Day 6

Deployment Preparation and Staging Environment Setup

(comforty)

Table of Content:

1. Overview.
2. Selecting a hosting platform.
3. Connecting Github.
4. Configuring build and deployment settings.
5. Setting up environment variable.
6. Checking website functionality after deployment.
7. Performance Testing.
8. Test Case Report.

Overview:

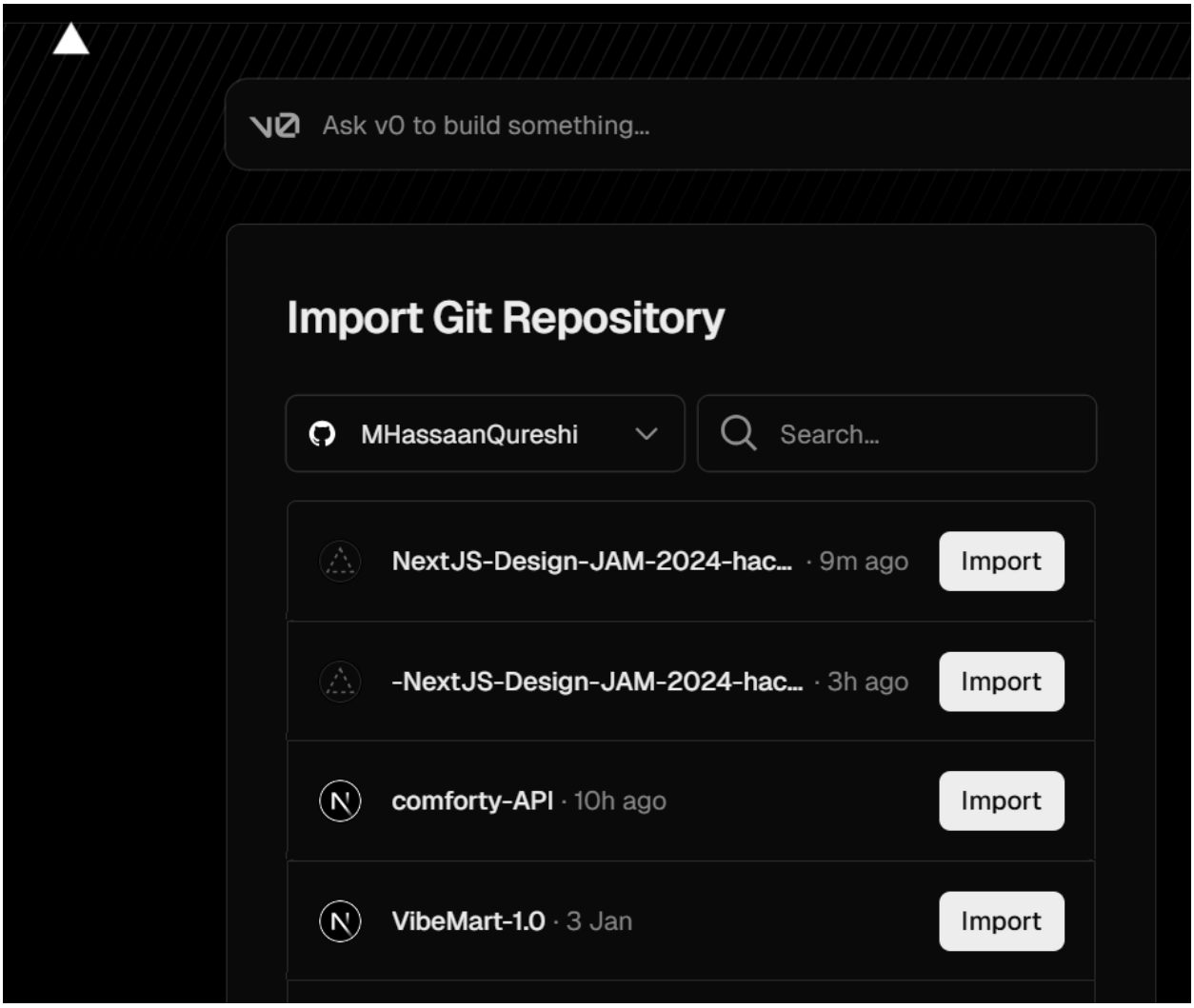
This report details the process and considerations for deploying a web application effectively and ensuring optimal performance.it include selecting a deployment platform, connecting your github, setting up your environment variable and performance testing.

Selecting a hosting environment:

For this project, we are going to use **Vercel** for the deployment of our application due to its simplicity, speed, and seamless integration with modern web development frameworks like Next.js. Vercel offers a hassle-free setup process, enabling us to connect our GitHub repository and automate the deployment pipeline efficiently. Its global Content Delivery Network (CDN) ensures fast load times and optimal performance for users across the globe. Additionally, Vercel's support for serverless functions, custom domains, and real-time preview links makes it a robust and developer-friendly choice for deploying modern web applications. With these advantages, Vercel provides the perfect platform to ensure our project is scalable, secure, and accessible to users everywhere.

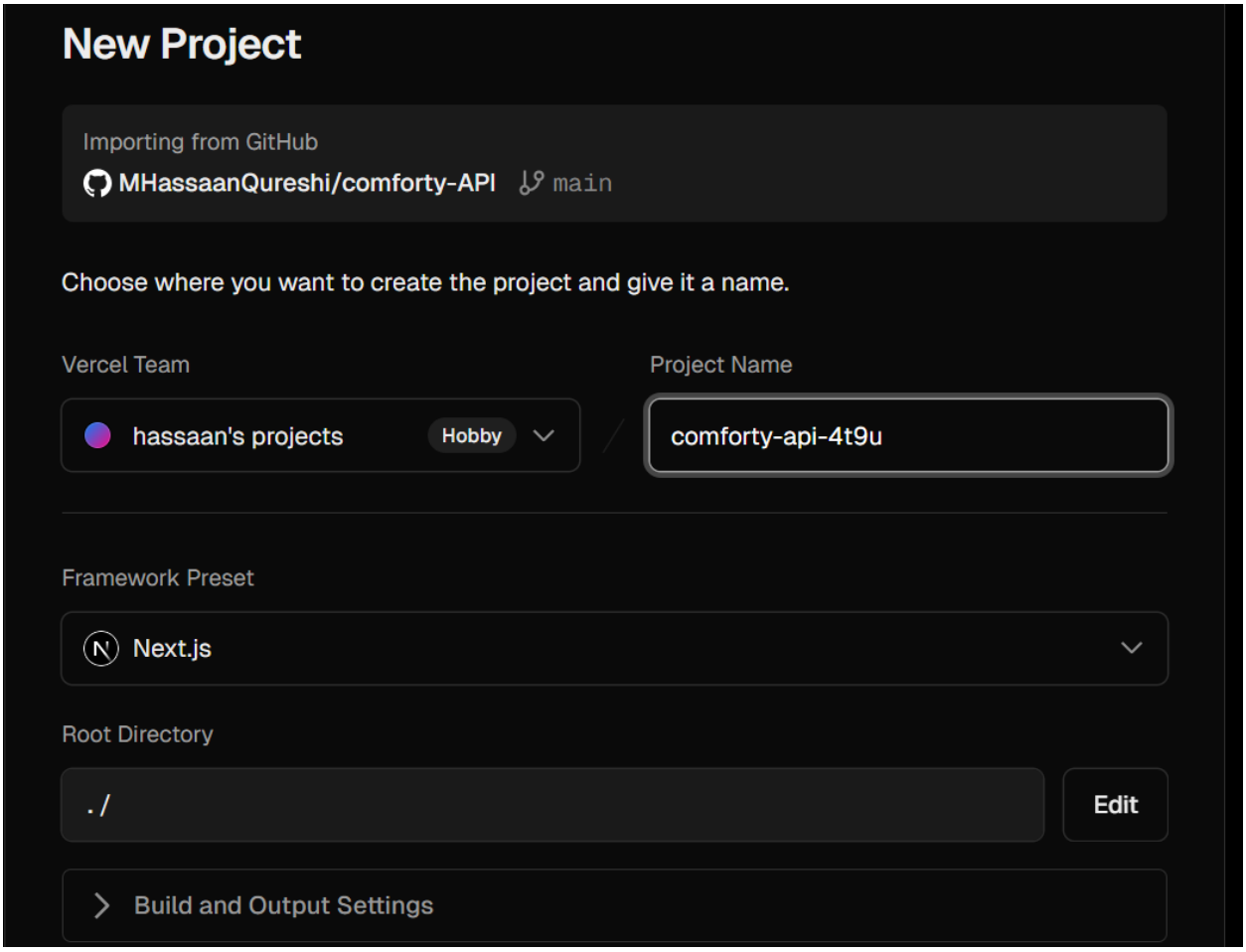
Connecting Github account:

After selecting we will connect our github account with vercel to import our project for easy deployment.

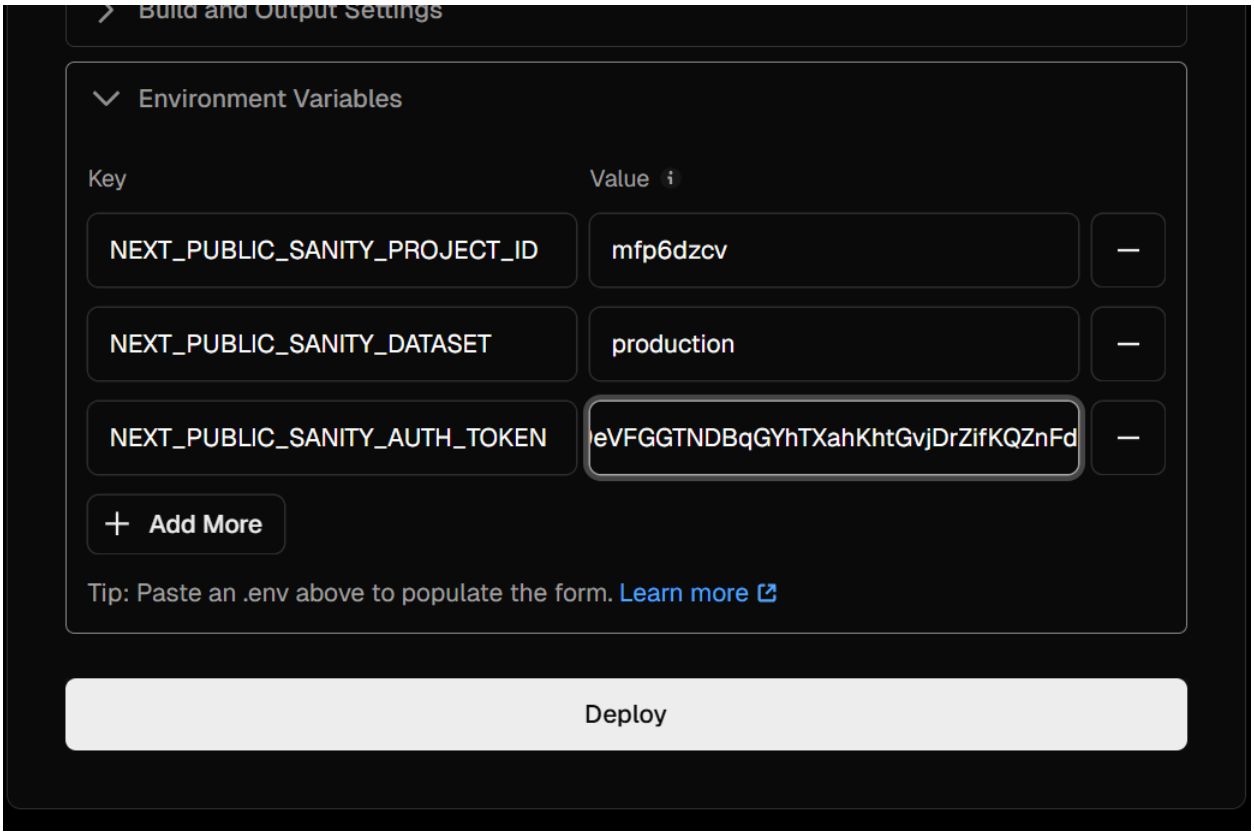


Setting up environment variable:

After importing our project this will pe our interface.



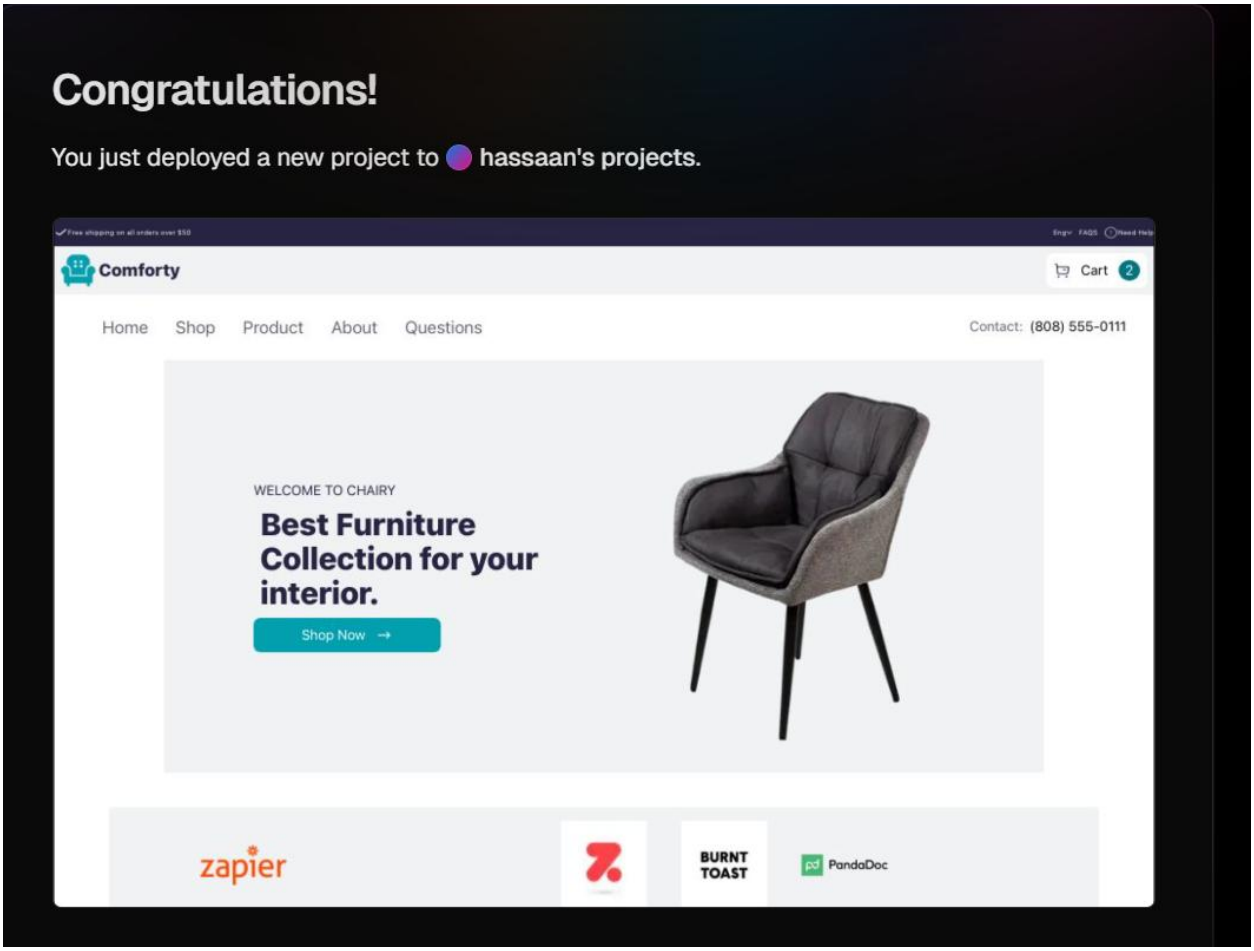
Here you will add your project name and select the root folder. below root folder we have environment variable part. That where we are going to include our environment variable that we have in our .env file. From there we are gong to enter our environment variable there.



After that click deploy. If you don’t include environment variable you would face error and failed deployment.

Checking Website functionality after deployment:

After successful deployment:



Check your website functionality like cart functionalities or check if

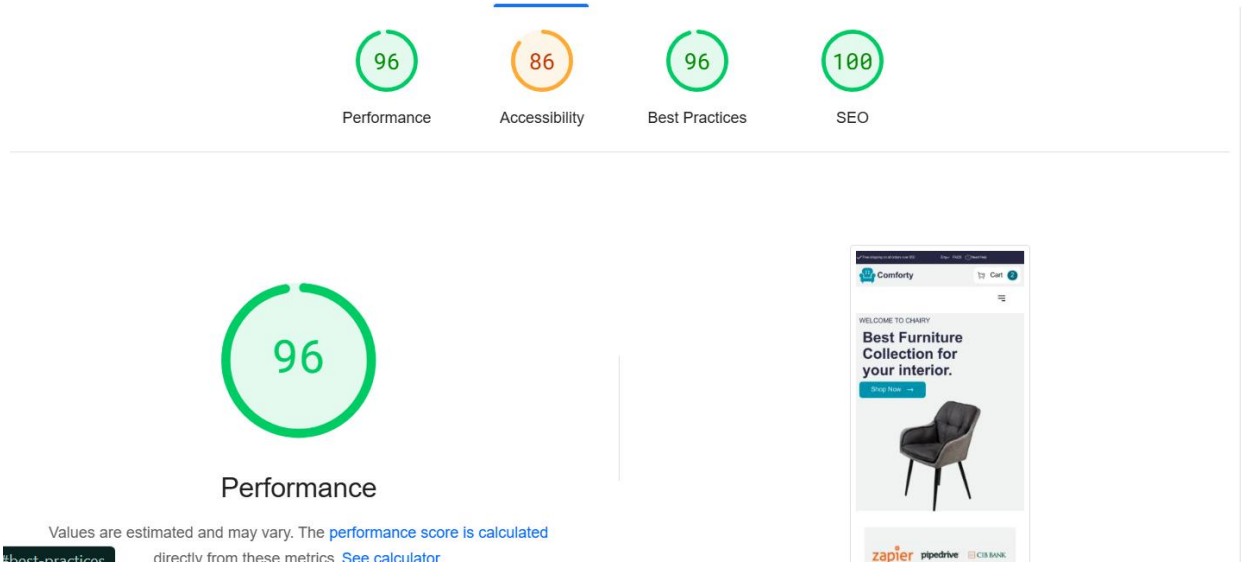
Your API is fetching data correctly notifications functionality and other functionality.

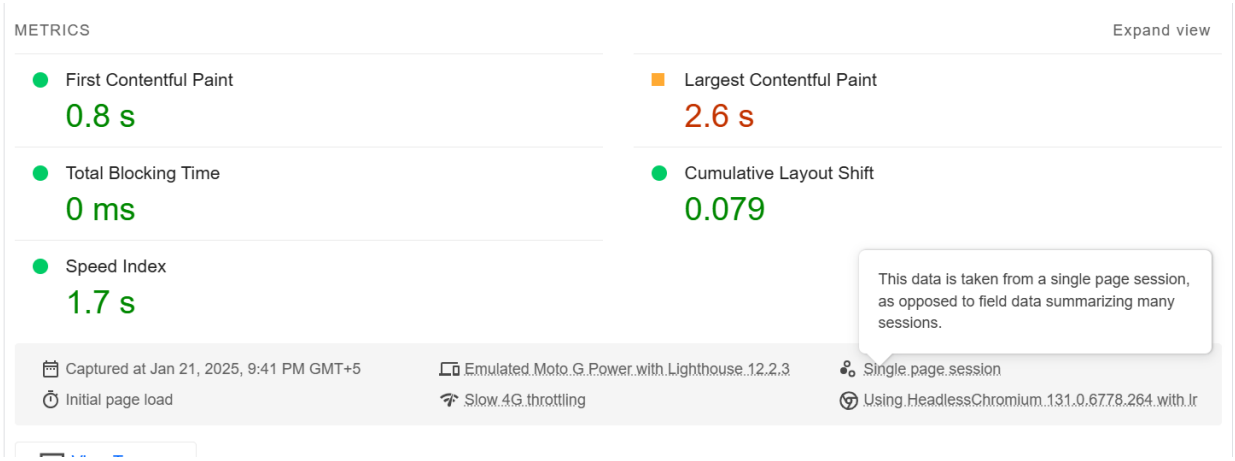
Performance testing:

After checking website functionality we are going to check our website performance.

To check a website performance there are multiple website to check performance. But for our comforty I used lighthouse and

these are the result I got of my website:
For Home page:
Small devices:

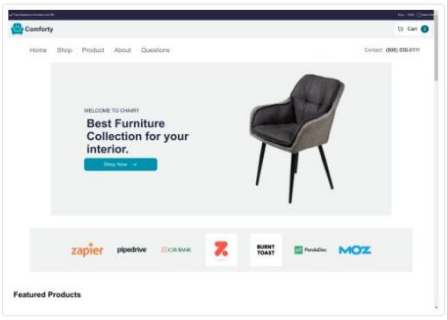


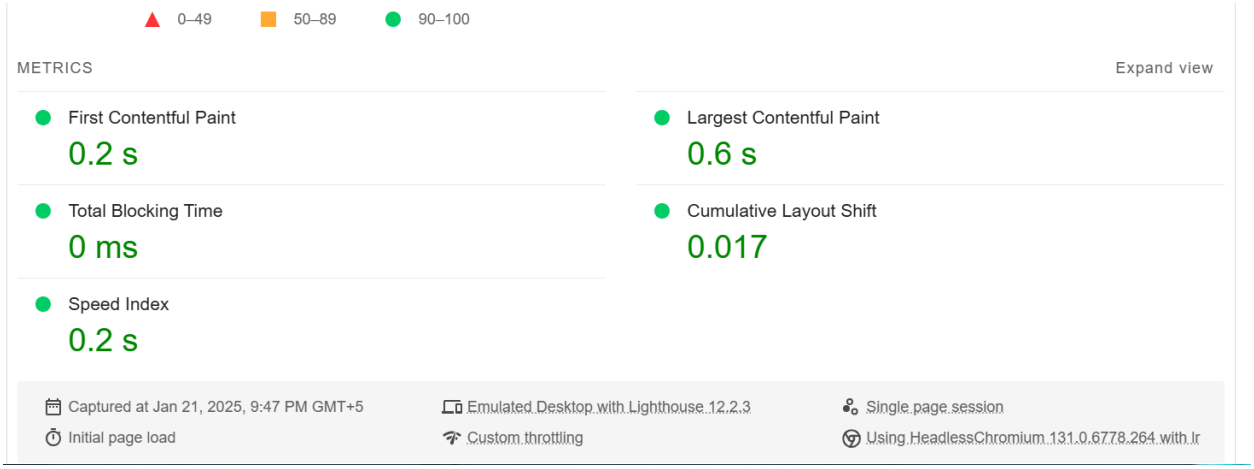


For desktop:

Performance

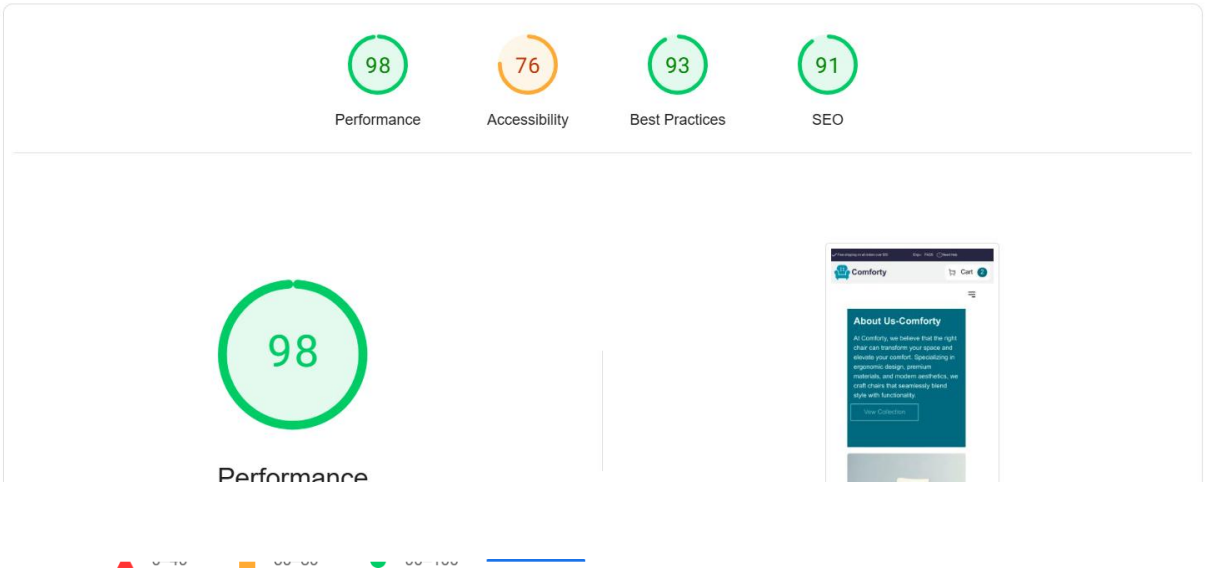
Values are estimated and may vary. The performance score is calculated directly from these metrics. See calculator.





For About page:

Small devices:

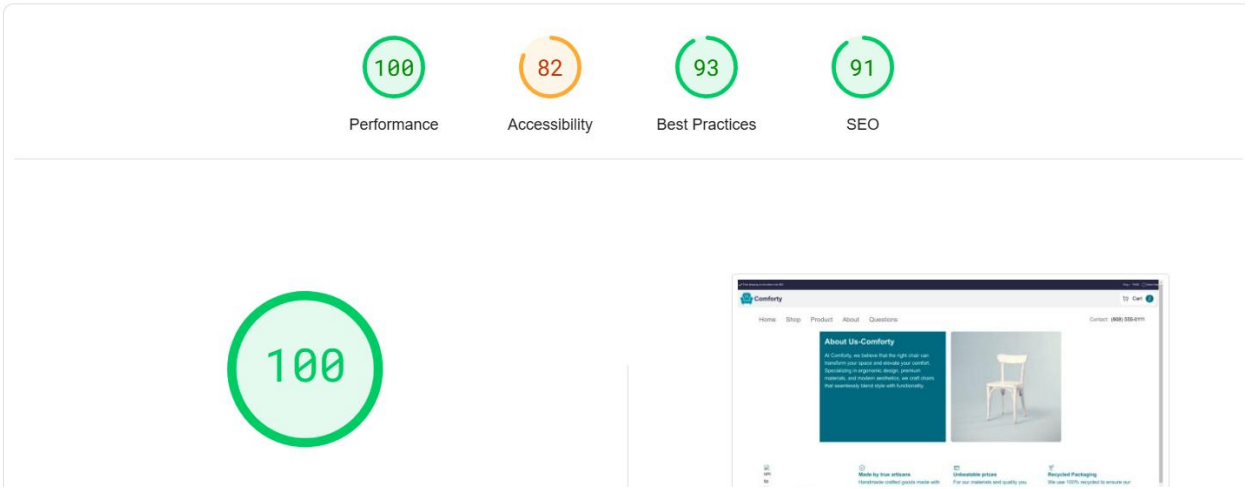


ETRICS Expand view

- First Contentful Paint 0.8 s
- Largest Contentful Paint 2.5 s
- Total Blocking Time 0 ms
- Cumulative Layout Shift 0.01
- Speed Index 0.9 s

Captured at Jan 21, 2025, 9:44 PM GMT+5
Initial page load
Emulated Moto G Power with Lighthouse 12.2.3
Slow 4G throttling
Single page session
Using HeadlessChromium 131.0.6778.264 with lr

For desktop:



▲ 0–49 ■ 50–89 ● 90–100

ETRICS Expand vi

● First Contentful Paint
0.2 s

● Largest Contentful Paint
0.6 s

● Total Blocking Time
0 ms

● Cumulative Layout Shift
0.015

● Speed Index
0.5 s

Captured at Jan 21, 2025, 9:44 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

View Treemap

Test Case Report:

I have included the test case report and test case file in .csv format separately in the same folder.