Experiment- 11

AIM: To use Google Lighthouse PWA Analysis Tool to test the PWA functioning

THEORY:

Lighthouse

Lighthouse is an open-source, automated tool for improving the quality of web pages. You can run it against any web page, public or requiring authentication. It has audits for performance, accessibility, progressive web apps, SEO and more.

You can run Lighthouse in Chrome DevTools, from the command line, or as a Node module. You give Lighthouse a URL to audit, it runs a series of audits against the page, and then it generates a report on how well the page did. From there, use the failing audits as indicators on how to improve the page. Each audit has a reference doc explaining why the audit is important, as well as how to fix it. You can also use Lighthouse CI to prevent regressions on your sites.

Lighthouse workflow:

- In Chrome DevTools: Easily audit pages that require authentication, and read your reports in a user-friendly format.
- From the command line: Automate your Lighthouse runs via shell scripts.
- As a Node module: Integrate Lighthouse into your continuous integration systems.
- From a web UI: Run Lighthouse and link to reports without installing a thing.

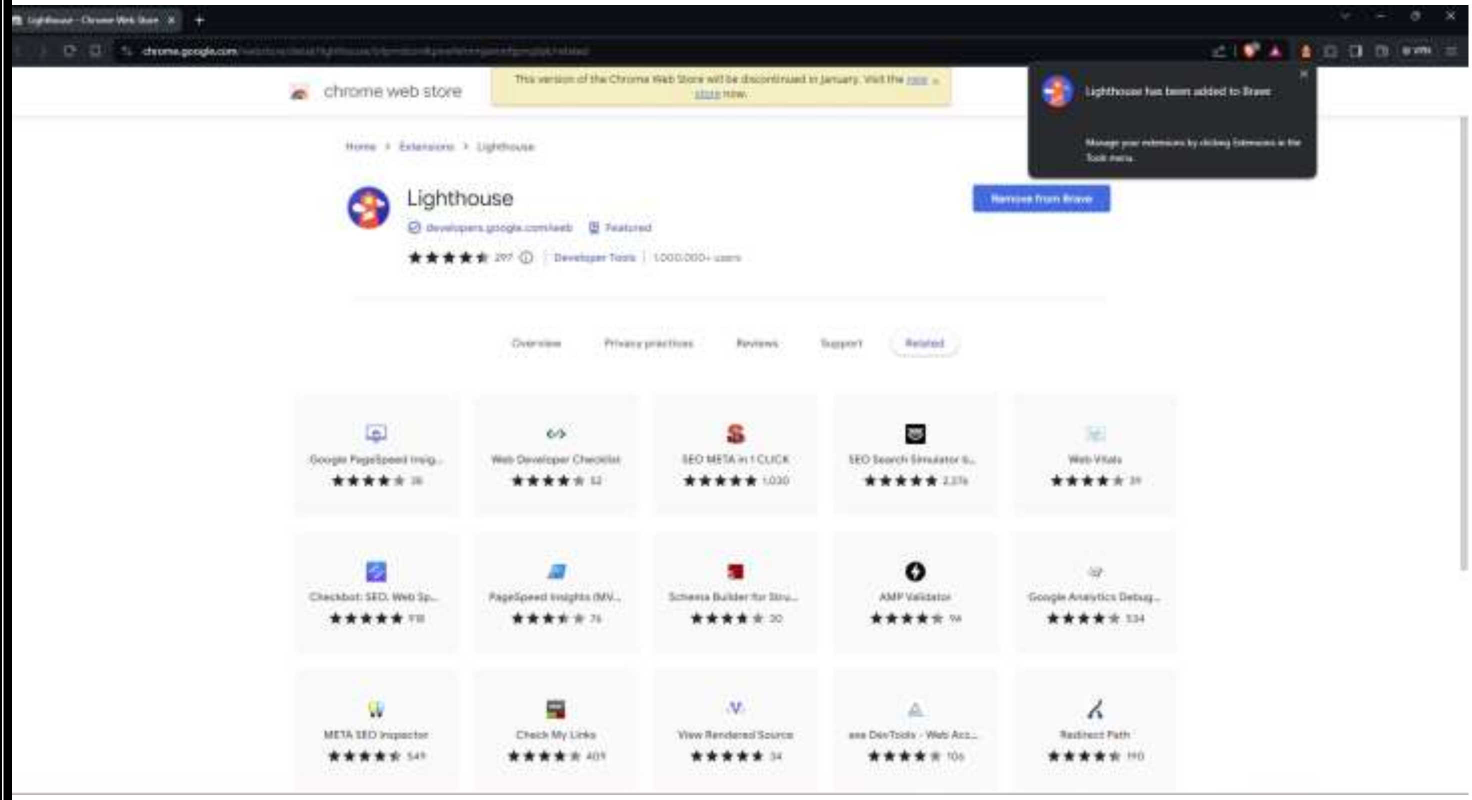
Note: The CLI and Node workflows require you to have an instance of Google Chrome installed on your machine.

To install the extension:

1. Download Google Chrome for Desktop.
2. Install the Lighthouse Chrome Extension from the Chrome Webstore.

To run an audit:

1. In Chrome, go to the page you want to audit.
2. Click Lighthouse Lighthouse. It should be next to the Chrome address bar. If not, open Chrome's main menu and access it at the top of the menu. After clicking, the Lighthouse menu expands.



chrome web store

This version of the Chrome Web Store will be discontinued in January. Visit the new store now.

Lighthouse has been added to Brave

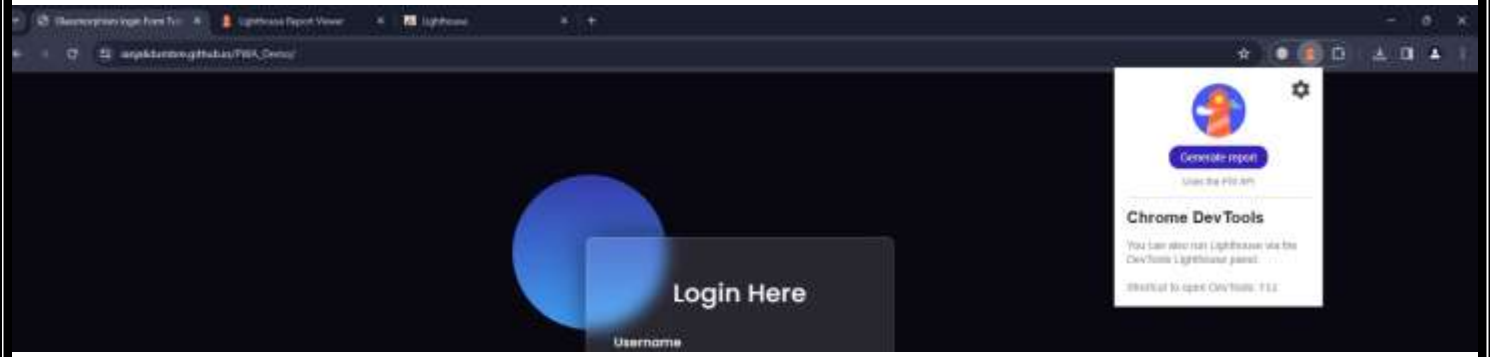
Manage your extensions by clicking Extensions in the Tools menu.

Home > Extensions > Lighthouse

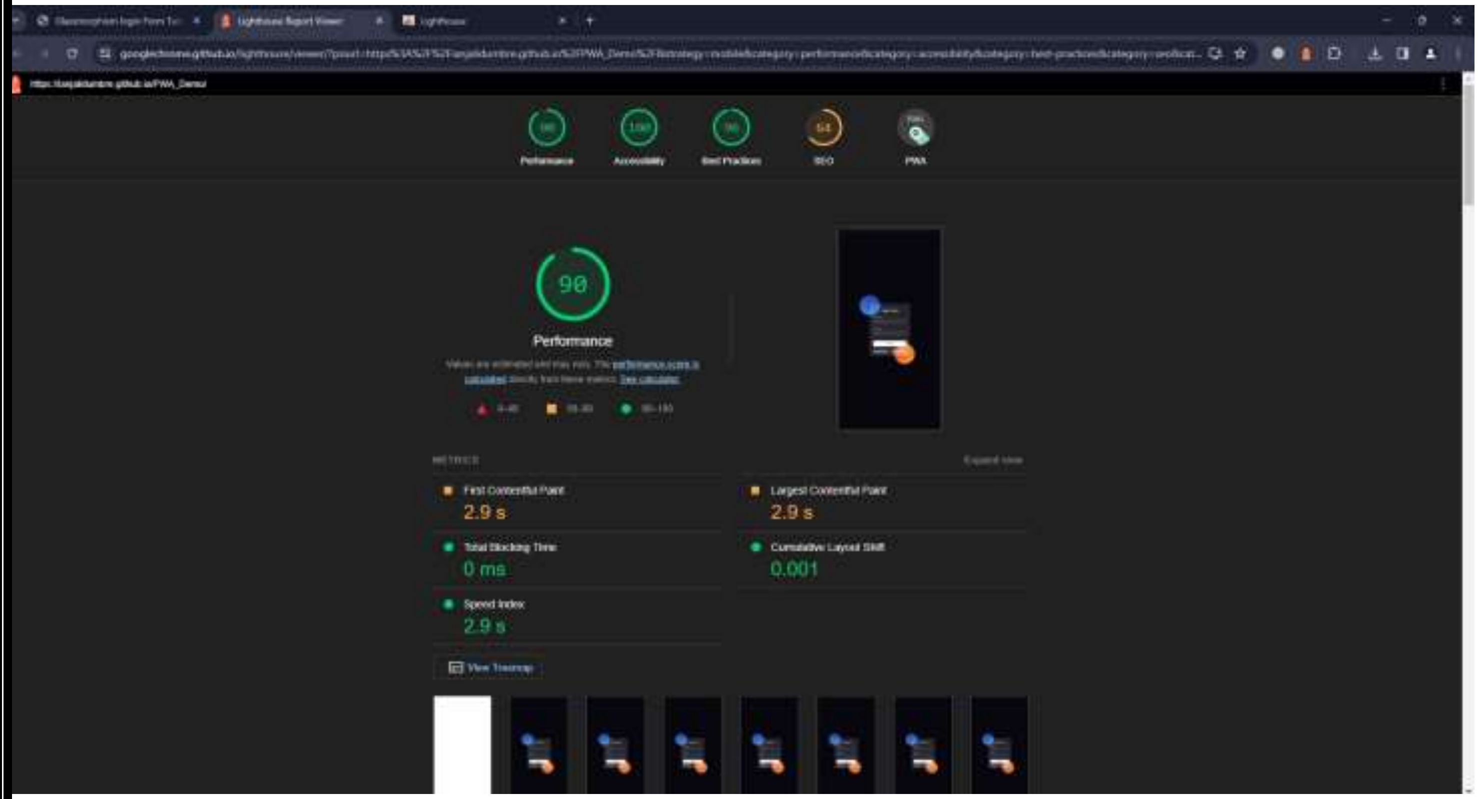
Lighthouse

developers.google.com/web Featured

297 | Developer Tools | 1,000,000+ users



You can also run Lighthouse via the DevTools Lighthouse panel.



Performance Accessibility Best Practices SEO PWA

90

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
2.9 s

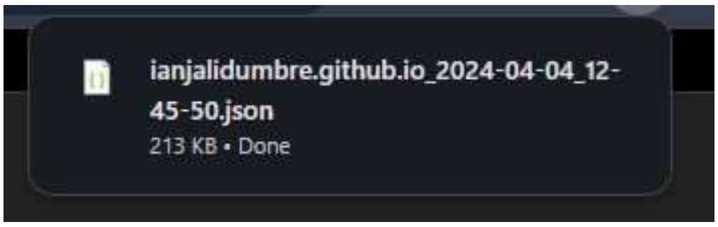
Largest Contentful Paint
2.9 s

Total Blocking Time
0 ms

Cumulative Layout Shift
0.001

Speed Index
2.9 s

View Treemap



ianjalidumbre.github.io_2024-04-04_12-45-50.json
213 KB • Done

Run PageSpeed Insights

To run Lighthouse on PageSpeed Insights:

1. Navigate to PageSpeed Insights.

2. Enter a web page URL.
3. Click **Analyze**.

Share reports as JSON

The Lighthouse Viewer needs the JSON output of a Lighthouse report. The list below explains how to get the JSON output, depending on what Lighthouse workflow you're using:

1. **Chrome DevTools**. Click **Download Report**

2. **Command line**. Run:

   ```
   lighthouse --output json --output-path <path/for/output.json>
   ```

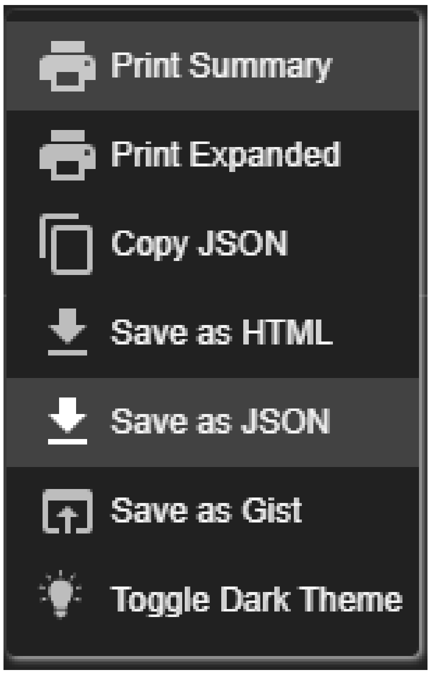
3. **Lighthouse Viewer**. Click **Export** > **Save as JSON**.

4. **Click on ⋮ and save as Json**

To view the report data:

1. Open the Lighthouse Viewer in Google Chrome.

2. Drag the JSON file onto the viewer, or click anywhere on the Viewer to open your file navigator and select the file.



Share reports as GitHub Gists

If you don't want to manually pass around JSON files, you can also share your reports as secret GitHub Gists. One benefit of Gists is free version control.

To export a report as a Gist from the report:

1. (If already on the viewer, skip this step) Click Export > Open In Viewer. The report opens in the Viewer, located at https://googlechrome.github.io/lighthouse/viewer/.

2. In the Viewer, click Share ⋮. The first time you do this, a popup asks permission to access your basic GitHub data, and to read and write to your Gits.

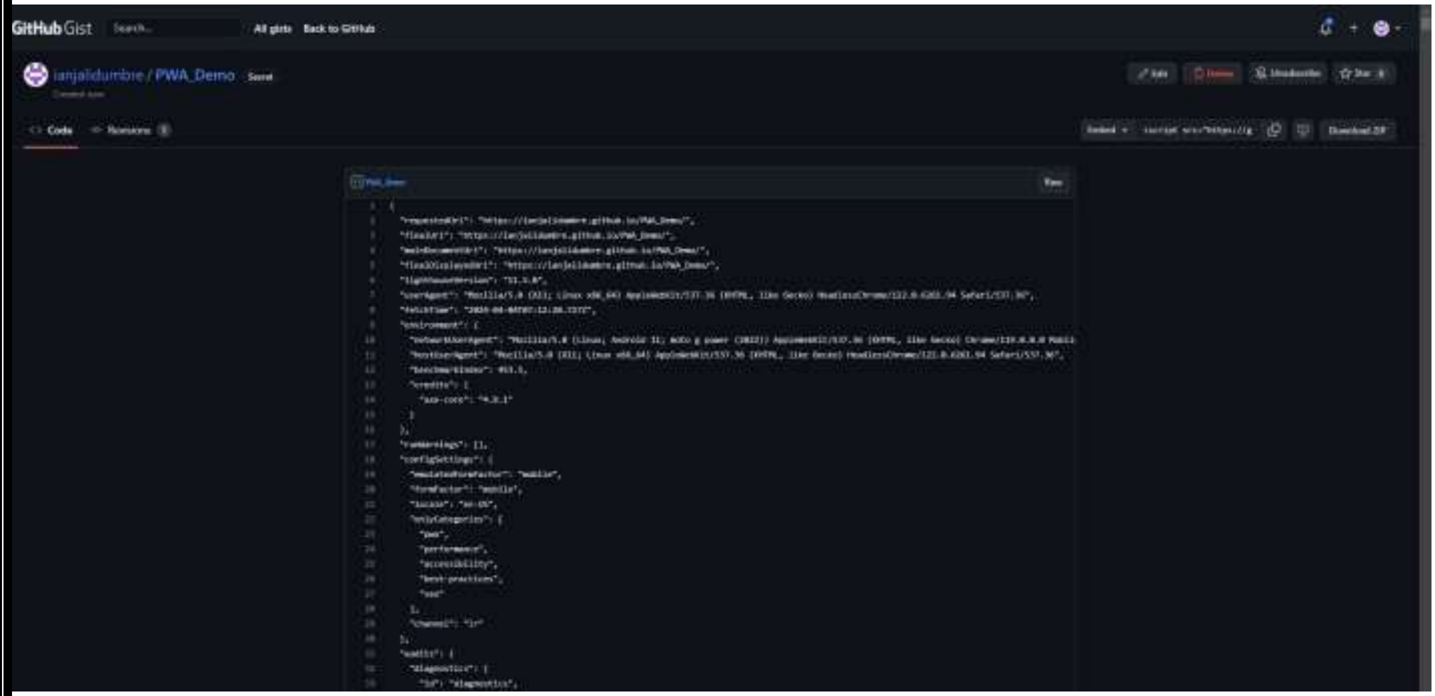
3. To export a report as a Gist from the CLI version of Lighthouse, just manually create a Gist and copy-paste the report's JSON output into the Gist. The Gist filename containing the JSON output must end in .lighthouse, report, Json. See Share reports as JSON for an example of how to generate.

JSON output from the command line tool.

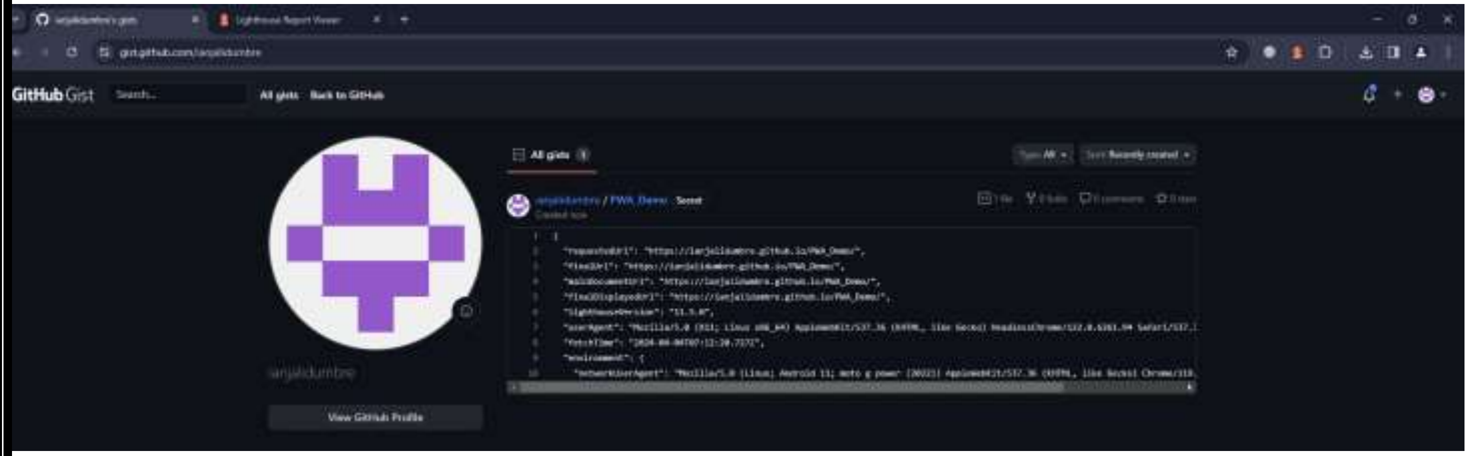
To view a report that's been saved as a Gist:

1. Add ?gist=<ID> to the Viewer's URL, where <ID> is the ID of the Gist.
2. https://googlechrome.github.io/lighthouse/viewer/?gist=<ID>
3. Open the Viewer, and paste the URL of a Gist into it.

4. **Click on ⋮ and save as Gists**
5. **Open your GitHub account and go to Your Gists to find the gists url**



```
{
  "requestedUrl": "https://ianjalidumbre.github.io/PWA_Demo/",
  "finalUrl": "https://ianjalidumbre.github.io/PWA_Demo/",
  "mainDocumentUrl": "https://ianjalidumbre.github.io/PWA_Demo/",
  "finalDisplayedUrl": "https://ianjalidumbre.github.io/PWA_Demo/",
  "lighthouseVersion": "11.5.0",
  "userAgent": "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) HeadlessChrome/122.0.6261.94 Safari/537.36",
  "fetchTime": "2024-04-04T07:12:20.737Z",
  "environment": {
    "networkUserAgent": "Mozilla/5.0 (Linux; Android 11; moto g power (2022)) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/119.0.0.0 Mobil
    "hostUserAgent": "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) HeadlessChrome/122.0.6261.94 Safari/537.36",
    "benchmarkIndex": 493.5,
    "credits": {
      "axe-core": "4.8.1"
    }
  },
  "runWarnings": [],
  "configSettings": {
    "emulatedFormFactor": "mobile",
    "formFactor": "mobile",
    "locale": "en-US",
    "onlyCategories": [
      "pwa",
      "performance",
      "accessibility",
      "best-practices",
      "seo"
    ],
    "channel": "cli"
  },
  "audits": {
    "diagnostics": {
      "id": "diagnostics",
```



```
{
  "requestedUrl": "https://ianjalidumbre.github.io/PWA_Demo/",
  "finalUrl": "https://ianjalidumbre.github.io/PWA_Demo/",
  "mainDocumentUrl": "https://ianjalidumbre.github.io/PWA_Demo/",
  "finalDisplayedUrl": "https://ianjalidumbre.github.io/PWA_Demo/",
  "lighthouseVersion": "11.5.0",
  "userAgent": "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) HeadlessChrome/122.0.6261.94 Safari/537.3
  "fetchTime": "2024-04-04T07:12:20.737Z",
  "environment": {
    "networkUserAgent": "Mozilla/5.0 (Linux; Android 11; moto g power (2022)) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/119.
```

Conclusion:-
Hence we have used Google Lighthouse PWA Analysis Tool to test the PWA functioning