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

Theory:Progressive Web Apps (PWAs) are web applications designed to provide a seamless, app-like experience on the web. Key characteristics of PWAs include:

1. Responsive Design: PWAs automatically adjust to different screen sizes and devices, providing a consistent user experience across mobile, tablet, and desktop.
2. Offline Capabilities: PWAs can function without an internet connection, thanks to service workers that cache assets and enable offline functionality.
3. App-like Experience: PWAs provide an app-like feel, including features such as home screen installation, push notifications, and improved loading performance.
4. Web App Manifest: A PWA includes a manifest file that defines how the app should appear when installed on the device, such as app name, icons, and theme color.

Google Lighthouse is an automated tool by Google that audits web pages for performance, accessibility, SEO, and best practices. Specifically, for PWAs, Lighthouse checks:

- Service Worker: Ensures that your app has a properly configured service worker for offline functionality.
- Web App Manifest: Validates that your app includes a manifest file with the necessary metadata for installation.
- Offline Functionality: Verifies that the app works offline by caching assets and supporting service worker operations.
- Performance Metrics: Assesses loading time, interactivity, and other critical aspects for a smooth user experience.

Objective:

- To evaluate the performance, PWA compliance, and best practices of the web app.
- To identify areas for improvement in the offline functionality, manifest file configuration, and service worker setup.

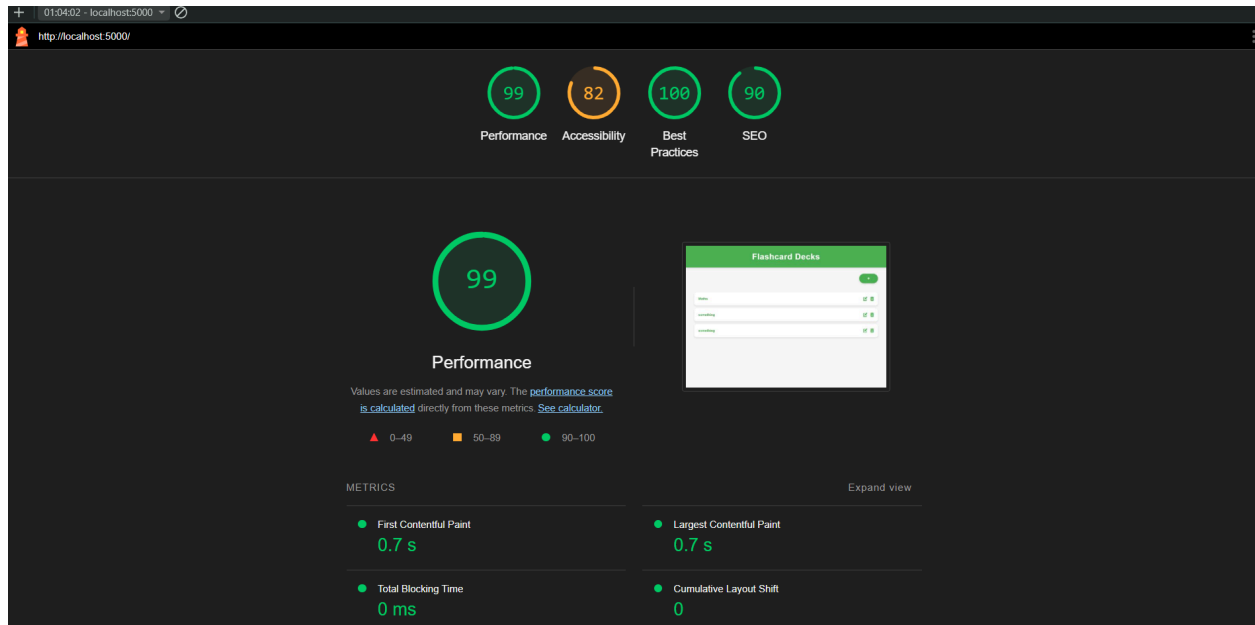Steps to Use Google Lighthouse for PWA Analysis:

1. Open Google Chrome:
   ○ Make sure you're using Google Chrome, as Lighthouse is integrated into Chrome's DevTools.
2. Open Your PWA in Google Chrome:
   ○ Navigate to your PWA URL in Google Chrome.
3. Open Chrome DevTools:
   ○ Right-click anywhere on the page and select Inspect (or use the shortcut Ctrl+Shift+I on Windows/Linux or Cmd+Opt+I on macOS).
   ○ Click on the Lighthouse tab in the DevTools panel. If it's not visible, click the » icon to find it.
4. Configure Lighthouse Settings:
   ○ Categories: Choose the categories you want to audit, including Performance, PWA, Accessibility, Best Practices, and SEO.
   ○ Device Mode: Select between Mobile or Desktop to simulate the audit on different devices.
   ○ Make sure to select the PWA category for a specific audit on Progressive Web App functionality.
5. Run the Lighthouse Audit:
   ○ Click the "Generate report" button to begin the audit. Lighthouse will analyze your PWA and generate a report.
6. Review the Results:
   ○ Once the audit is completed, you'll see a detailed report with scores in different categories. In the PWA section, Lighthouse will evaluate:
     ■ Service Worker: Checks if a service worker is installed and functioning correctly.
     ■ Web App Manifest: Verifies the presence and correctness of your manifest file.
     ■ Offline Capability: Assesses if your PWA can function offline.
     ■ App Installability: Tests whether the app can be installed on the home screen.
7. Analyze Issues and Recommendations:

- ○ Review the findings and follow Lighthouse's suggestions to improve areas such as offline support, manifest configuration, and service worker implementation.
8. Re-test After Fixes:
   - ○ After applying fixes to the identified issues, re-run the Lighthouse audit to ensure the PWA now meets the required standards.

Screenshots:

99 Performance

82 Accessibility

100 Best Practices

90 SEO

99

Performance

Values are estimated and may vary. The performance score is calculated directly from these metrics. See calculator.

▲ 0–49   ■ 50–89   ● 90–100

Flashcard Decks

METRICS

Expand view

● First Contentful Paint
0.7 s

● Largest Contentful Paint
0.7 s

● Total Blocking Time
0 ms

● Cumulative Layout Shift
0

DIAGNOSTICS

▲ Eliminate render-blocking resources — Potential savings of 280 ms ⌄

▲ Reduce unused CSS — Potential savings of 14 KiB ⌄

■ Minify CSS — Potential savings of 2 KiB ⌄

■ Enable text compression — Potential savings of 6 KiB ⌄

○ Avoid large layout shifts — 1 layout shift found ⌄

○ Avoid chaining critical requests — 1 chain found ⌄

○ Minimize third-party usage — Third-party code blocked the main thread for 0 ms ⌄

○ Largest Contentful Paint element — 710 ms ⌄

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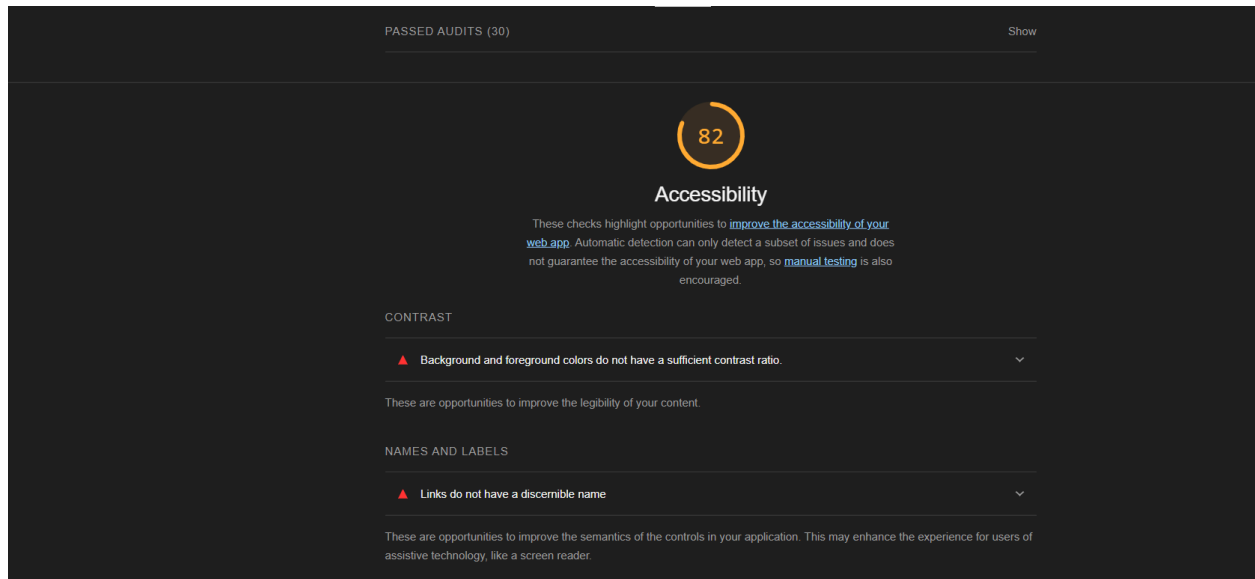
Conclusion:

Using the Google Lighthouse PWA Analysis Tool is a crucial step in ensuring that your Progressive Web App meets key functionality standards. By following the steps above, you can identify and resolve any issues related to service workers, manifest files, and offline capabilities. This process helps optimize performance, enhances the user experience, and ensures compliance with PWA best practices.