# MPL Experiment 11 (PWA)

**Name:** Devansh Wadhwani – 64

Kunal Punjabi – 43

Manav Punjabi – 44

Hitesh Rohra - 46 **Class:** D15A

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

**Theory:**

**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.

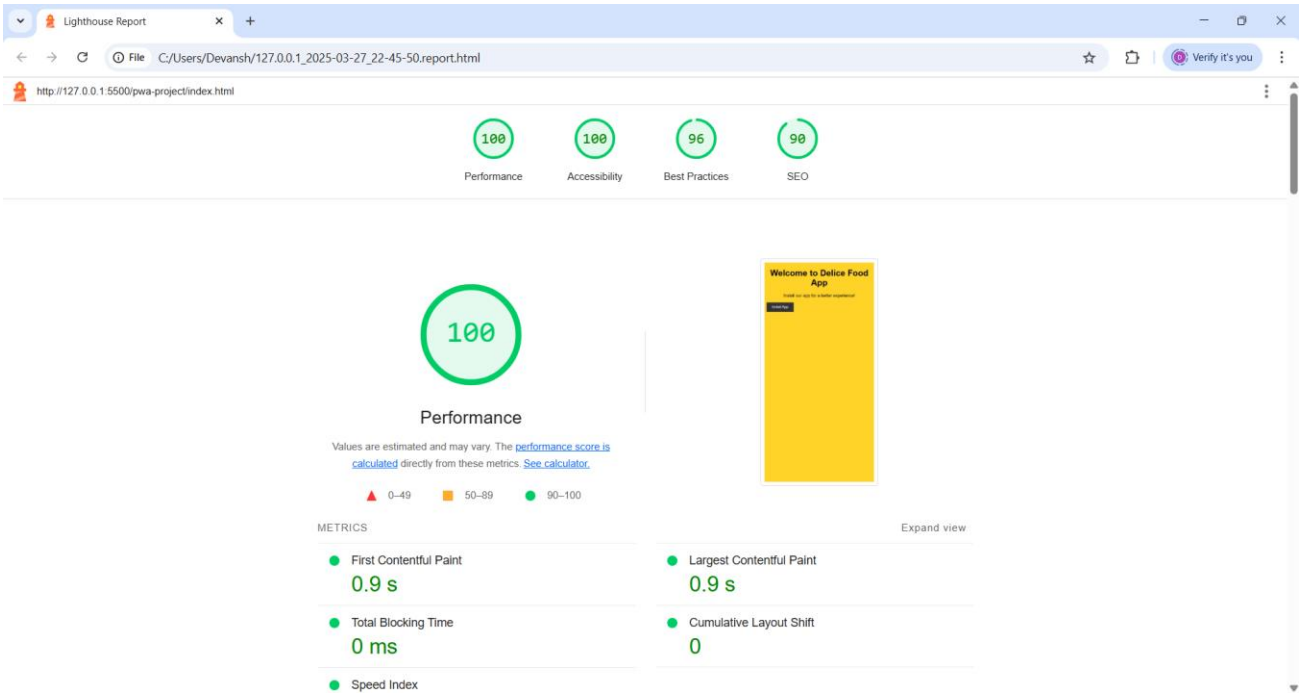
- **PWA Score (Mobile):**

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.

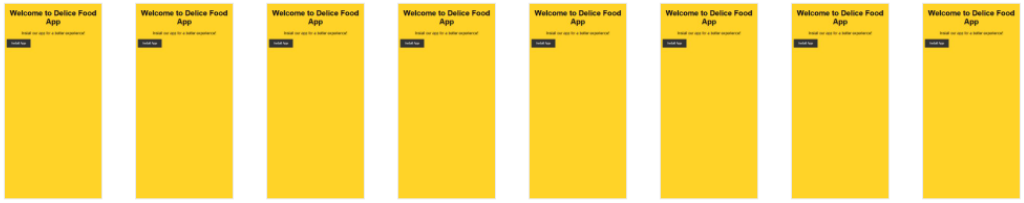
- **Accessibility:**
  As you might have guessed, this metric is a measure of how accessible your website is, across a plethora of accessibility features that can be implemented in your page (such as the 'aria-' attributes like aria-required, audio captions, button names, etc.). Unlike the other metrics 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

**0utput:**


Lighthouse Report
File C:/Users/Devansh/127.0.0.1_2025-03-27_22-45-50.report.html
Verify it's you
http://127.0.0.1:5500/pwa-project/index.html
100
Performance
100
Accessibility
96
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
Welcome to Delice Food App
METRICS
Expand view
First Contentful Paint
0.9 s
Largest Contentful Paint
0.9 s
Total Blocking Time
0 ms
Cumulative Layout Shift
0
Speed Index

0.9 s

View Treemap




Show audits relevant to: All FCP LCP

DIAGNOSTICS

- Eliminate render-blocking resources — Potential savings of 290 ms
- Page prevented back/forward cache restoration — 1 failure reason
- Avoid chaining critical requests — 2 chains found
- Largest Contentful Paint element — 910 ms

More information about the performance of your application. These numbers don't directly affect the Performance score.



## 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.

ADDITIONAL ITEMS TO MANUALLY CHECK (10) Hide

- Interactive controls are keyboard focusable
- Interactive elements indicate their purpose and state
- The page has a logical tab order
- Visual order on the page follows DOM order
- User focus is not accidentally trapped in a region
- The user's focus is directed to new content added to the page

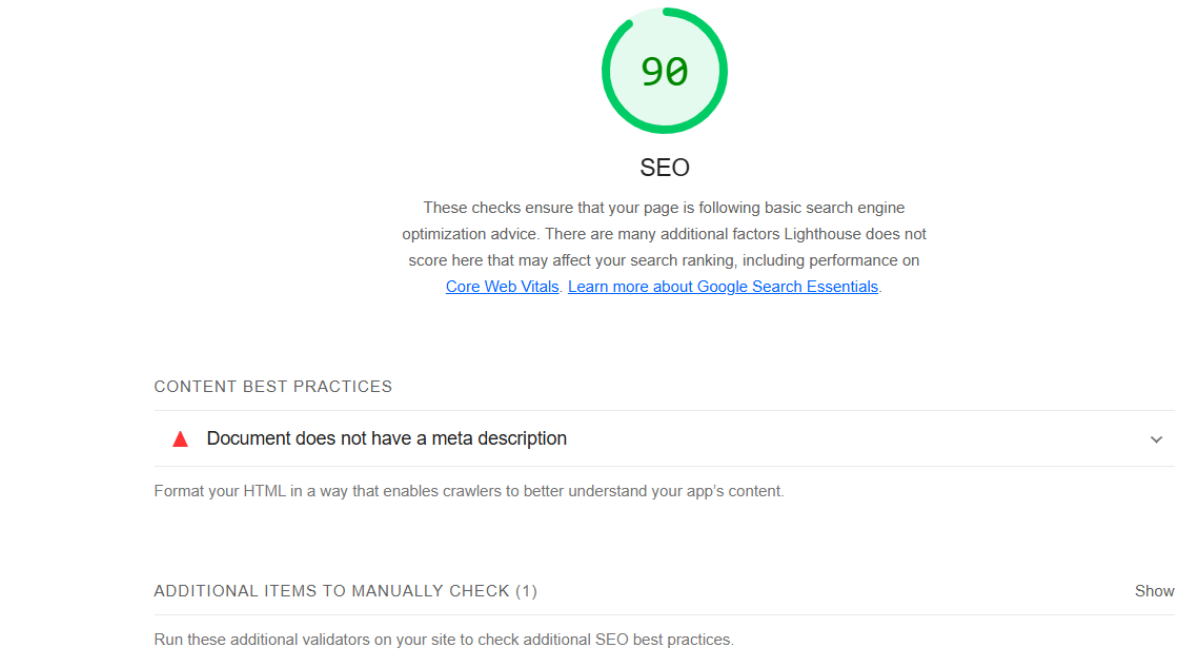

Best Practices

GENERAL

- Browser errors were logged to the console

TRUST AND SAFETY

- Ensure CSP is effective against XSS attacks
- Use a strong HSTS policy
- Ensure proper origin isolation with COOP
- Mitigate clickjacking with XFO or CSP




**Conclusion:**

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