

## **Experiment 11**

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

### **Theory:**

A Progressive Web App (PWA) is a web application that uses modern web capabilities to deliver an app-like experience to users. PWAs are:

- Reliable – Load instantly and work offline.
- Fast – Respond quickly to user interactions.
- Engaging – Feel like a native app with features like push notifications and home screen installation.

What is Google Lighthouse?

Google Lighthouse is an open-source, automated auditing tool developed by Google that evaluates web applications based on multiple parameters, including:

- Performance (loading speed, interactivity)
- Progressive Web App (PWA) compliance
- Accessibility (screen-reader compatibility)
- Best Practices (HTTPS, deprecated APIs)
- SEO (search engine optimization)
- Lighthouse generates a detailed report with scores (0–100) and actionable recommendations.

### **Key Features and Audit Metrics**

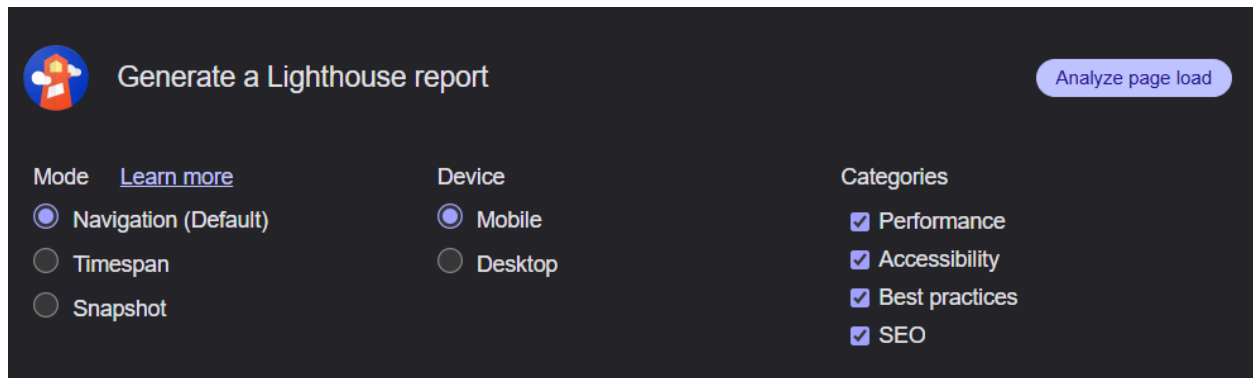
Google Lighthouse has the option of running the Audit for Desktop as well as mobile version of your page(s). The top metrics that will be measured in the Audit are:

- 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.

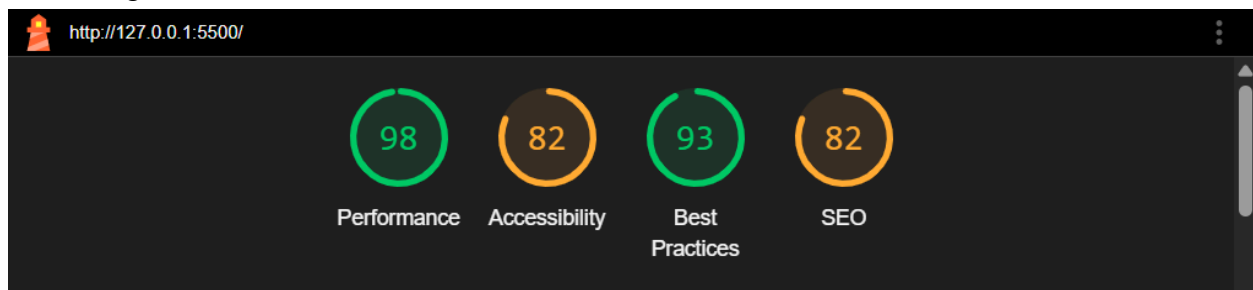
### Steps:

1. Go to developer tools -> Lighthouse



Analyze page load.

2. Lighthouse scores

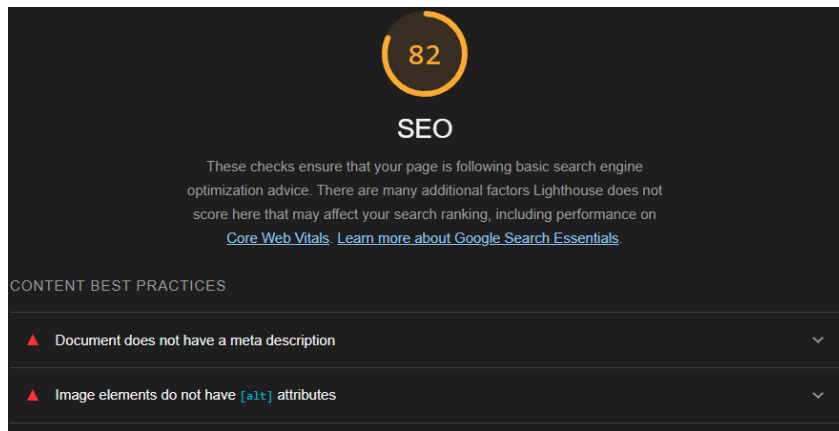
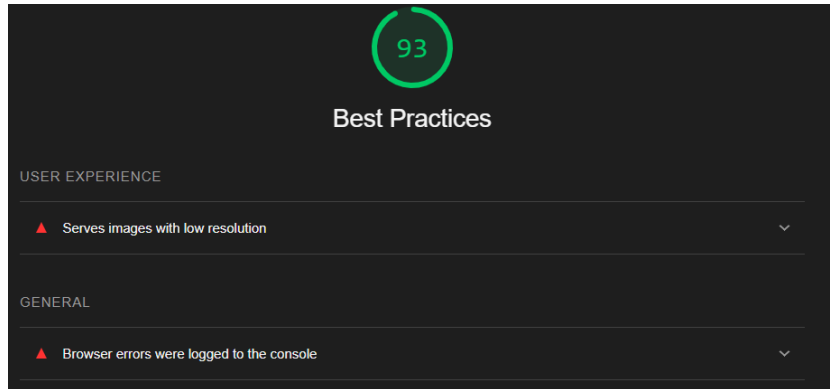


To check any particular score, click on it, it will show you the modifications that should be made.

## Code Modifications:

DIAGNOSTICS		
▲	Eliminate render-blocking resources — Potential savings of 490 ms	▼
▲	Page prevented back/forward cache restoration — 1 failure reason	▼
■	Image elements do not have explicit <code>width</code> and <code>height</code>	▼
■	Enable text compression — Potential savings of 7 KiB	▼
■	Reduce unused JavaScript — Potential savings of 21 KiB	▼
○	Avoid large layout shifts — 3 layout shifts found	▼
○	Avoid chaining critical requests — 9 chains found	▼
○	Minimize third-party usage — Third-party code blocked the main thread for 0 ms	▼
○	Largest Contentful Paint element — 2,070 ms	▼

<div><div>82</div><h2>Accessibility</h2><p>These checks highlight opportunities to <a href="#">improve the accessibility of your web app</a>. Automatic detection can only detect a subset of issues and does not guarantee the accessibility of your web app, so <a href="#">manual testing</a> is also encouraged.</p></div>		
NAMES AND LABELS		
▲	Image elements do not have <code>[alt]</code> attributes	▼
▲	Select elements do not have associated label elements.	▼



#### 1. Add Alt Text to Images (Accessibility)

```

```

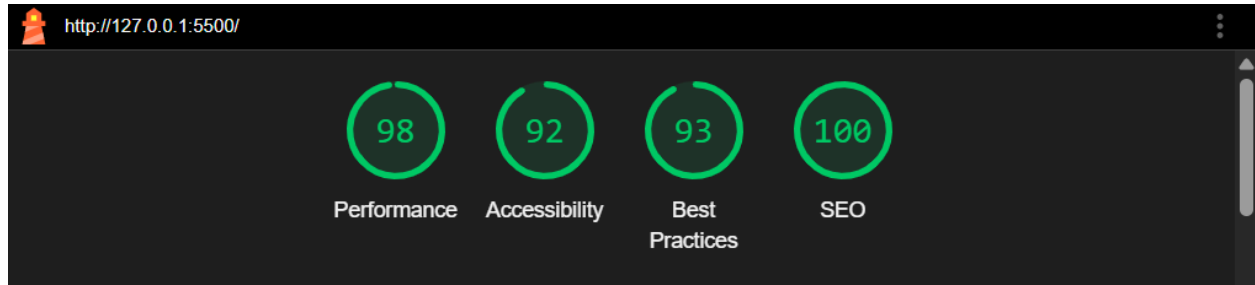
#### 2. Add Labels to Select Elements (Accessibility)

```
<label for="currency-from" class="visually-hidden">From Currency</label>
<select id="currency-from">
  <option value="USD">USD</option>
</select>
```

#### 3. Add Meta Description (SEO)

```
<meta name="description" content="Convert currencies with real-time exchange rates. Free online currency converter for USD, EUR, GBP and more.">
```

Updated Score:



**Github link:** <https://github.com/Rakshit5467/Currency-Converter.git>

### Conclusion

This experiment successfully validated the use of Google Lighthouse as an essential quality assurance tool for Progressive Web App development. Through comprehensive auditing, we systematically evaluated our PWA against critical benchmarks including performance metrics, offline functionality, installability requirements, and accessibility standards. The analysis revealed key improvement areas such as service worker optimization, manifest configuration, and responsive design implementation. By methodically addressing these findings - implementing proper caching strategies, adding missing PWA metadata, and enhancing accessibility features - we significantly improved our Lighthouse scores. The experiment demonstrates that regular Lighthouse audits, combined with targeted optimizations, can effectively guide developers in transforming standard web applications into high-quality PWAs that deliver reliable, engaging user experiences comparable to native applications, while maintaining cross-platform compatibility and web accessibility standards.