

120A3051

Shreya Idate

Batch: E3

Experiment No: 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:

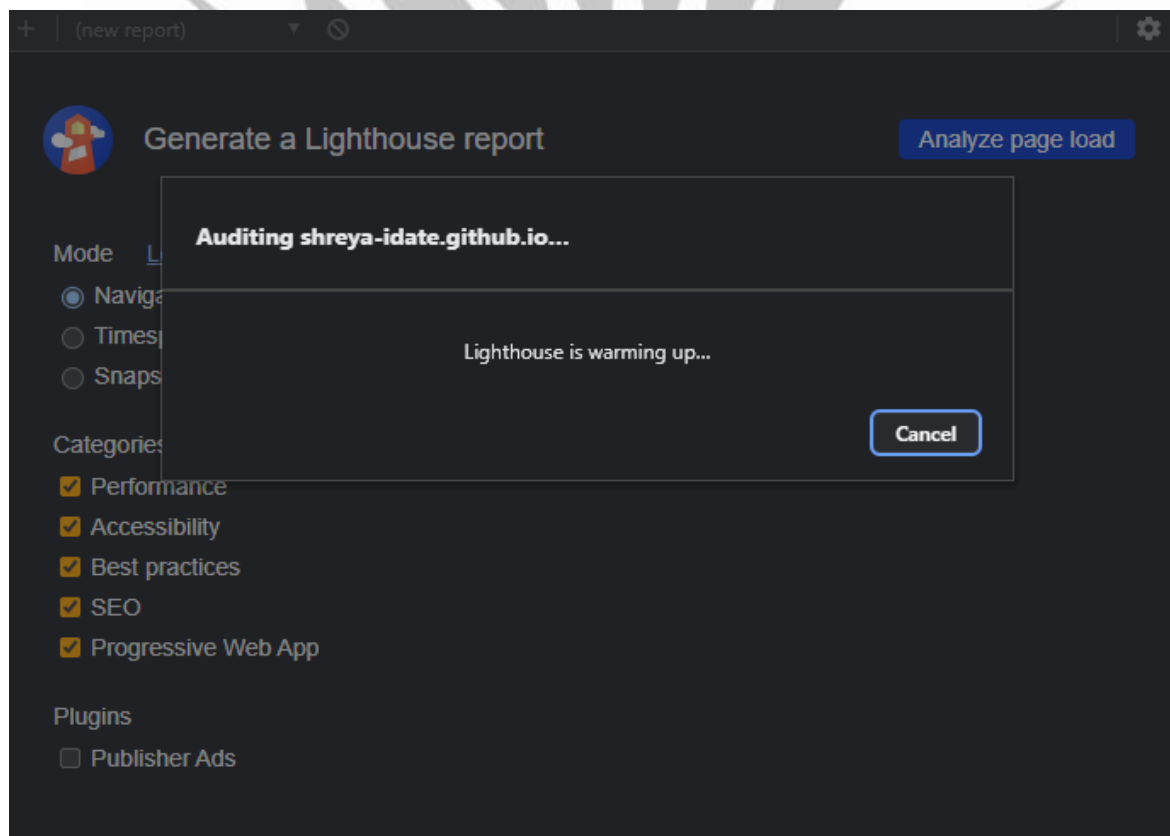
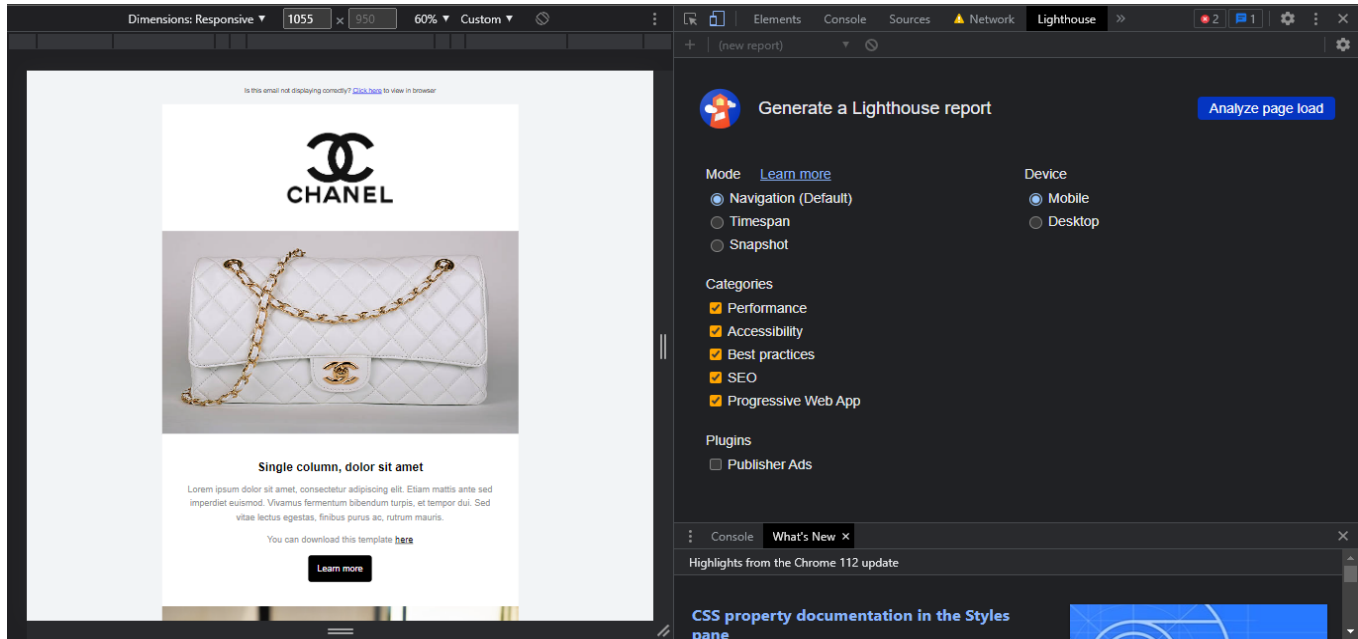
- In Chrome DevTools: Easily audit pages that require authentication, and read your reports in a user-friendly format.
- From the command line: Automate your Lighthouse runs via shell scripts.
- As a Node module: Integrate Lighthouse into your continuous integration systems.
- From a web UI: Run Lighthouse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

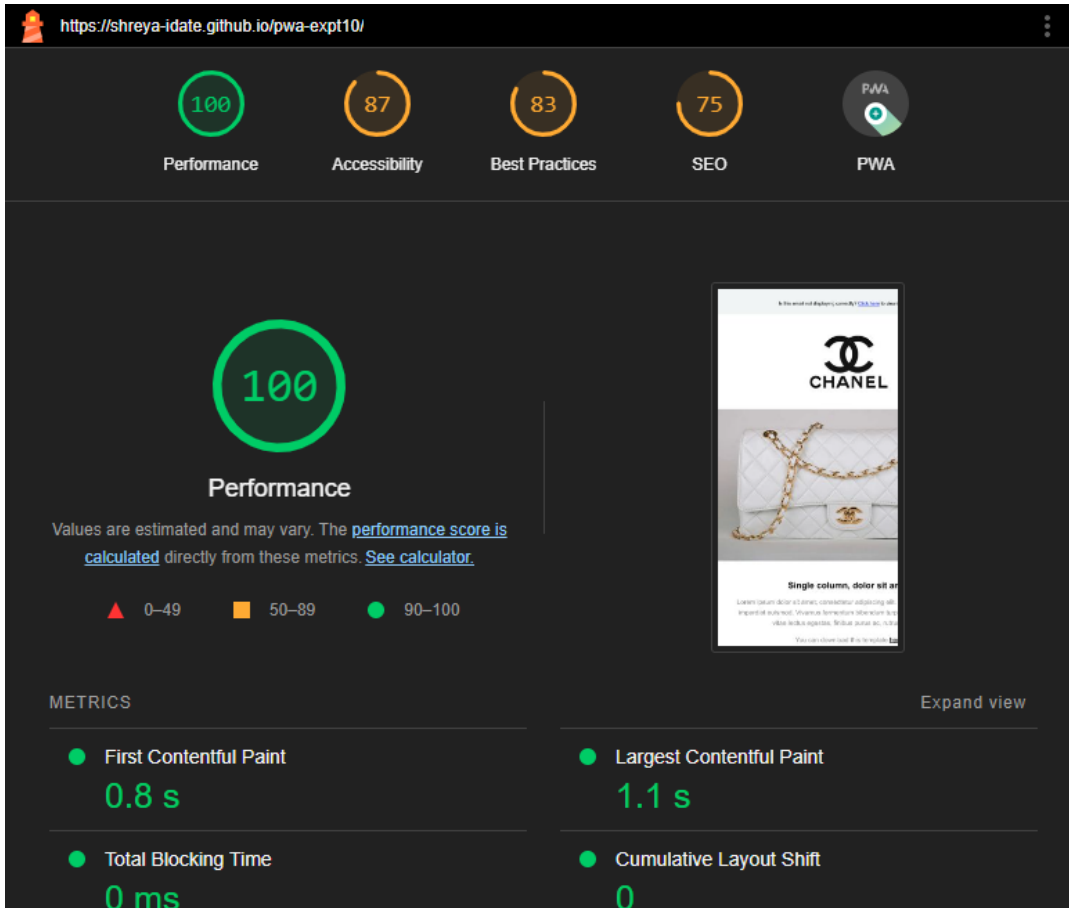
Note: The CLI and Node workflows require you to have an instance of Google Chrome installed on your machine.

Run Lighthouse in Chrome DevTools

Lighthouse powers the Audits panel of Chrome DevTools. To run a report:

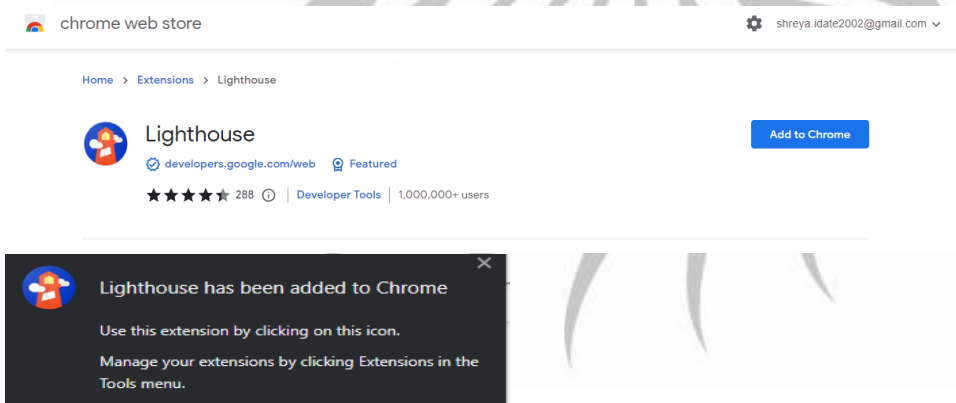
1. Download **Google Chrome for Desktop**.
2. In Google Chrome, go to the URL you want to audit. You can audit any URL on the web.
3. **Open Chrome DevTools**.
4. Click the **Lighthouse** tab.
5. Click on **Generate Report** : Lighthouse runs its audits against the currently-focused page, then opens up a new tab with a report of the results.





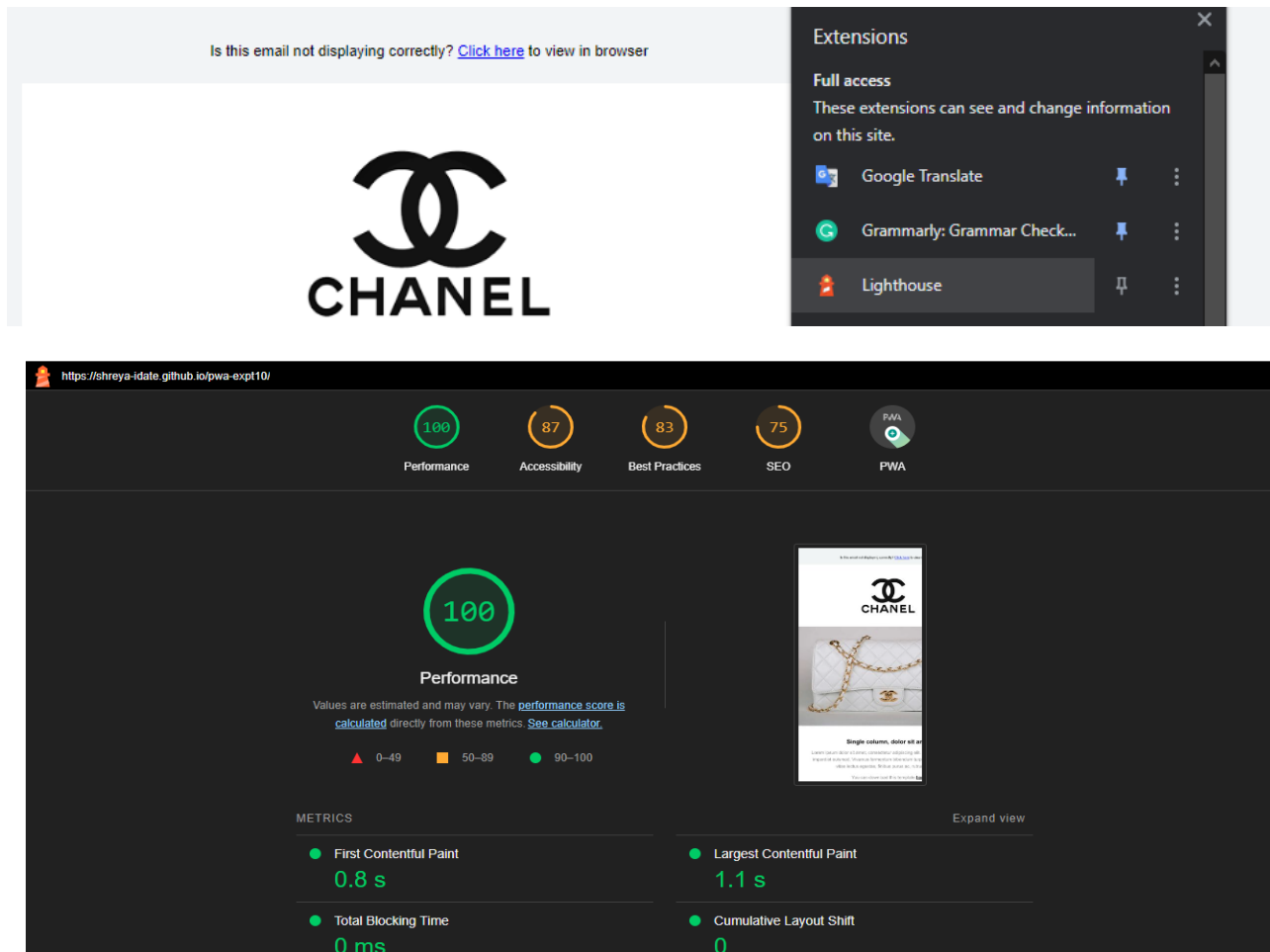
To install the extension:

1. Download Google Chrome for Desktop.
2. Install the [Lighthouse Chrome Extension](#) from the Chrome Webstore.



To run an audit:

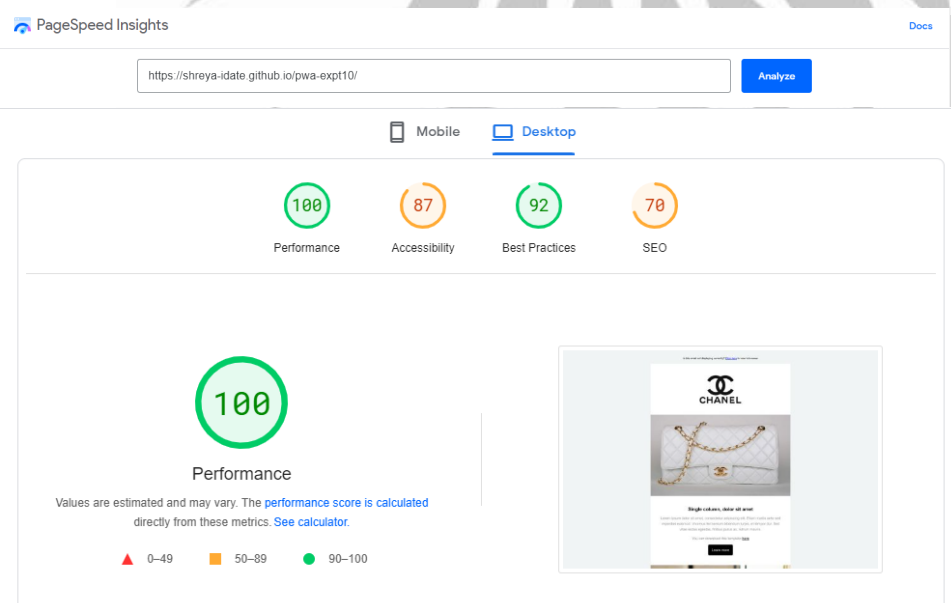
1. In Chrome, go to the page you want to audit.
2. Click Lighthouse. It should be next to the Chrome address bar. If not, open Chrome's main menu and access it at the top of the menu. After clicking, the Lighthouse menu expands.

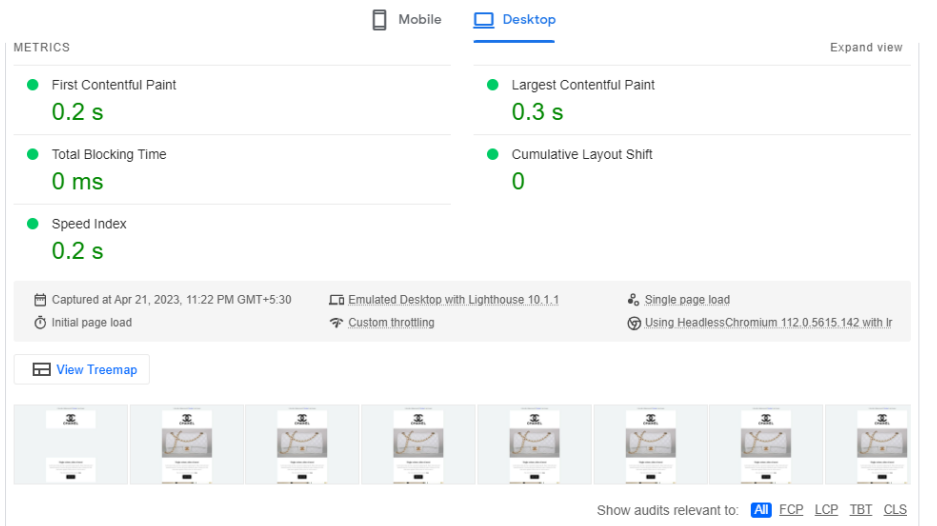


Run PageSpeed Insights

To run Lighthouse on PageSpeed Insights:

1. Navigate to [PageSpeed Insights](#).
2. Enter a web page URL.
Click **Analyze**.






Share reports as JSON

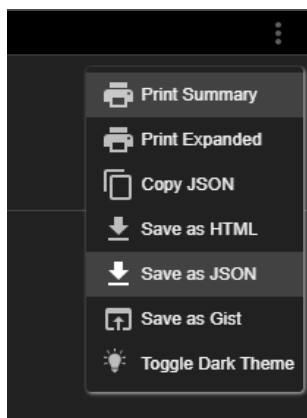
The Lighthouse Viewer needs the JSON output of a Lighthouse report. The list below explains how to get the JSON output, depending on what Lighthouse workflow you're using:

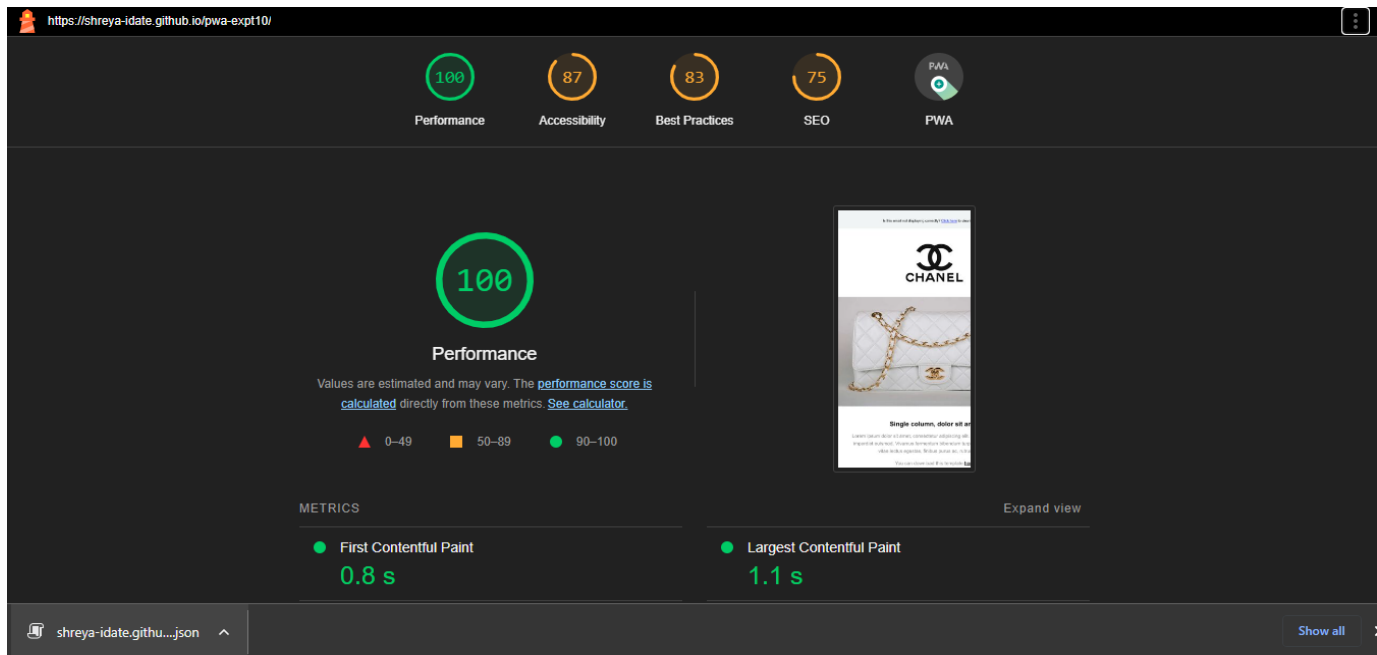
1. **Chrome DevTools.** Click **Download Report** ↓
2. **Command line.** Run:

```
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 GitHubGists

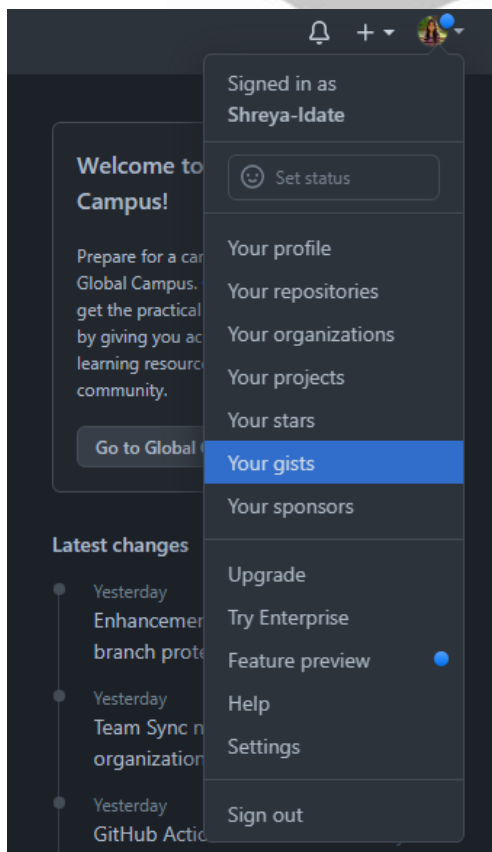
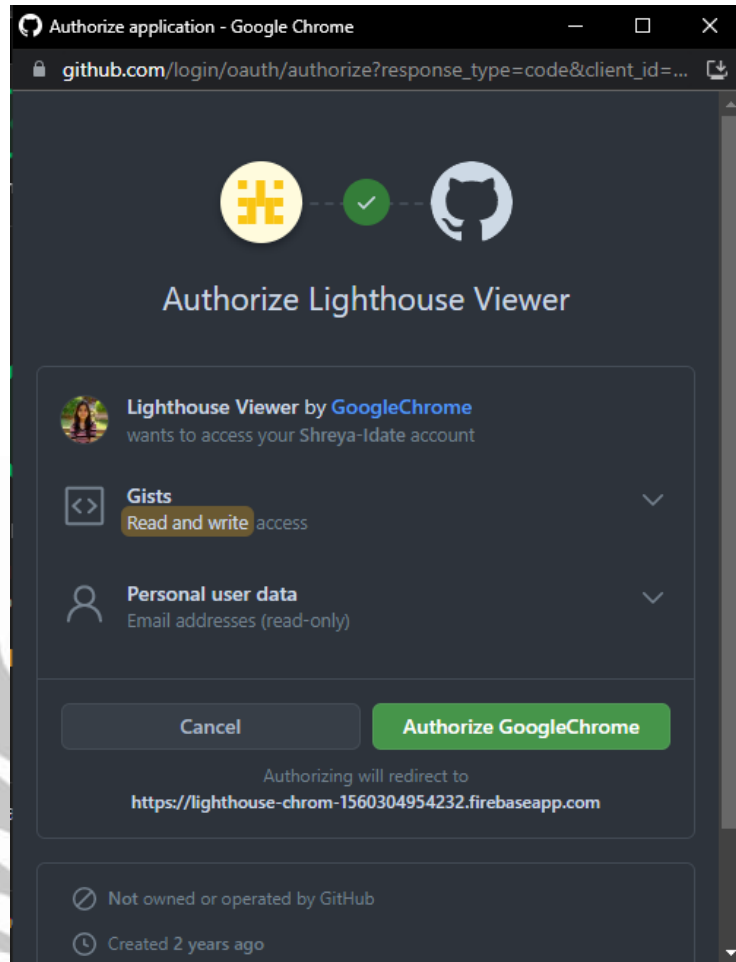
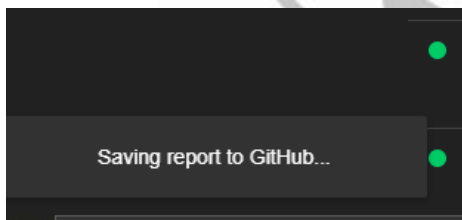
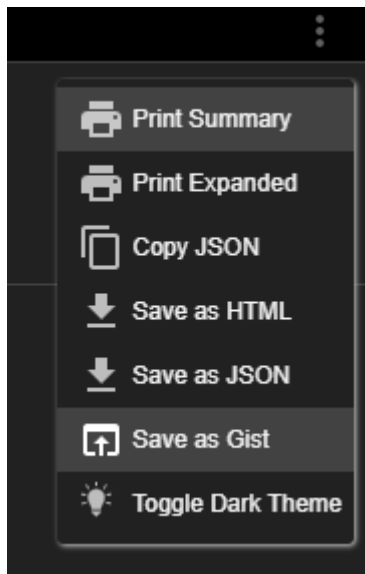
If you don't want to manually pass around JSON files, you can also share your reports as secret GitHubGists. One benefit of Gists is free version control.

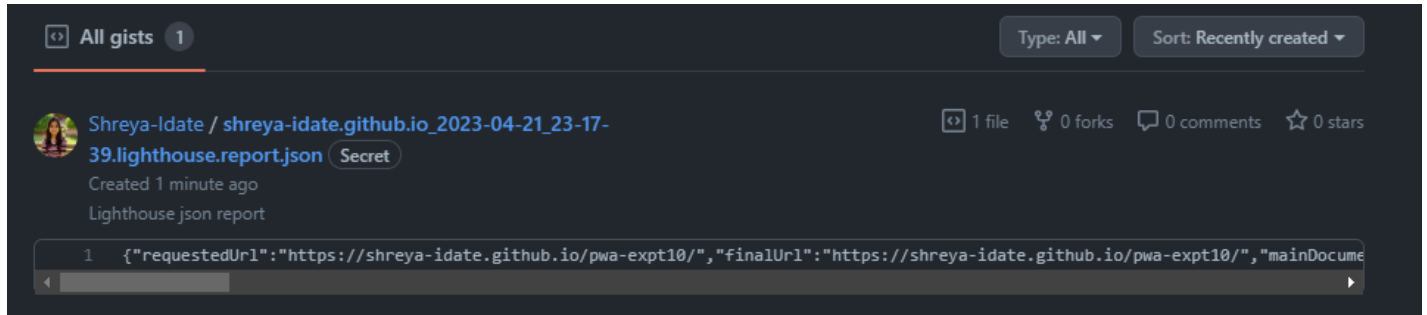
To export a report as a Gist from the report:

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 Gists.
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 gistsurl





The screenshot shows a GitHub Gist interface. At the top, it says 'All gists 1'. There are filters for 'Type: All' and 'Sort: Recently created'. The gist is titled 'Shreya-Idate / shreya-idate.github.io_2023-04-21_23-17-39.lighthouse.report.json' and is marked as 'Secret'. It was created '1 minute ago' and is a 'Lighthouse json report'. The gist contains 1 file. The code snippet shown is a JSON object:

```
1  {"requestedUrl":"https://shreya-idate.github.io/pwa-expt10/","finalUrl":"https://shreya-idate.github.io/pwa-expt10/","mainDocume
```

Conclusion: - We have successfully learnt how to use Google Lighthouse PWA Analysis Tool to test the PWA functioning.

