Experiment 11

Aim :

To use google Lighthouse PWA Analysis Tool to test the PWA functioning.

Theory :

Reference : https://www.semrush.com/blog/google-lighthouse/

Google Lighthouse :

Google Lighthouse is a tool that lets you audit your web application based on a number of parameters including (but not limited to) performance, based on a number of metrics, mobile compatibility, Progressive Web App (PWA) implementations, etc. All you have to do is run it on a page or pass it a URL, sit back for a couple of minutes and get a very elaborate report, not much short of one that a professional auditor would have compiled in about a week.

The best part is that you have to set up almost nothing to get started. Let's begin by looking at some of the top features and audit criteria used by Lighthouse.

Key Features and Audit Metrics

Google Lighthouse has the option of running the Audit for Desktop as well as mobile version of your page(s). The top metrics that will be measured in the Audit are:

Performance: This score is an aggregation of how the page fared in aspects such as (but not limited to) loading speed, time taken for loading for basic frame(s), displaying meaningful content to the user, etc. To a layman, this score is indicative of how decently the site performs, with a score of 100 meaning that you figure in the 98th percentile, 50 meaning that you figure in the 75th percentile and so on.

PWA Score (Mobile): Thanks to the rise of Service Workers, app manifests, etc., a lot of modern web applications are moving towards the PWA paradigm,

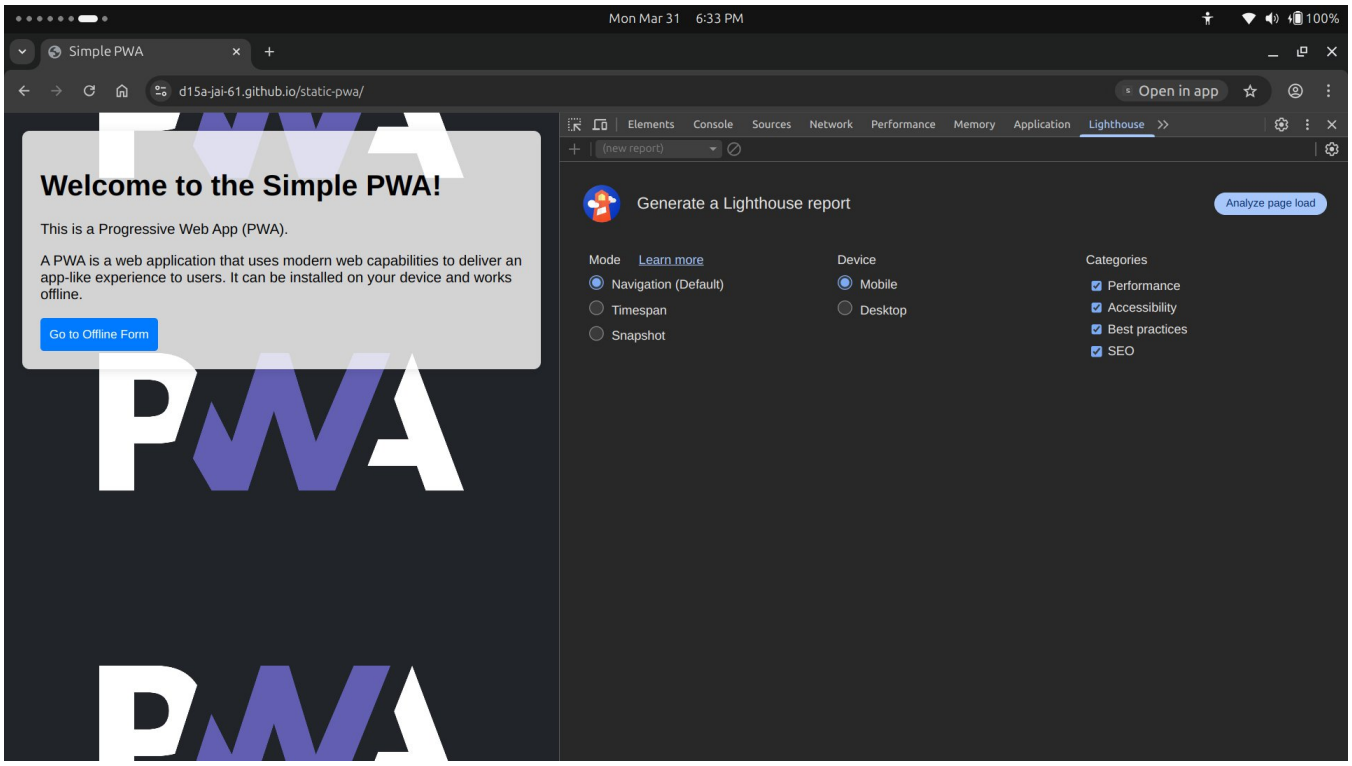
where the objective is to make the application behave as close as possible to native mobile applications. Scoring points are based on the Baseline PWA checklist laid down by Google which includes Service Worker implementation(s), viewport handling, offline functionality, performance in script-disabled environments, etc.

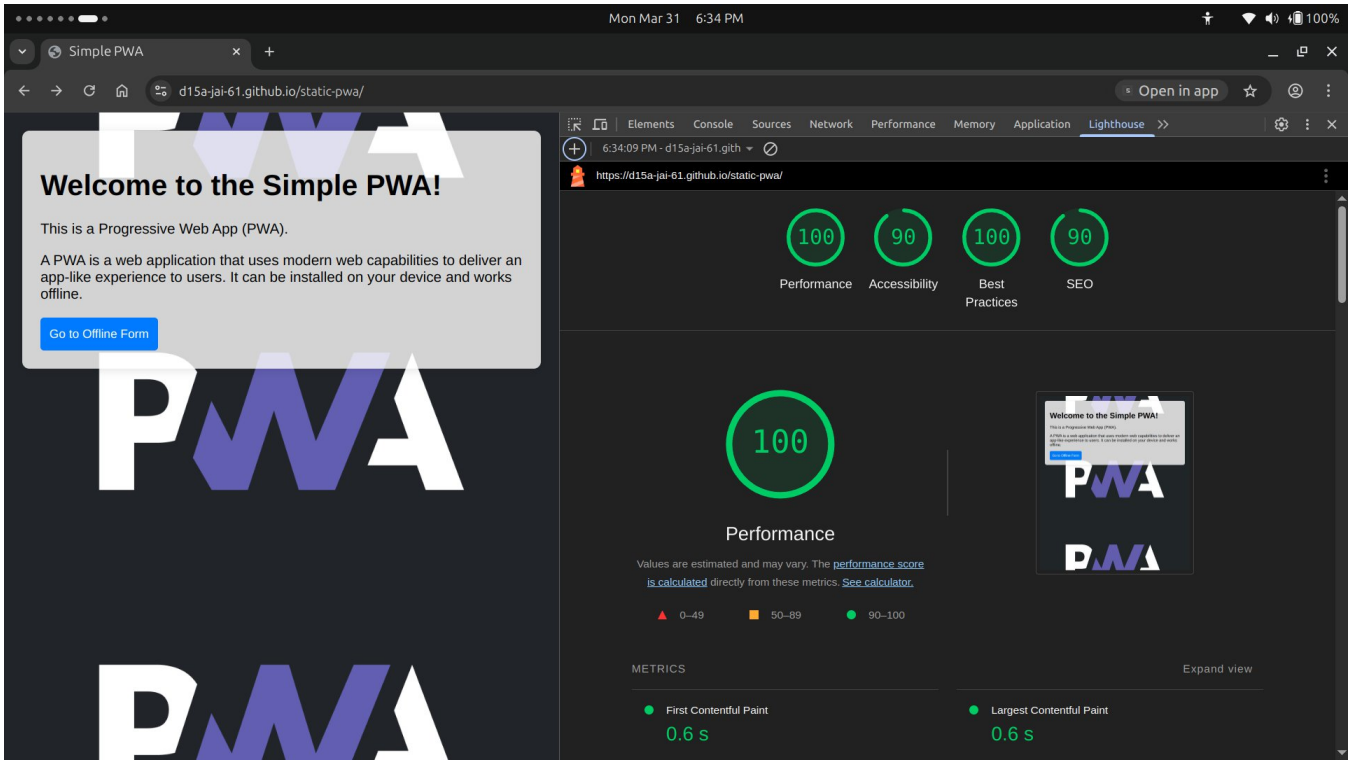
Accessibility: As you might have guessed, this metric is a measure of how accessible your website is, across a plethora of accessibility features that can be implemented in your page (such as the ‘aria-’ attributes like aria-required, audio captions, button names, etc.). Unlike the other metrics though, Accessibility metrics score on a pass/fail basis i.e. if all possible elements of the page are not screen-reader friendly (HTML5 introduced features that would make pages easy to interpret for screen readers used by visually challenged people like tag names, tags such as <section>, <article>, etc.), you get a 0 on that score. The aggregate of these scores is your Accessibility metric score.

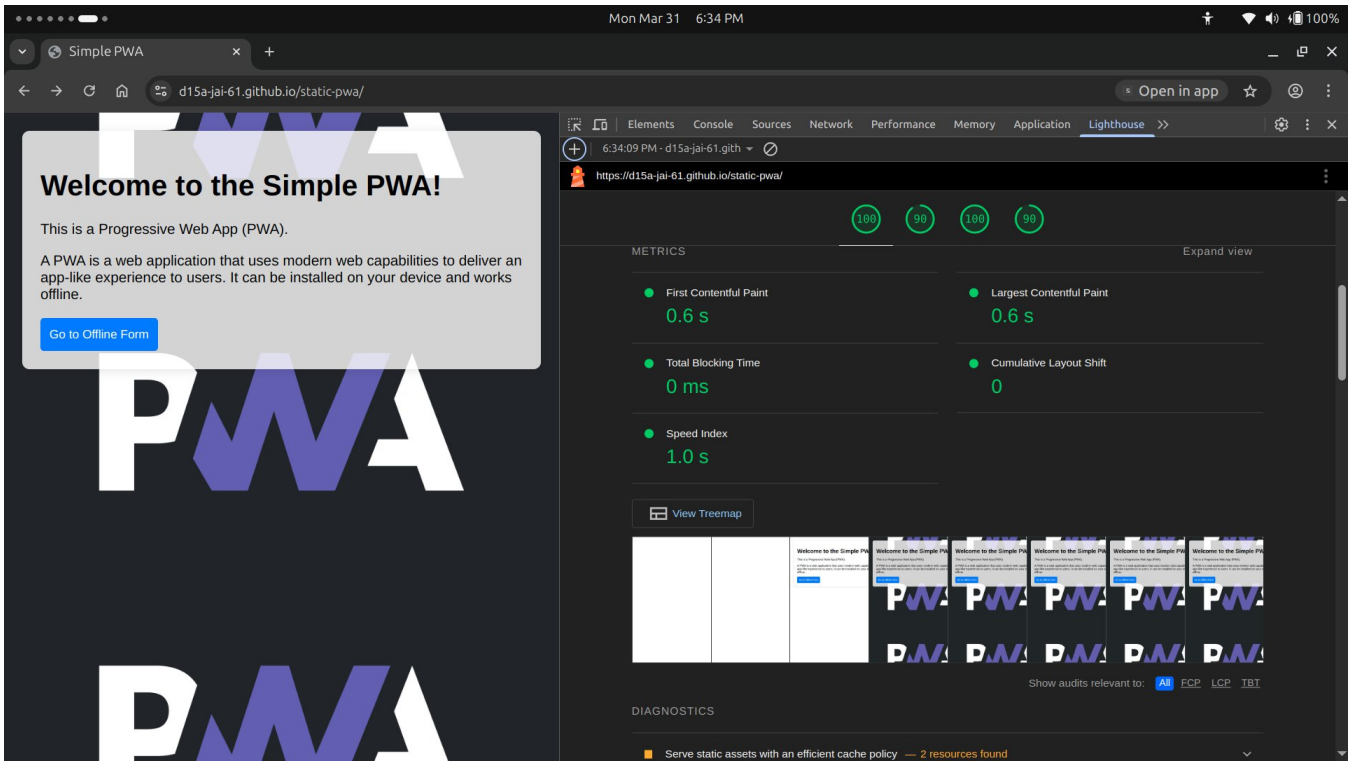
Best Practices: As any developer would know, there are a number of practices that have been deemed ‘best’ based on empirical data. This metric is an aggregation of many such points, including but not limited to:Use of HTTPS

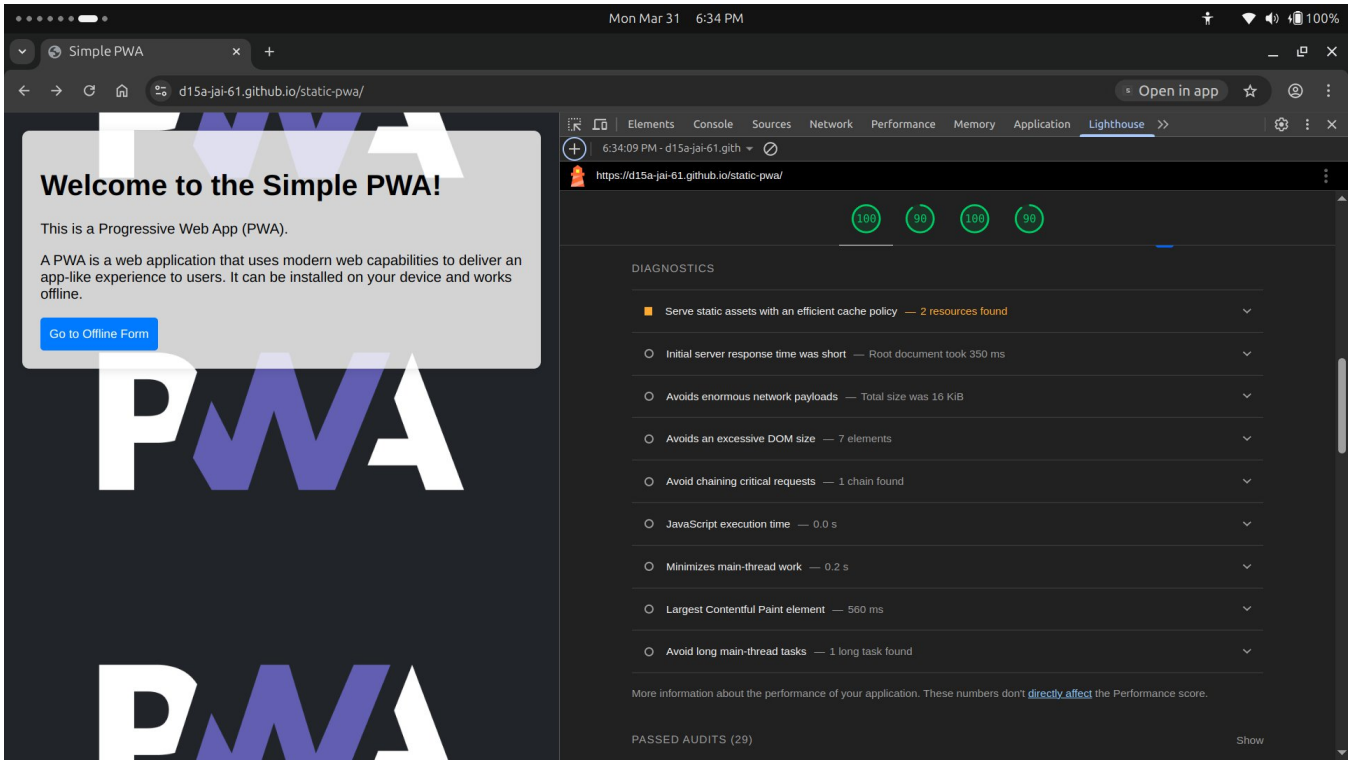
Avoiding the use of deprecated code elements like tags, directives, libraries, etc. Password input with paste-into disabled

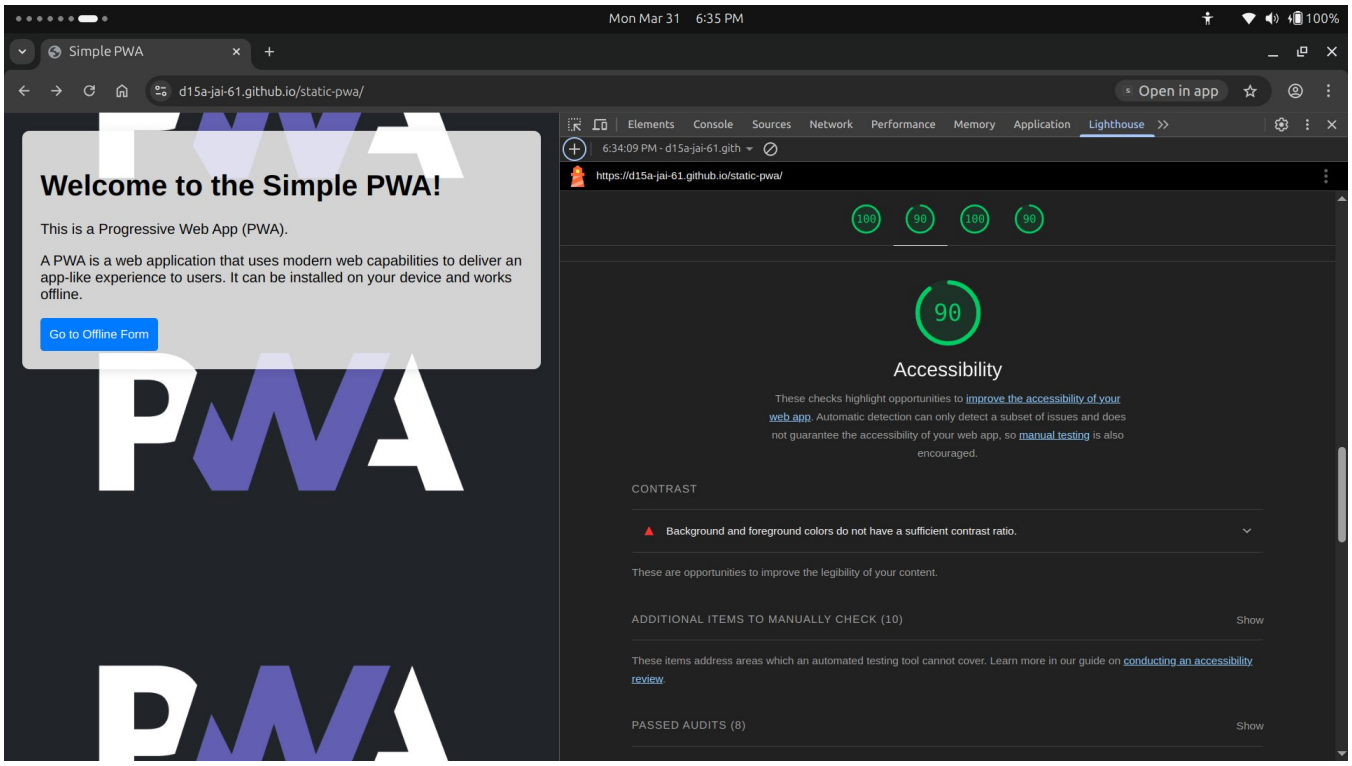
Geo-Location and cookie usage alerts on load, etc.

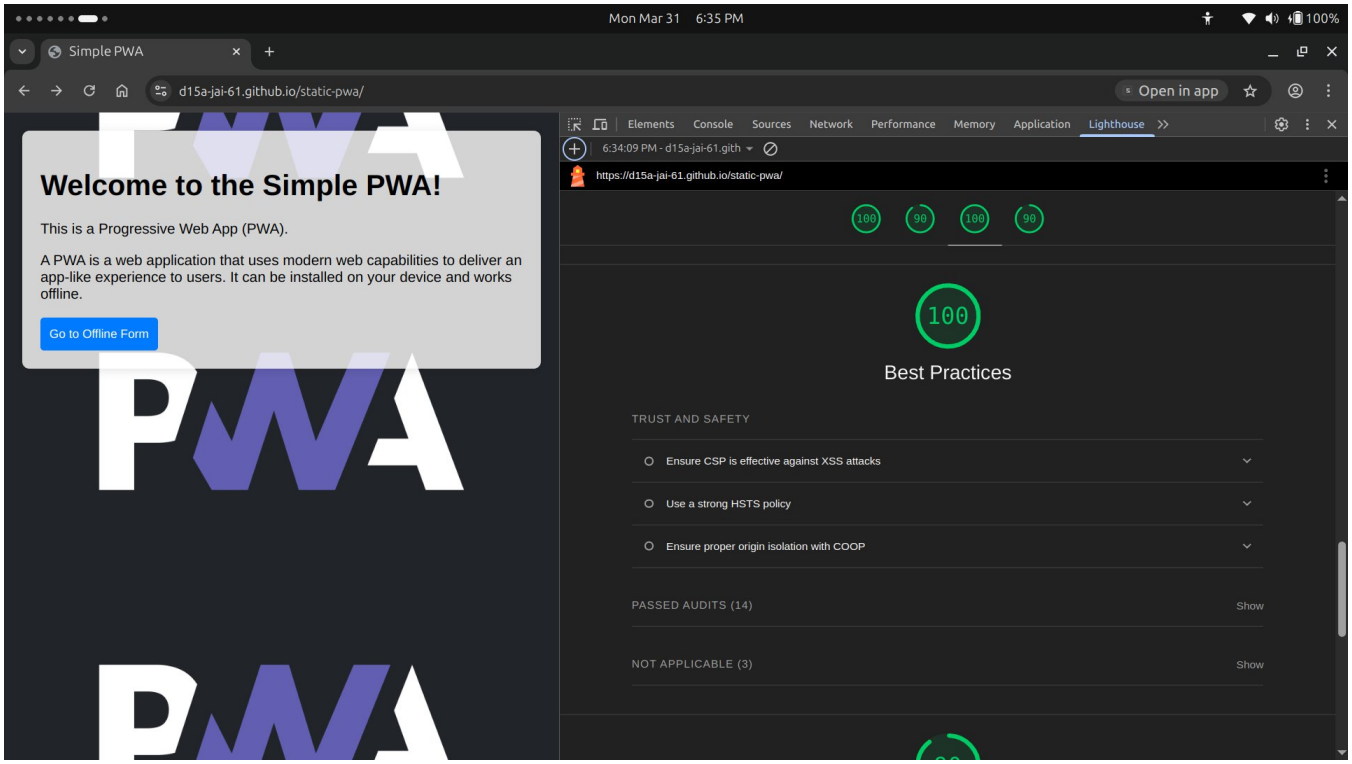

Mon Mar 31 6:33 PM
Simple PWA
d15a-jai-61.github.io/static-pwa/
Open in app
Welcome to the Simple PWA!
This is a Progressive Web App (PWA).
A PWA is a web application that uses modern web capabilities to deliver an app-like experience to users. It can be installed on your device and works offline.
Go to Offline Form
Elements Console Sources Network Performance Memory Application Lighthouse
Generate a Lighthouse report
Analyze page load
Mode Learn more
Navigation (Default)
Timespan
Snapshot
Device
Mobile
Desktop
Categories
Performance
Accessibility
Best practices
SEO

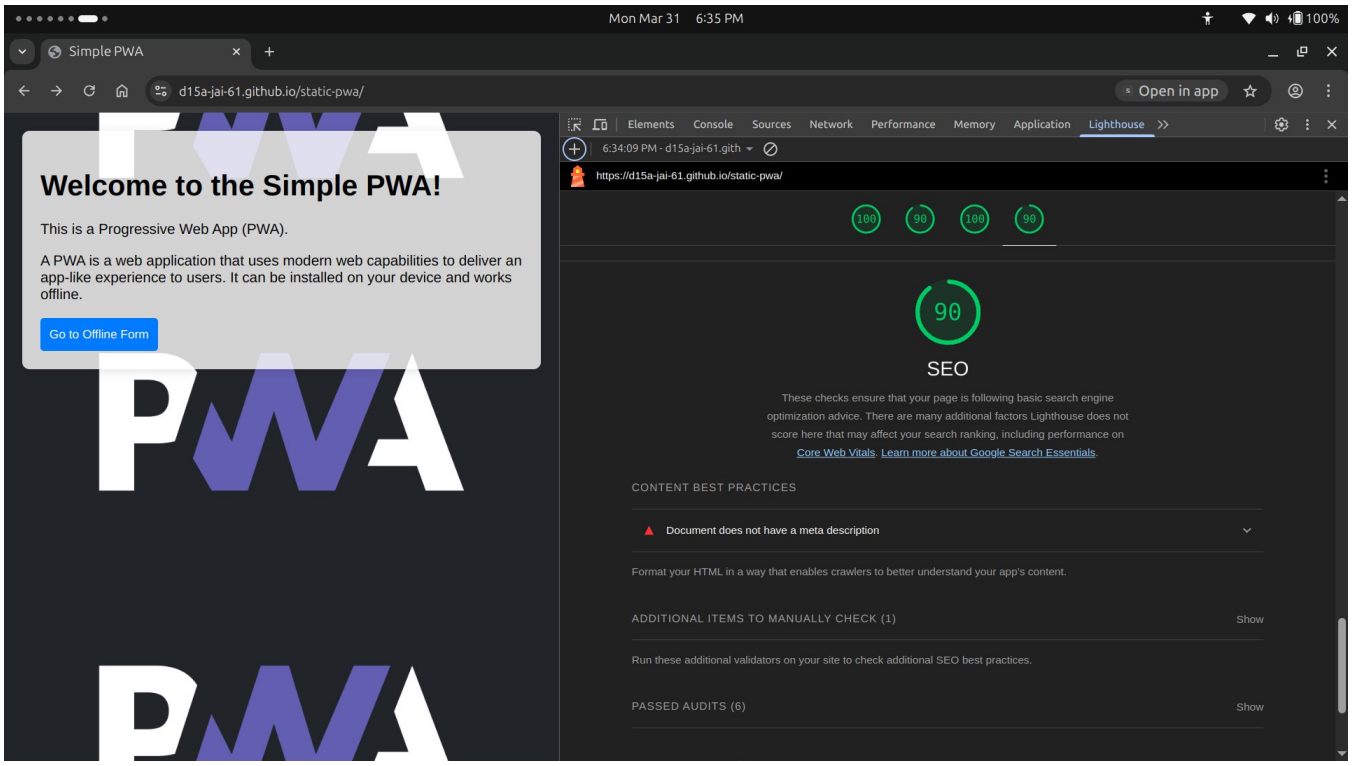

Mon Mar 31 6:34 PM
Simple PWA
d15a-jai-61.github.io/static-pwa/
Open in app
Welcome to the Simple PWA!
This is a Progressive Web App (PWA).
A PWA is a web application that uses modern web capabilities to deliver an app-like experience to users. It can be installed on your device and works offline.
Go to Offline Form
Elements Console Sources Network Performance Memory Application Lighthouse
https://d15a-jai-61.github.io/static-pwa/
100 Performance
90 Accessibility
100 Best Practices
90 SEO
100
Performance
Values are estimated and may vary. The performance score is calculated directly from these metrics. See calculator.
0–49 50–89 90–100
METRICS
Expand view
First Contentful Paint
0.6 s
Largest Contentful Paint
0.6 s


Mon Mar 31 6:34 PM
100%
Simple PWA
d15a-jai-61.github.io/static-pwa/
Open in app
Welcome to the Simple PWA!
This is a Progressive Web App (PWA).
A PWA is a web application that uses modern web capabilities to deliver an app-like experience to users. It can be installed on your device and works offline.
Go to Offline Form
Elements Console Sources Network Performance Memory Application Lighthouse
https://d15a-jai-61.github.io/static-pwa/
100 90 100 90
METRICS
Expand view
First Contentful Paint
0.6 s
Largest Contentful Paint
0.6 s
Total Blocking Time
0 ms
Cumulative Layout Shift
0
Speed Index
1.0 s
View Treemap
Show audits relevant to: All FCP LCP TBT
DIAGNOSTICS
Serve static assets with an efficient cache policy — 2 resources found


Mon Mar 31 6:34 PM
100%
Simple PWA
d15a-jai-61.github.io/static-pwa/
Open in app
Welcome to the Simple PWA!
This is a Progressive Web App (PWA).
A PWA is a web application that uses modern web capabilities to deliver an app-like experience to users. It can be installed on your device and works offline.
Go to Offline Form
Elements Console Sources Network Performance Memory Application Lighthouse
https://d15a-jai-61.github.io/static-pwa/
100 90 100 90
DIAGNOSTICS
Serve static assets with an efficient cache policy — 2 resources found
Initial server response time was short — Root document took 350 ms
Avoids enormous network payloads — Total size was 16 KiB
Avoids an excessive DOM size — 7 elements
Avoid chaining critical requests — 1 chain found
JavaScript execution time — 0.0 s
Minimizes main-thread work — 0.2 s
Largest Contentful Paint element — 560 ms
Avoid long main-thread tasks — 1 long task found
More information about the performance of your application. These numbers don't directly affect the Performance score.
PASSED AUDITS (29)
Show


Mon Mar 31 6:35 PM
Simple PWA
d15a-jai-61.github.io/static-pwa/
Open in app
Welcome to the Simple PWA!
This is a Progressive Web App (PWA).
A PWA is a web application that uses modern web capabilities to deliver an app-like experience to users. It can be installed on your device and works offline.
Go to Offline Form
Elements Console Sources Network Performance Memory Application Lighthouse
https://d15a-jai-61.github.io/static-pwa/
100 90 100 90
90
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.
CONTRAST
Background and foreground colors do not have a sufficient contrast ratio.
These are opportunities to improve the legibility of your content.
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 (8)
Show


Mon Mar 31 6:35 PM
Simple PWA
d15a-jai-61.github.io/static-pwa/
Open in app
Welcome to the Simple PWA!
This is a Progressive Web App (PWA).
A PWA is a web application that uses modern web capabilities to deliver an app-like experience to users. It can be installed on your device and works offline.
Go to Offline Form
Elements Console Sources Network Performance Memory Application Lighthouse
https://d15a-jai-61.github.io/static-pwa/
100 90 100 90
100
Best Practices
TRUST AND SAFETY
Ensure CSP is effective against XSS attacks
Use a strong HSTS policy
Ensure proper origin isolation with COOP
PASSED AUDITS (14)
Show
NOT APPLICABLE (3)
Show



Conclusion: Thus we successfully used google Lighthouse PWA Analysis Tool for testing the PWA functioning.