

MAD & PWA Lab

Journal

Experiment No.	11
Experiment Title.	To use google Lighthouse PWA Analysis Tool to test the PWA functioning.
Roll No.	14
Name	Komal Milind Deolekar
Class	D15A
Subject	MAD & PWA Lab
Lab Outcome	LO6: Develop and Analyze PWA Features and deploy it over app hosting solution
Grade:	

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

Theory :

Reference : <https://www.semrush.com/blog/google-lighthouse/>

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

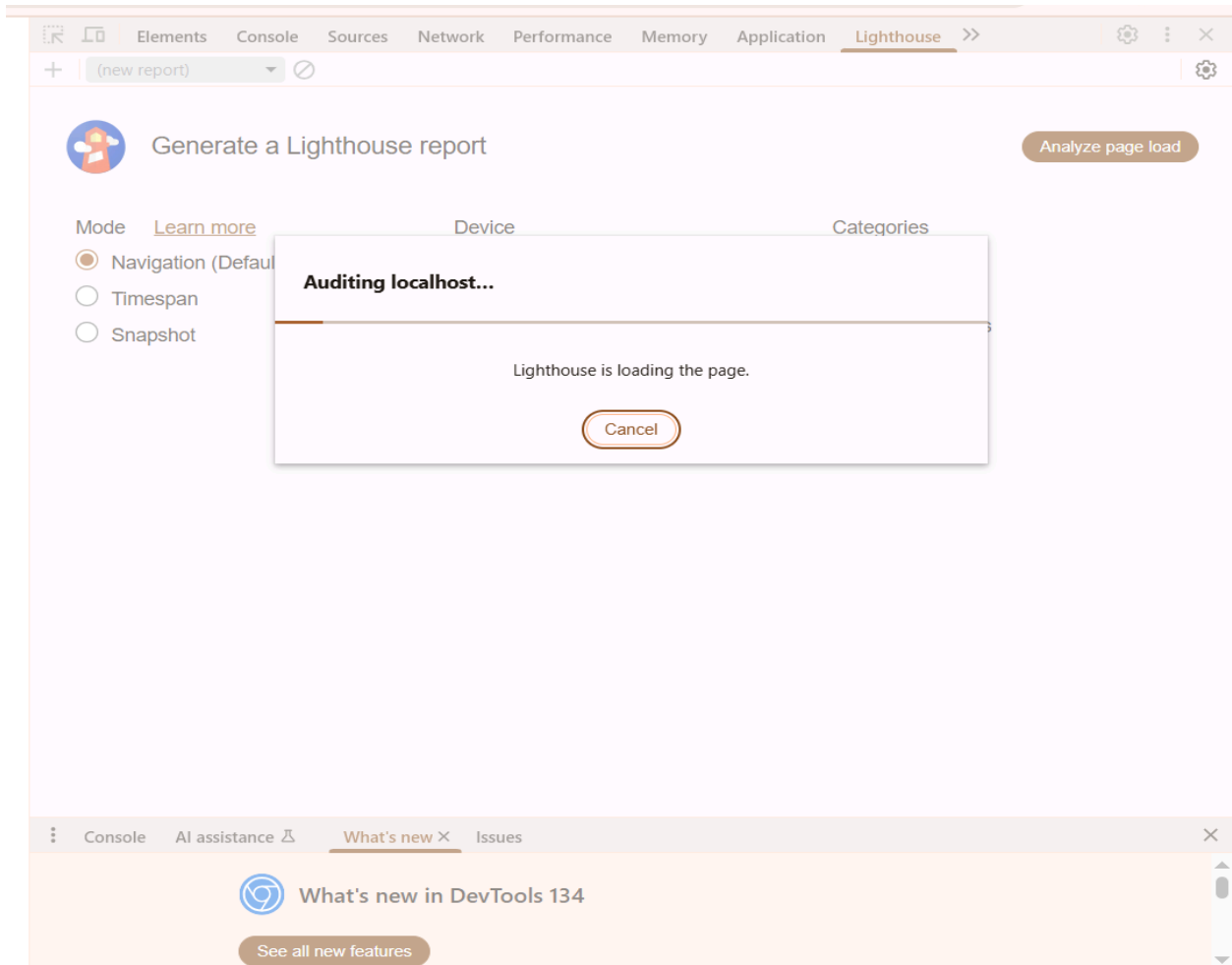
Google Lighthouse has the option of running the Audit for Desktop as well as mobile version of your page(s). The top metrics that will be measured in the Audit are:

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

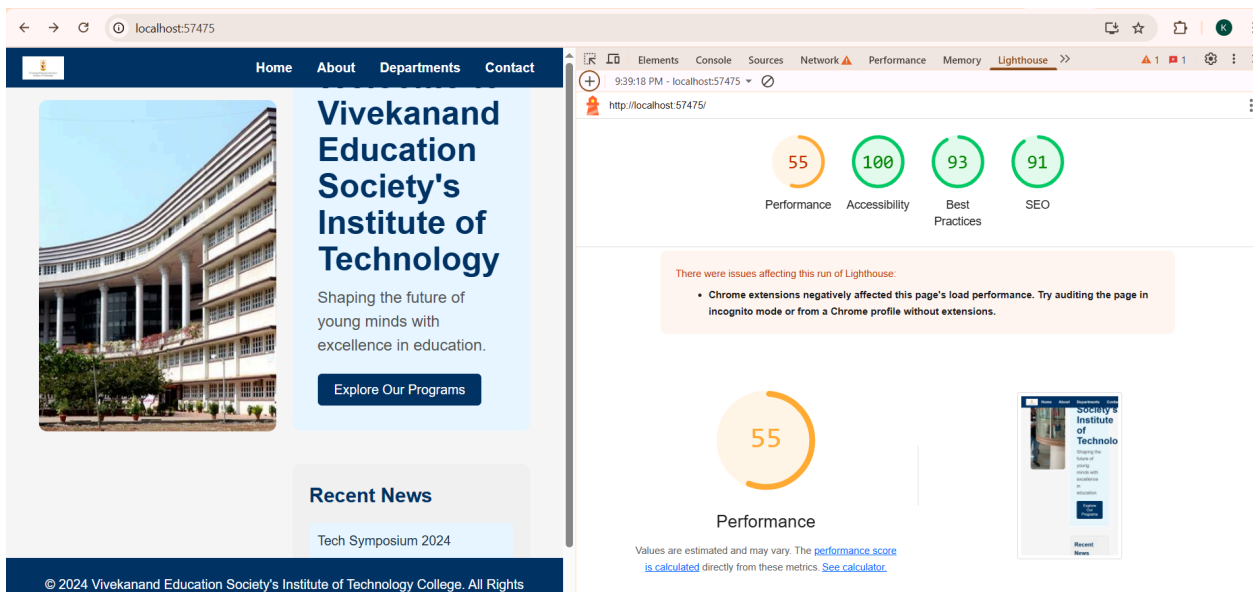
the PWA paradigm, where the objective is to make the application behave as close as possible to native mobile applications. Scoring points are based on the Baseline PWA checklist laid down by Google which includes Service Worker implementation(s), viewport handling, offline functionality, performance 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.

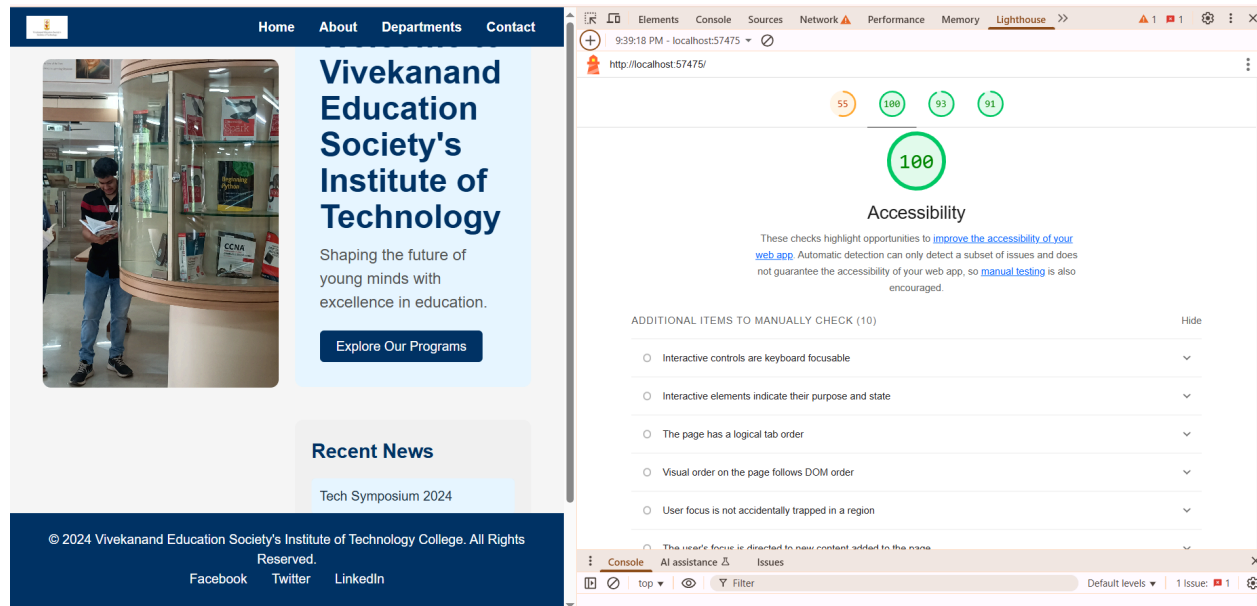
Output :



Performance



Accessibility



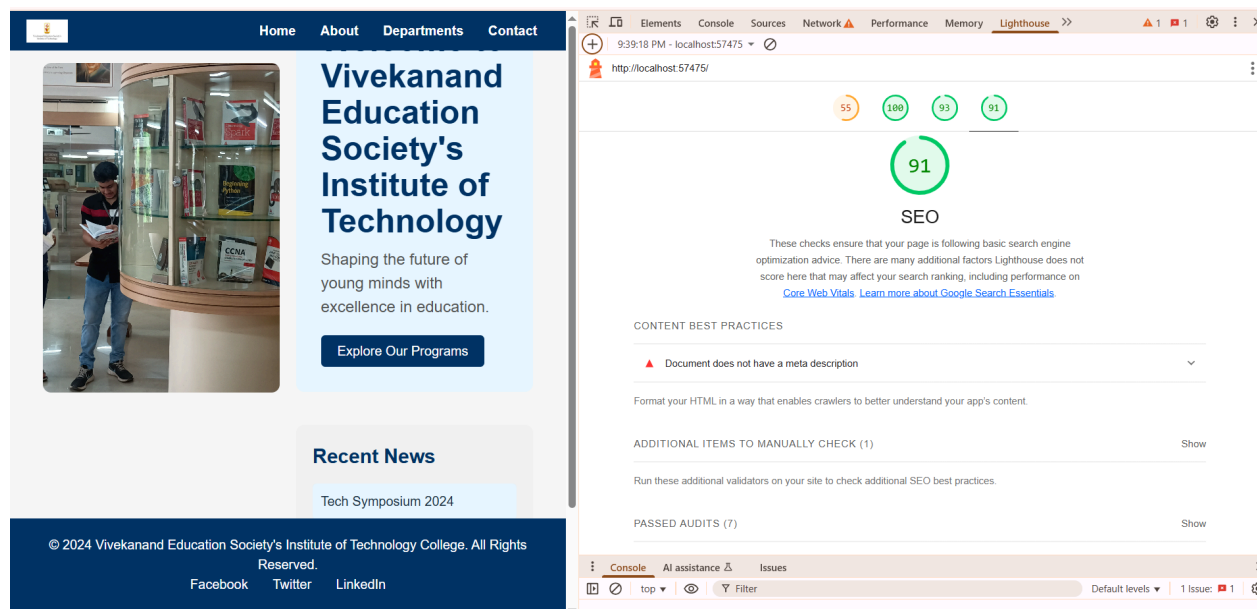
The screenshot displays the VESIT Website and its Accessibility audit results. The website features a dark blue header with navigation links: Home, About, Departments, and Contact. The main content area includes a large image of a student in a library, the text "Vivekanand Education Society's Institute of Technology", the tagline "Shaping the future of young minds with excellence in education.", and a button "Explore Our Programs". Below this is a "Recent News" section with the title "Tech Symposium 2024". The footer contains copyright information for 2024 and social media links for Facebook, Twitter, and LinkedIn.

The Accessibility audit results are shown on the right. The overall score is 100. The audit highlights opportunities to improve the accessibility of the web app. The additional items to manually check are:

- 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 audit also notes that the user's focus is directed to new content added to the page.

SEO



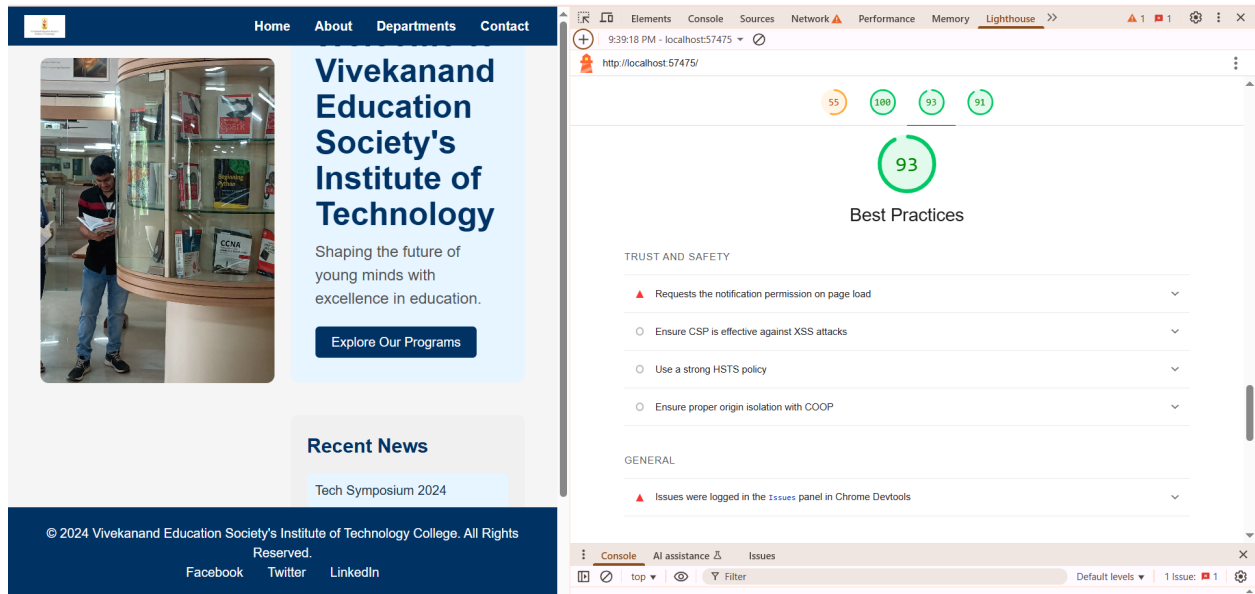
The screenshot displays the VESIT Website and its SEO audit results. The website features a dark blue header with navigation links: Home, About, Departments, and Contact. The main content area includes a large image of a student in a library, the text "Vivekanand Education Society's Institute of Technology", the tagline "Shaping the future of young minds with excellence in education.", and a button "Explore Our Programs". Below this is a "Recent News" section with the title "Tech Symposium 2024". The footer contains copyright information for 2024 and social media links for Facebook, Twitter, and LinkedIn.

The SEO audit results are shown on the right. The overall score is 91. The audit highlights opportunities to improve the SEO of the web app. The content best practices are:

- Document does not have a meta description

The audit also notes that the user's focus is directed to new content added to the page.

Best practices



Changes made to the code :

Converted Images to webp from jpg , png

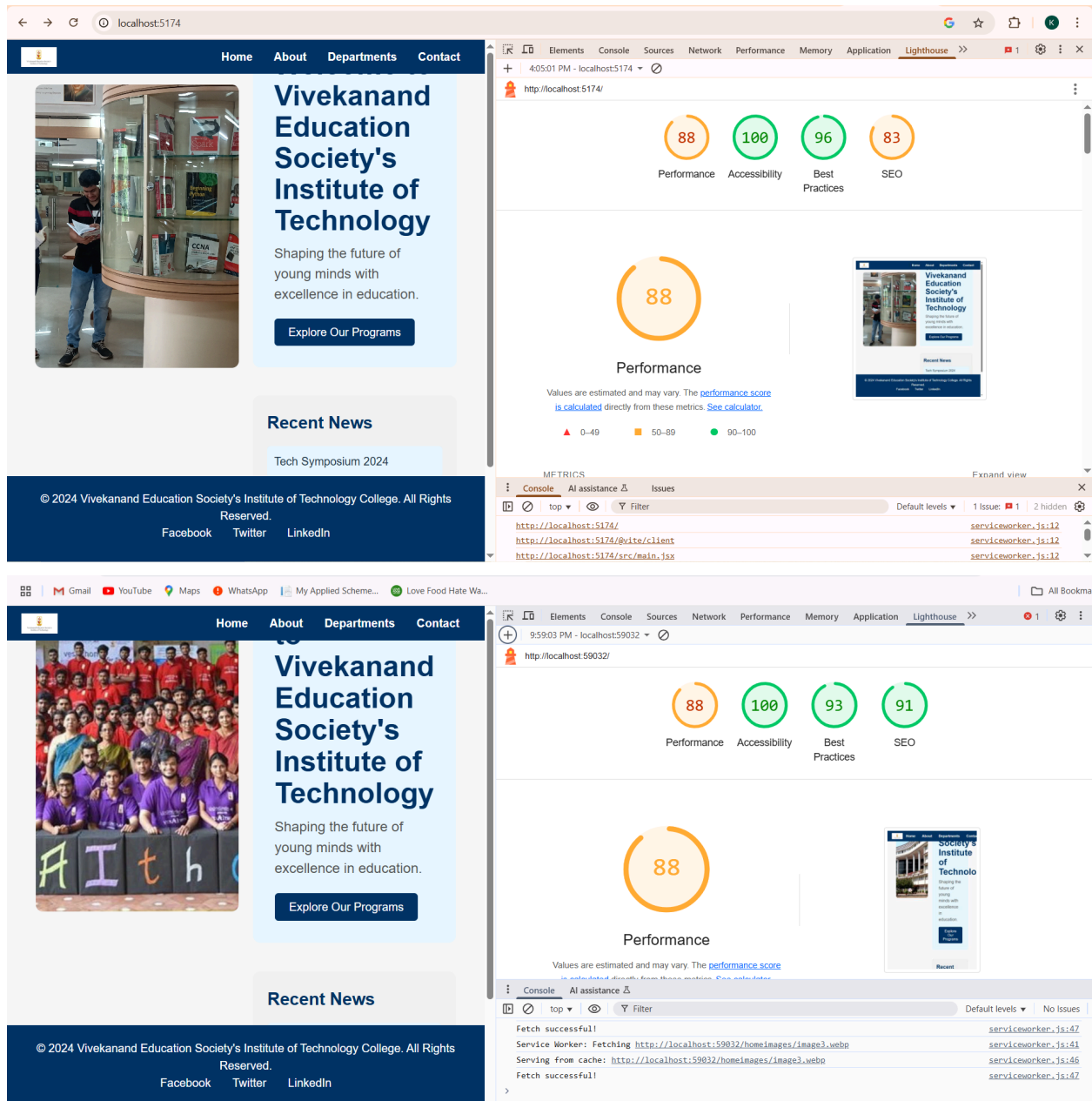
Changes in vite.config for adding compression

```
import { defineConfig } from 'vite'
import react from '@vitejs/plugin-react'
import viteCompression from 'vite-plugin-compression'
```

```
// https://vitejs.dev/config/
export default defineConfig({
  plugins: [react(), viteCompression()],
  build: {
    minify: 'esbuild',
    terserOptions: {
      compress: {
        drop_console: true, // remove console logs
      }
    }
  }
})
```

For preloading this time taking asset

```
<link rel="preload" href="/assets/index-DWhKUY4_.js" as="script">
```



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