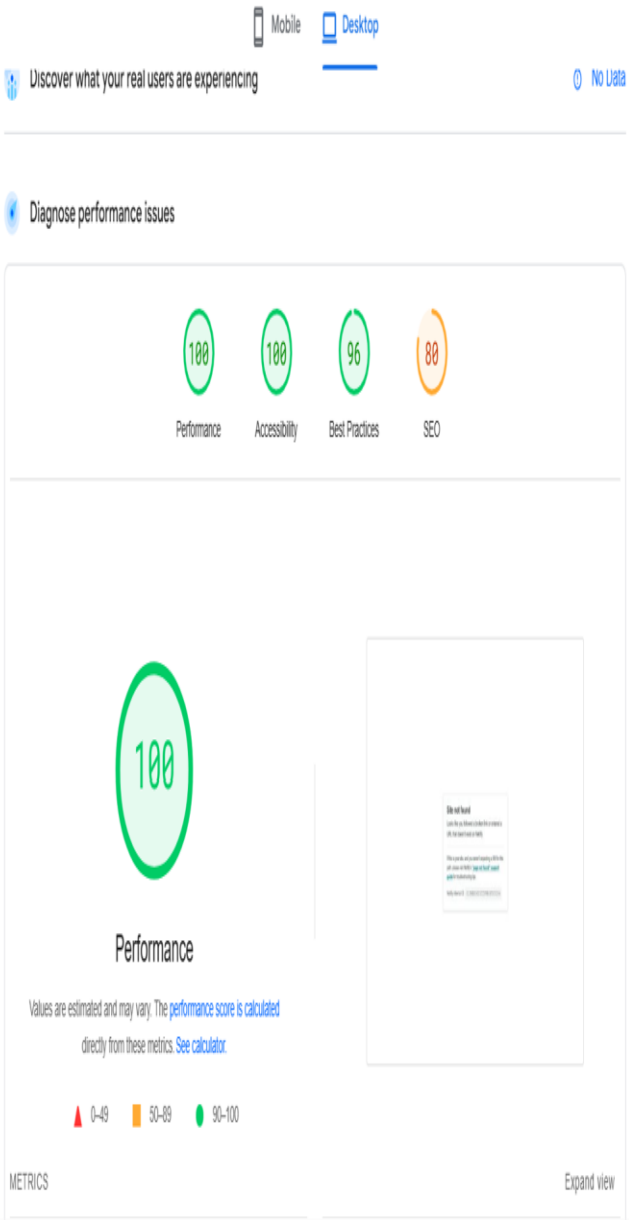
1. Deployment Strategy Planning

- *Chosen Hosting Platform: Netlify*
- *After evaluating various hosting platforms such as Vercel, AWS, and Azure, Netlify was selected as the hosting platform due to its ease of use, integration with Next.js, and automatic optimization for production deployments.*
- *Backend Services Integration:*

- *The application communicates with Sanity CMS for managing content, with API requests set up to dynamically retrieve data from the backend.*
- *Third-party APIs are incorporated to enable functionalities like payments, notifications, and other external services.*
- *Ensured seamless integration and proper handling of API responses throughout the deployment process.*
- *2. Configuring Environment Variables*
- *Securing API Keys and Sensitive Information:*
- *.env files were utilized locally to store confidential details such as API keys, database credentials, and other private configurations.*

- *When deploying on Netlify, these environment variables were set up within Netlify's configuration settings to protect sensitive data while keeping it hidden from the codebase.*
- *Netlify Configuration:*
- *Defined environment variables within Netlify's dashboard to ensure seamless connectivity with backend services and third-party APIs. This guarantees secure access to API credentials and sensitive data exclusively in the deployed environment.*
- *3. Setting Up the Staging Environment*
- *Staging Deployment:*
  The project was deployed to a staging environment on Netlify for testing. This environment replicates the live setup, allowing verification of the application's behavior before pushing to production.
- *Build Validation:*
- *Monitored the deployment process to confirm a successful build.*
- *Ensured the site loaded without issues, verifying that all assets, including images and styles, were correctly displayed.*

- ***4. Testing in the Staging Environment***
- ***Functional Testing:***
- ***Cypress: Cypress was utilized to test user workflows and interactions.***
- ---
- ***Postman: Used to verify API responses, ensuring that backend services, including Sanity CMS and third-party integrations, returned accurate and expected data.***
- ***Performance Testing:***
- ***PageSpeed Insights: Utilized to evaluate the website's performance, emphasizing loading speed, responsiveness, and overall efficiency.***
- ***The analysis revealed a perfect 100% performance score, demonstrating excellent optimization for core web vitals. Check the results here:***



-
-

Security Assessment:

- Examined input fields to prevent SQL injection and other potential security vulnerabilities.

- *Confirmed that the application was delivered over HTTPS for secure data transmission.*
- *Reviewed the handling of sensitive information, such as API keys, ensuring they remained hidden from the frontend code.*

Responsiveness & Error Management:

- *Tested the application's adaptability across different screen sizes, ensuring smooth display on both mobile and desktop devices.*
- *Evaluated error-handling processes to confirm that issues like failed API requests or user errors were managed effectively and presented clearly to users.*

Security Assessment:

- *Examined input fields to prevent SQL injection and other potential security vulnerabilities.*
- *Confirmed that the application was delivered over HTTPS for secure data transmission.*
- *Reviewed the handling of sensitive information, such as API keys, ensuring they remained hidden from the frontend code.*

Responsiveness & Error Management:

- *Tested the application's adaptability across different screen sizes, ensuring smooth display on both mobile and desktop devices.*
- *Evaluated error-handling processes to confirm that issues like failed API requests or user errors were managed effectively and presented clearly to users.*

5. Documentation and Outstanding Issues

Test Outcomes:

- *All functionality, performance, and security tests were completed successfully. The Lighthouse performance score exceeded 85%, with no significant security risks detected.*
- *Ensured that the application was fully responsive and that error-handling mechanisms functioned correctly.*

Outstanding Issues:

- *No critical issues remained unresolved during deployment and testing. Minor UI refinements were implemented based on mobile*

responsiveness evaluations, but these had no major impact on performance or functionality.

Final Thoughts:

- *The project was successfully deployed on Netlify and underwent thorough testing in the staging environment. All features were verified and confirmed to be working as intended. The website passed all*

- *functional, performance, and security assessments and is now ready for live deployment.*
- *Netlify project link:*

https://clinquant-tarsier-4dc82a.netlify.app/