

EXPERIMENT NO: - 11**Name:-** Tejas Gunjal**Class:-** D15A**Roll:No: -** 18**AIM: -** To use google Lighthouse PWA Analysis Tool to test the PWA functioning.

Theory: -

Google Lighthouse is an open-source tool designed to evaluate and optimize web applications based on multiple key parameters, including performance, accessibility, Progressive Web App (PWA) implementation, and best practices. It provides a detailed automated audit that helps developers improve their websites efficiently. Unlike traditional manual audits that can take days or weeks, Lighthouse generates insights within minutes with minimal setup.

One of Lighthouse's biggest advantages is its ease of use—simply run it on a webpage or provide a URL, and it will generate a comprehensive report. Lighthouse supports audits for both desktop and mobile versions of a website, ensuring an optimal user experience across different devices.

Key Features and Audit Metrics**1. Performance**

This metric evaluates how efficiently a webpage loads and becomes interactive. It considers several factors, including:

- Page Load Speed – Measures how quickly content appears to users.
- First Contentful Paint (FCP) – Time taken for the first visible content to load.
- Largest Contentful Paint (LCP) – Measures how long the largest visible content takes to fully render.
- Cumulative Layout Shift (CLS) – Ensures content does not shift unexpectedly, improving user experience.
- Time to Interactive (TTI) – The time taken for the webpage to become fully functional.

Lighthouse assigns a performance score from 0 to 100, where:

- 100 → Top 2% of websites (98th percentile)
- 50 → Around the 75th percentile
- Lower scores → Indicate areas needing optimization

2. Progressive Web App (PWA) Score

With the increasing adoption of PWAs, web applications now strive to deliver an app-like experience. Lighthouse evaluates a website's PWA readiness based on Google's Baseline PWA Checklist, which includes:

- **Service Worker Implementation** – Enables offline functionality and background synchronization.
- **App Manifest Compliance** – Provides metadata for mobile app-like integration.
- **Viewport Configuration** – Ensures responsiveness across different screen sizes.
- **Performance in Script-Disabled Environments** – Verifies that the page functions properly even if JavaScript is disabled.

A high PWA score indicates that the application meets essential criteria for speed, reliability, and mobile usability.

3. Accessibility

This metric determines how well a website supports users with **visual, cognitive, or physical disabilities**. Lighthouse evaluates accessibility based on:

- **ARIA Attributes** – Enhances screen reader support (e.g., aria-required).
- **Text Alternatives for Media** – Ensures images, audio, and video content are accessible.
- **Semantic HTML Usage** – Encourages proper use of elements like <section>, <article>, and <button> for better screen-reader compatibility.

Unlike other metrics, **accessibility follows a pass/fail approach**—missing key features can **significantly impact the overall score**. Improving accessibility ensures **inclusivity for all users**.

4. Best Practices

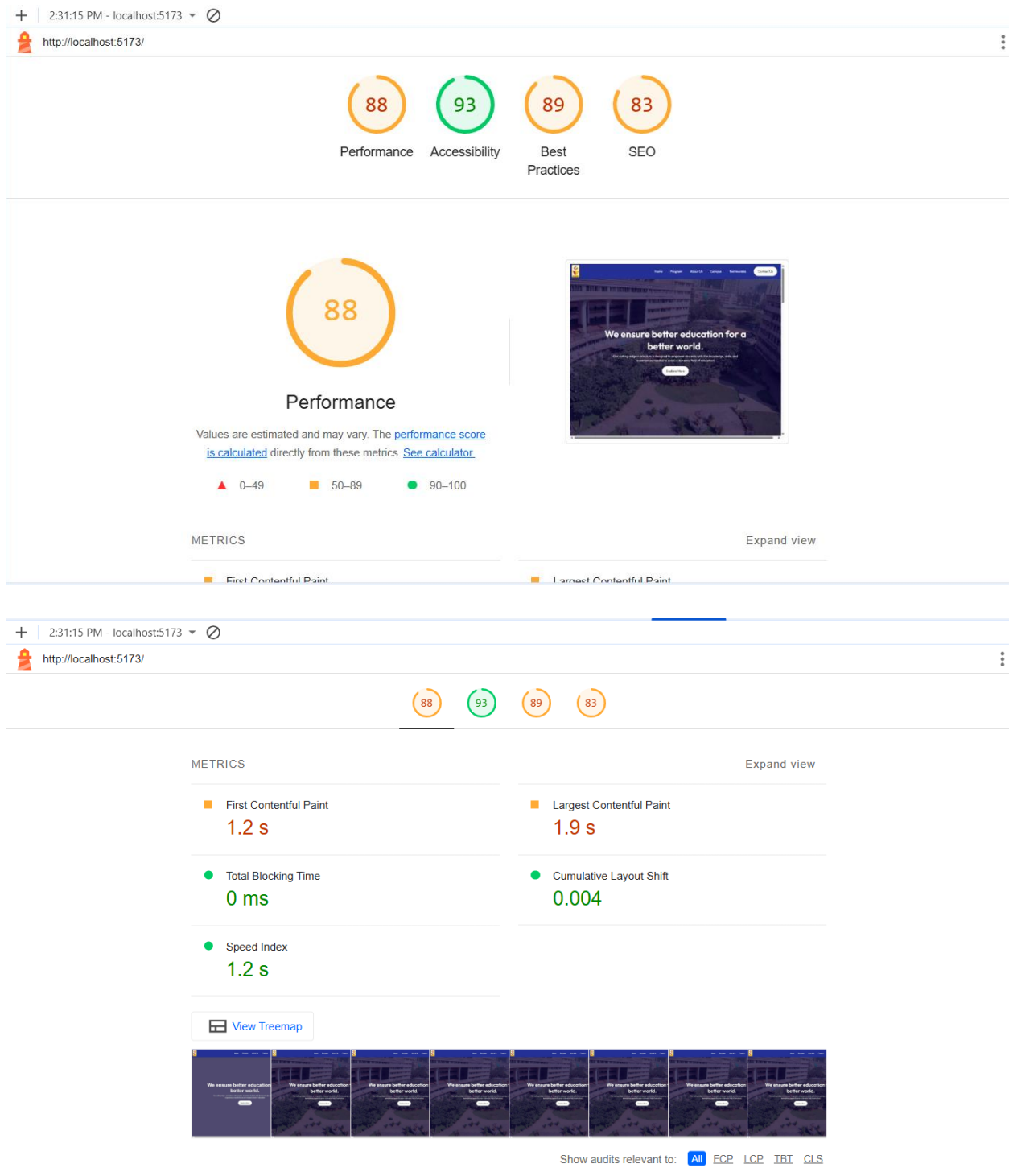
Lighthouse also assesses whether the website follows **modern web development best practices**, including:

- **Use of HTTPS** – Ensures secure data transmission.
- **Avoiding Deprecated Code** – Prevents the use of outdated elements, directives, and libraries.
- **Secure Password Inputs** – Disables paste-into fields to reduce security risks.
- **User Security Alerts** – Provides notifications for **geo-location access and cookie usage**.

A high **Best Practices score** means the website meets industry standards, leading to better **security, maintainability, and overall performance**.

Manifest.json

```
{
  "short_name": "VESIT",
  "name": "Vivekanand Education Society Institue of Technology",
  "description": "Vivekanand Education Society's Institute of Technology, also known as VESIT or V. E. S. Institute of Technology, was established in 1984 as an engineering college affiliated with the University of Mumbai.",
  "start_url": "/",
  "display": "standalone",
  "background_color": "#ffffff",
  "theme_color": "#000000",
  "icons": [
    {
      "src": "/src/assets/icon1.png",
      "sizes": "192x192",
      "type": "image/png"
    },
    {
      "src": "/src/assets/icon1.png",
      "sizes": "512x512",
      "type": "image/png"
    }
  ]
}
```

Output: -**a) Performance**

2:31:15 PM - localhost:5173

http://localhost:5173/

88 93 89 83

SHOW DETAILS RELEVANT TO:

DIAGNOSTICS

- ▲ Enable text compression — Potential savings of 1,316 KiB
- ▲ Minify JavaScript — Potential savings of 781 KiB
- ▲ Largest Contentful Paint element — 1,950 ms
- ▲ Reduce unused JavaScript — Potential savings of 798 KiB
- ▲ Page prevented back/forward cache restoration — 1 failure reason
- Image elements do not have explicit `width` and `height`
- Serve images in next-gen formats — Potential savings of 37,194 KiB
- Warnings:** Unable to locate resource ...assets/college.png
- Properly size images — Potential savings of 41,201 KiB
- Efficiently encode images — Potential savings of 21 KiB

b) Accessibility

2:31:15 PM - localhost:5173

http://localhost:5173/

88 93 89 83

93

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.

NAMES AND LABELS

- ▲ Links do not have a discernible name

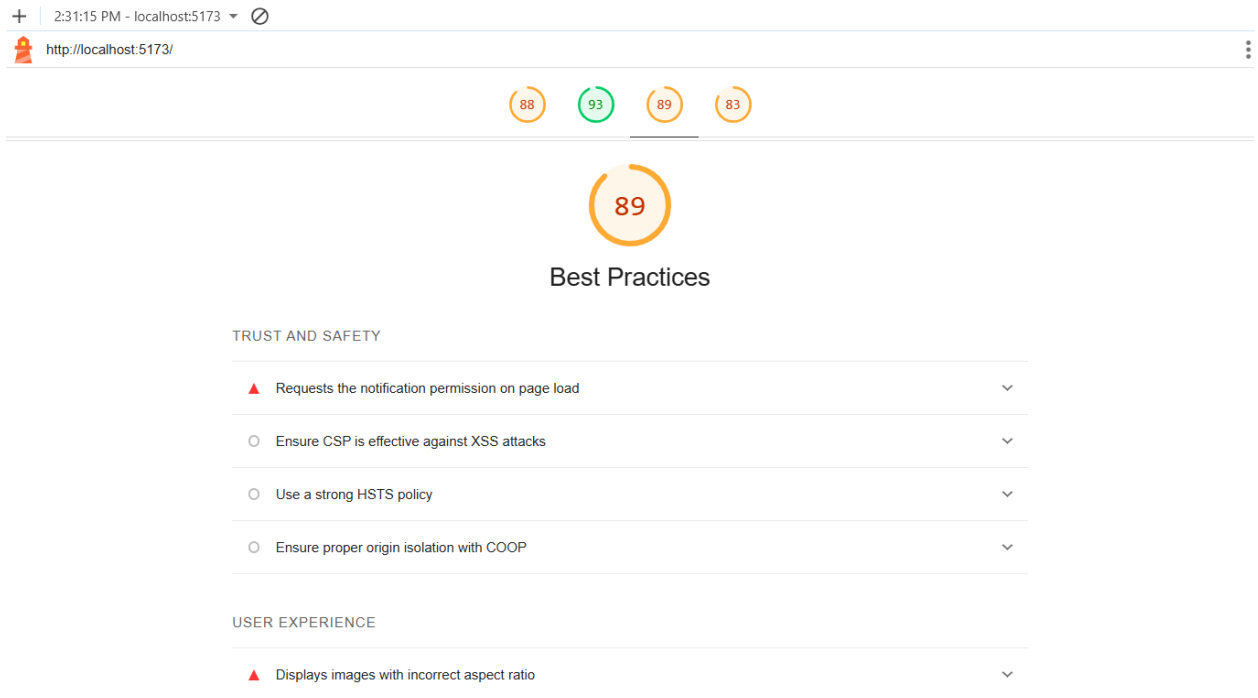
These are opportunities to improve the semantics of the controls in your application. This may enhance the experience for users of assistive technology, like a screen reader.

ADDITIONAL ITEMS TO MANUALLY CHECK (10)

These items address areas which an automated testing tool cannot cover. Learn more in our guide on [conducting an accessibility review](#).

Show

c) Best Practices



d) SEO [Search Engine Optimization]

