Experiment: Using Google Lighthouse for PWA Analysis

Objective:
To use Google Lighthouse PWA Analysis Tool to test the Progressive Web App (PWA) functionality.

Theory:
Google Lighthouse is an open-source automated tool used to audit and analyze web applications for performance, accessibility, best practices, SEO, and Progressive Web App (PWA) capabilities. Lighthouse evaluates whether a web app meets PWA standards, such as:

- Service worker registration for offline support
- Web App Manifest inclusion
- HTTPS security compliance
- Fast and responsive performance
- Installability and app-like user experience

Lighthouse assigns a score based on these criteria, helping developers optimize their PWA.

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:

1. **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.
2. **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.
3. **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.
4. **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.

Requirements:

- Google Chrome browser
- A PWA-enabled web application
- Chrome DevTools or Lighthouse CLI

Procedure:

1. Open Lighthouse in Chrome DevTools

- Launch the PWA in Google Chrome.
- Open DevTools using `F12` or `Ctrl+Shift+I`.
- Navigate to the **Lighthouse** tab.

2. Configure Lighthouse Audit

- Select the **Progressive Web App** category.
- Ensure other relevant categories like Performance and Best Practices are checked.
- Choose the mode:

    - **Mobile** (default) for testing mobile performance.
    - **Desktop** for desktop evaluation.
- Click **Generate Report**.

3. Analyze the Results

- Lighthouse generates a PWA score based on:
    - **Fast and reliable:** Checks service worker caching.
    - **Installable:** Verifies manifest and service worker.
    - **PWA Optimizations:** Ensures proper web app experience.
- Click on individual sections to view recommendations for improvements.

4. Running Lighthouse via CLI (Optional)

Install Lighthouse CLI using Node.js:

```
npm install -g lighthouse
```

-

Run an audit on a PWA URL:

```
lighthouse https://your-pwa-url.com --view
```

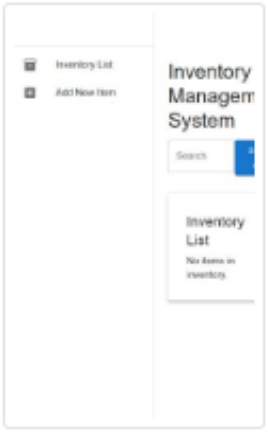
-

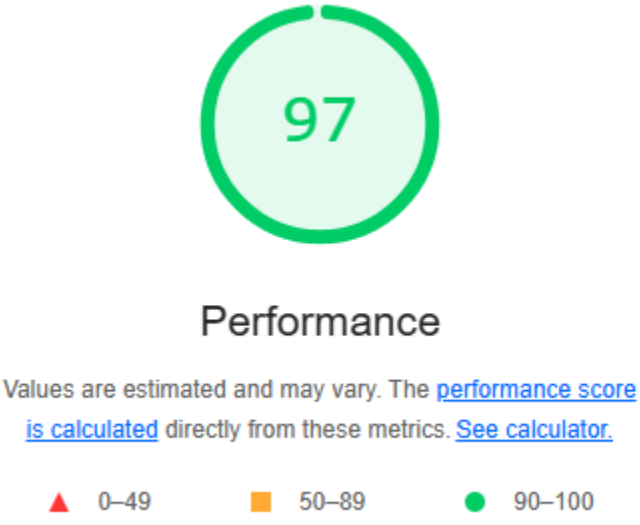
Observations:

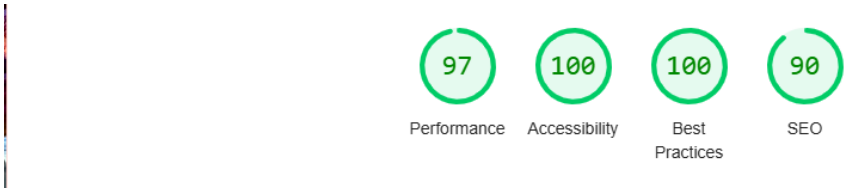
- The PWA is analyzed based on Lighthouse criteria.
- Identified areas needing improvement (e.g., caching strategies, manifest completeness, accessibility fixes).
- Scores indicate overall PWA compliance and optimization level.

Conclusion:
By using Google Lighthouse, we effectively evaluated the PWA's functionality, installability, and performance, enabling improvements for a better user experience.

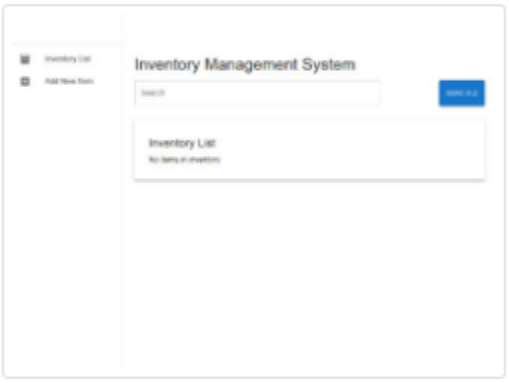
Mobile View:

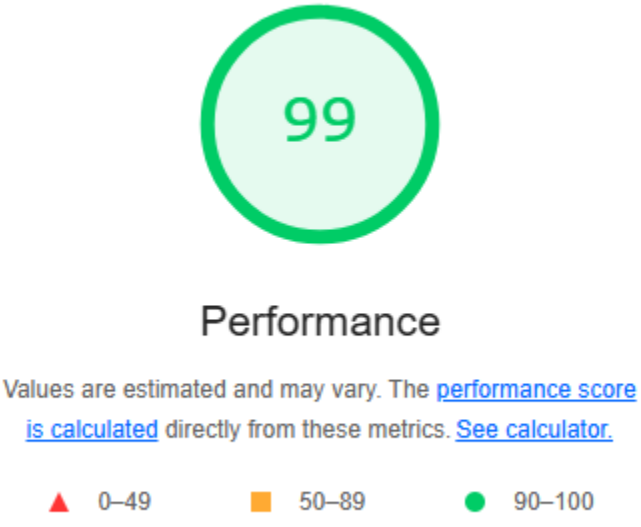


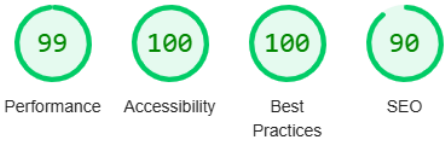




Desktop View:






99
Performance
100
Accessibility
100
Best Practices
90
SEO