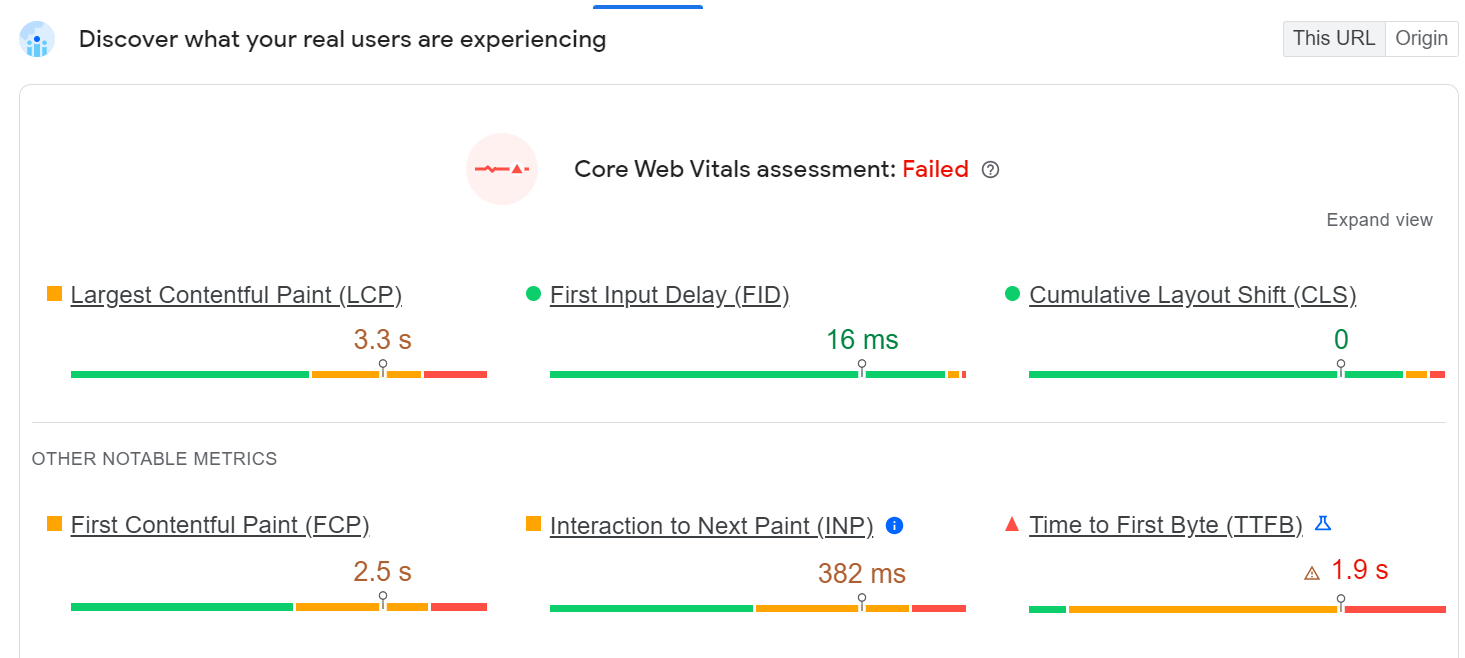
**Eduminatti**

Assignment for Web Development Internship

* **FIRST TASK**

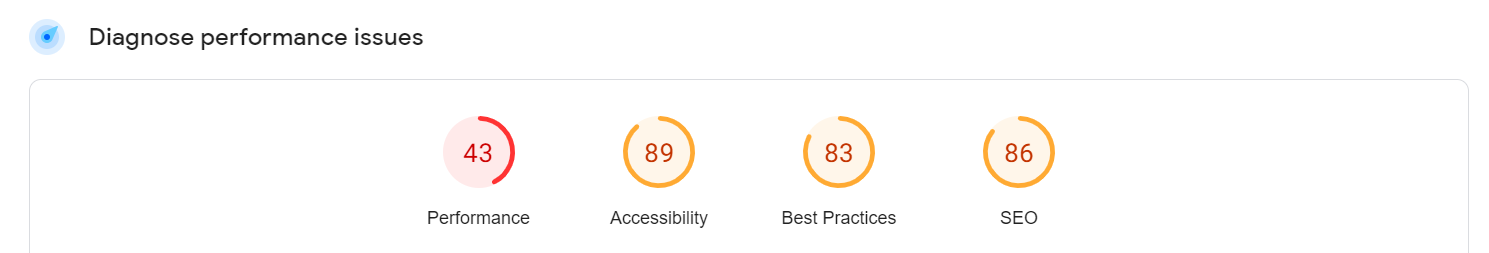
The Pageload speed of “<https://edumynation.com/category/schools-in-abuDhabi>” is :

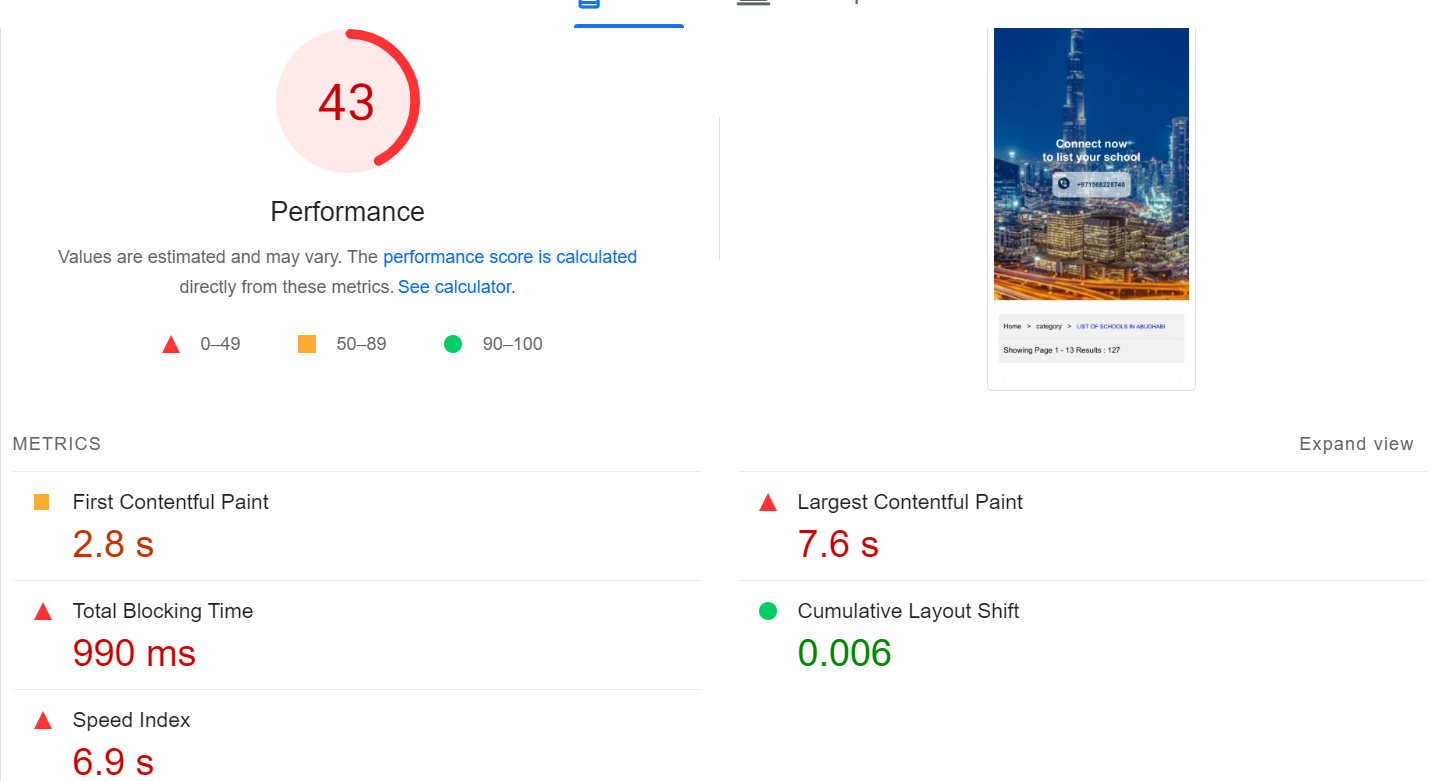
**In Mobile**



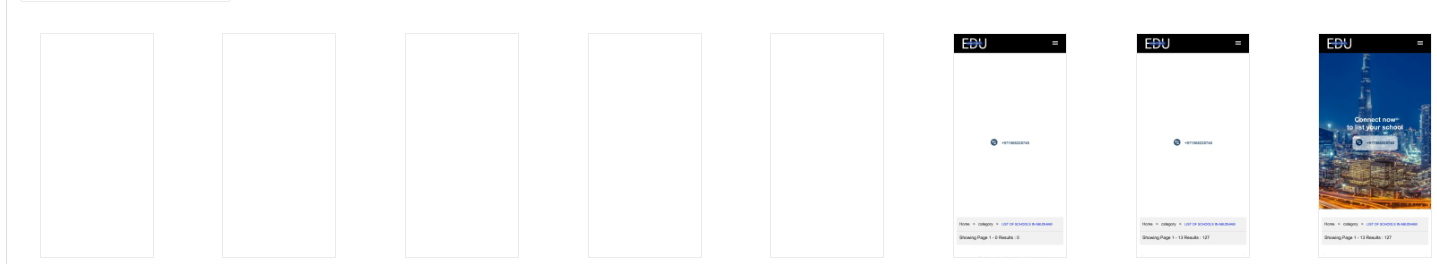
**Core Web Vitals ( CWV)** :

* Core Web Vitals are three metrics included in Google’s page experience signals that quantify the user experience of interacting with a page.
* Core Web Vitals measure how quickly a page is visible, how long it takes before a user can interact with the page, and check the visual stability of the pages.
* The metrics that currently make up Core Web Vitals are Largest Contentful Paint (LCP), First Input Delay (FID), and Cumulative Layout Shift (CLS), though these could change over time.
* [**Largest Contentful Paint (LCP)**](https://web.dev/lcp/)**:** Measures loading performance, and should occur within 2.5 seconds of when the page first starts loading.
* [**First Input Delay (FID)**](https://web.dev/fid/)**:** Measures time to interactivity, and should be less than 100 milliseconds.
* [**Cumulative Layout Shift (CLS)**](https://web.dev/cls/)**:** Measures visual stability, and should be less than 0.1.





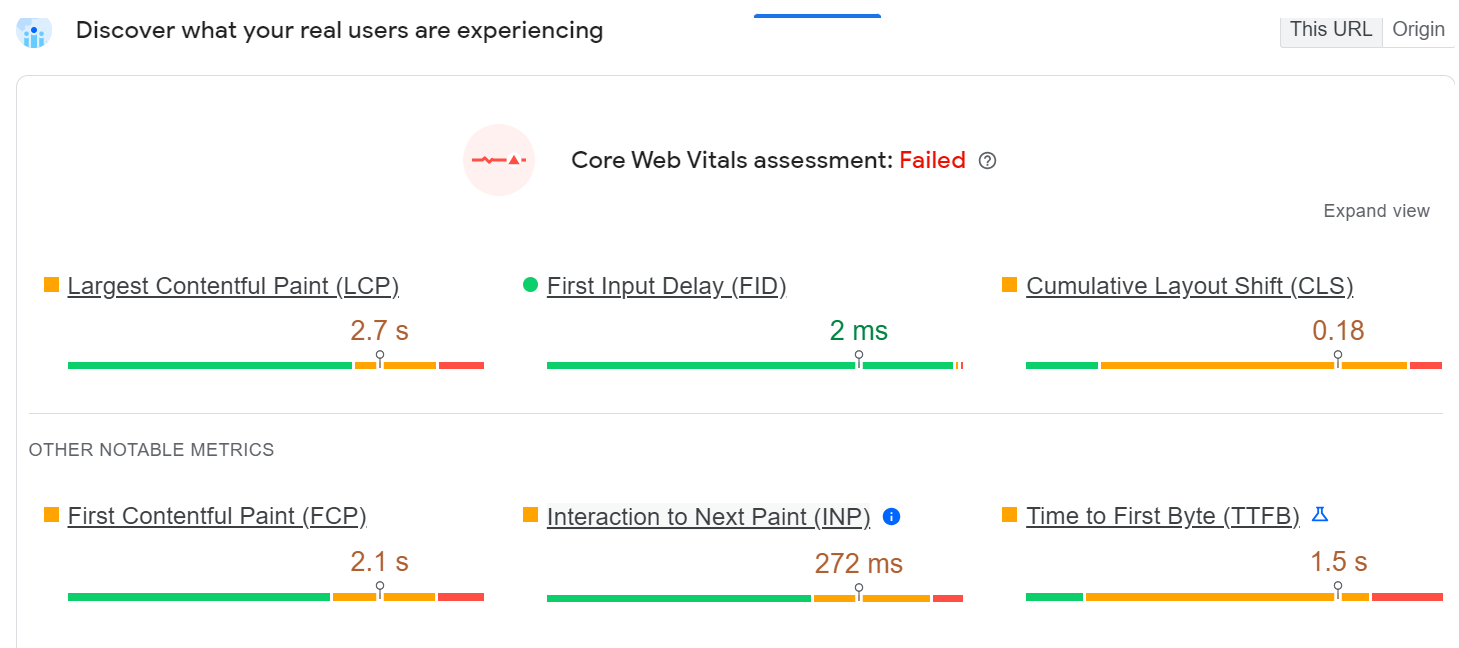
* **First Contentful Pain**t marks the time at which the first text or image is painted.
* **Largest Contentful Paint** marks the time at which the largest text or image is painted.
* **Total Blocking Time**: Sum of all time periods between FCP and Time to Interactive, when task length exceeded 50ms, expressed in milliseconds.
* **Cumulative layout shift** measures the movement of visible elements within the viewport.
* **Speed Index** shows how quickly the contents of a page are visibly populated.

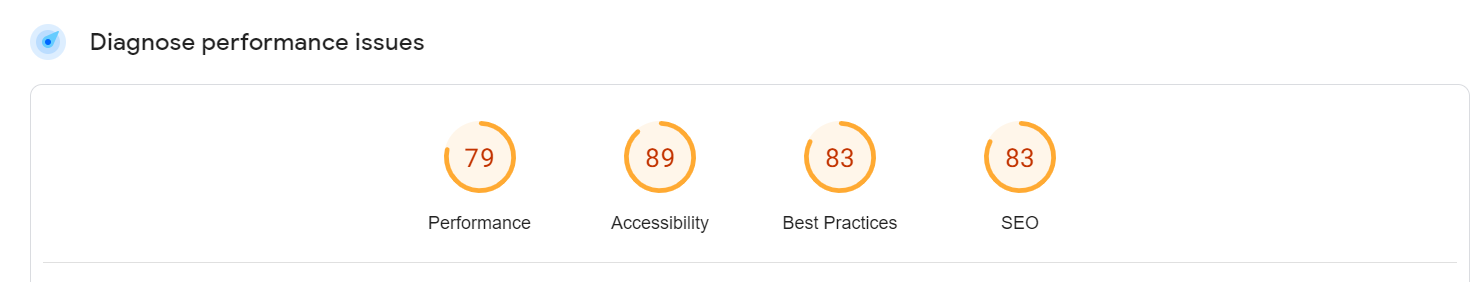


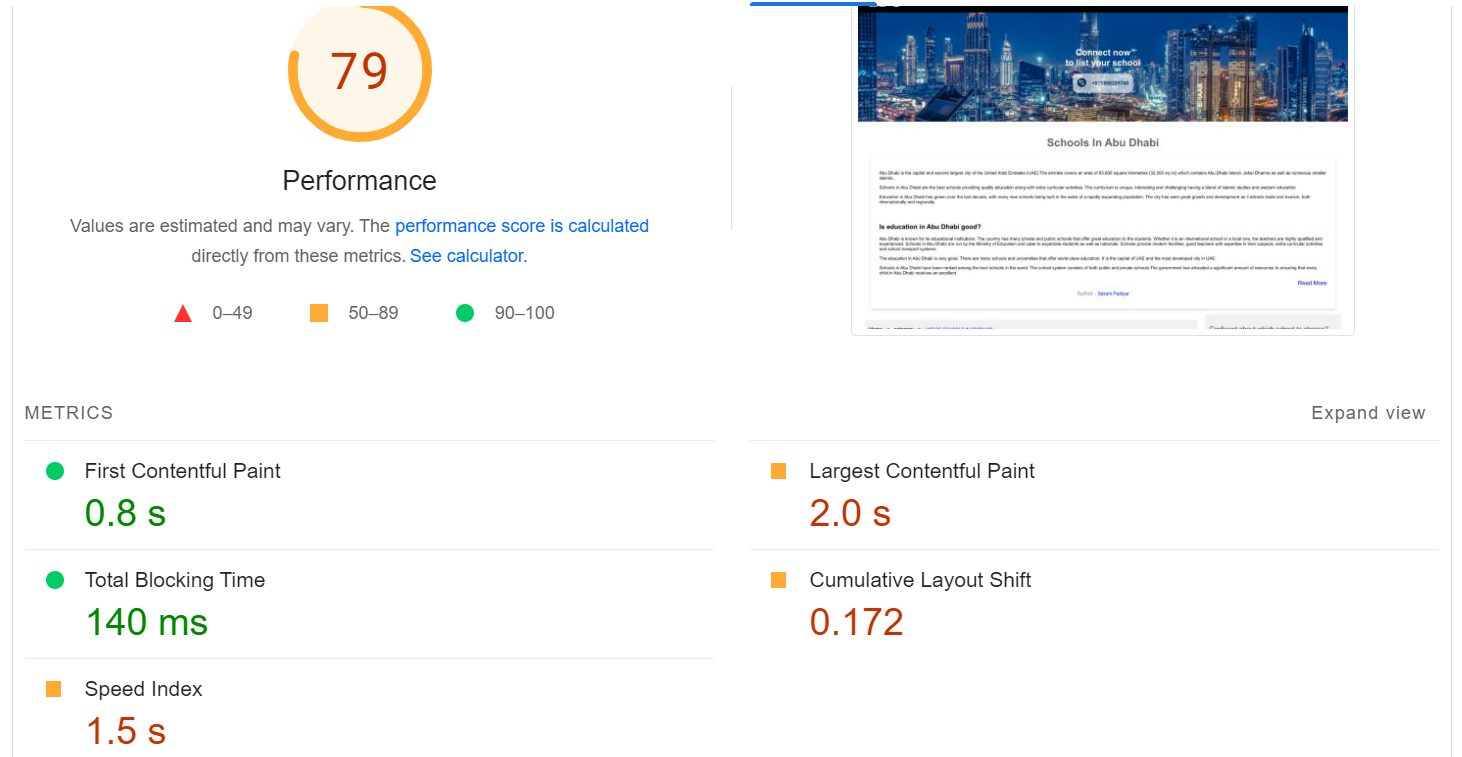
This suggestions can help your page load faster.

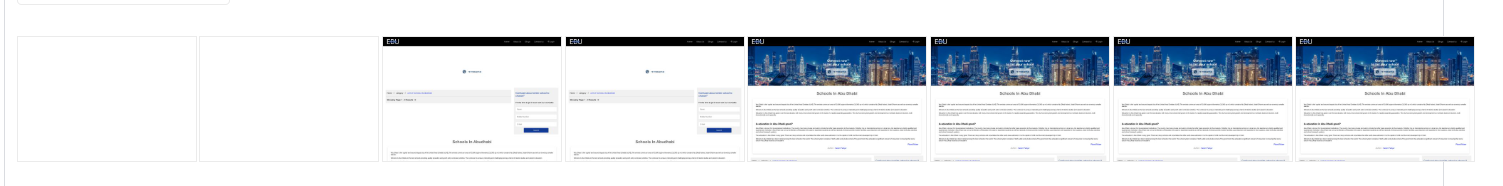
1. **Reduce unused JavaScript** : Reduce unused JavaScript and defer loading scripts until they are required to decrease bytes consumed by network activity.
2. **Reduce initial server response time** : Keep the server response time for the main document short because all other requests depend on it.
3. **Eliminate render-blocking resources** : Resources are blocking the first paint of your page. Consider delivering critical JS/CSS inline and deferring all non-critical JS/styles.
4. **Properly size images :** Serve images that are appropriately-sized to save mobile data and improve load time.
5. **Defer off-screen images** : Consider lazy loading offscreen and hidden images after all critical resources have finished loading to lower Time to Interactive.

**IN DESKTOP**









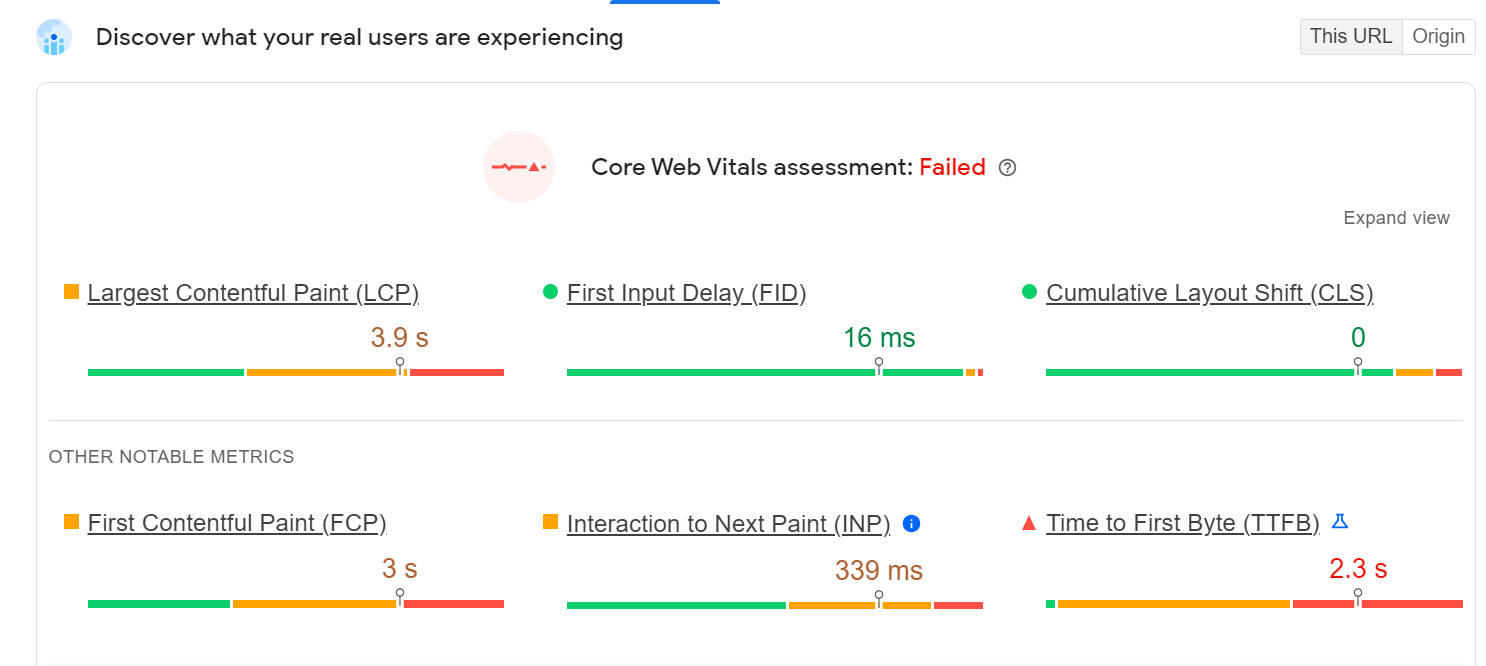
These suggestions can help your page load faster.

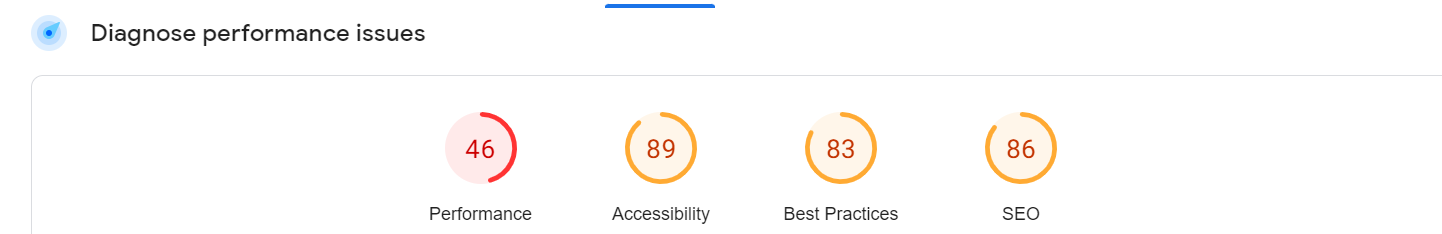
1. **Properly size images** : Serve images that are appropriately-sized to save mobile data and improve load time.
2. **Reduce unused JavaScript** : Reduce unused JavaScript and defer loading scripts until they are required to decrease bytes consumed by network activity.
3. **Reduce initial server response time** : Keep the server response time for the main document short because all other requests depend on it.
4. **Server images in next-gen formats** : Image formats like WebP and AVIF often provide better compression than PNG or JPEG, which means faster downloads and less data consumption.
5. **Eliminate render-blocking resources** : Resources are blocking the first paint of your page. Consider delivering critical JS/CSS inline and deferring all non-critical JS/styles.

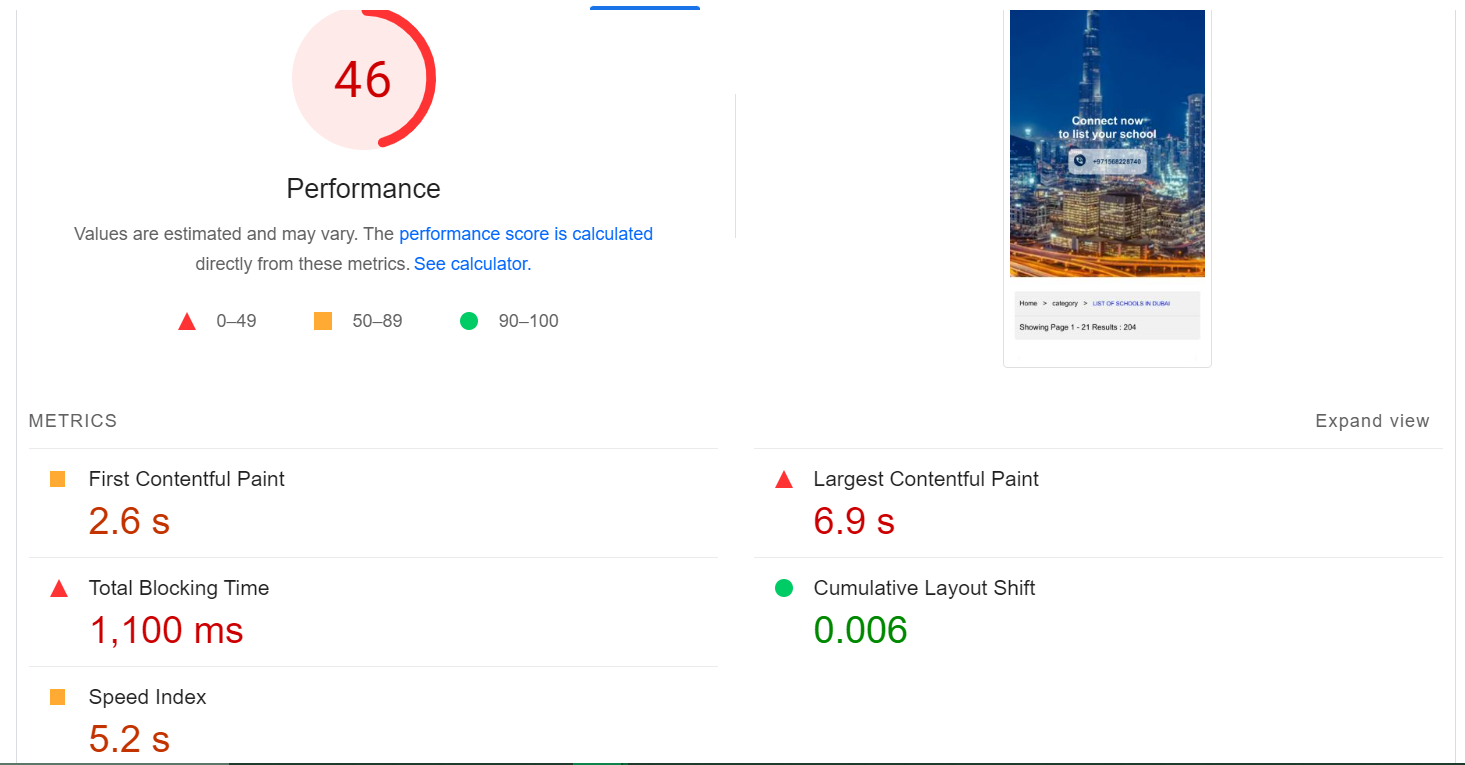
**SECOND TASK**

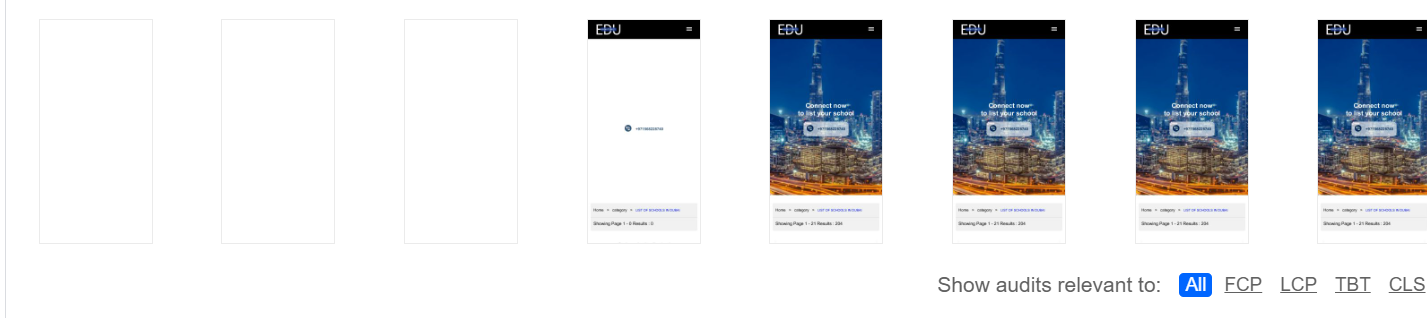
The Pageload speed of “https://edumynation.com/category/schools-in-dubai” is :

**In Mobile**

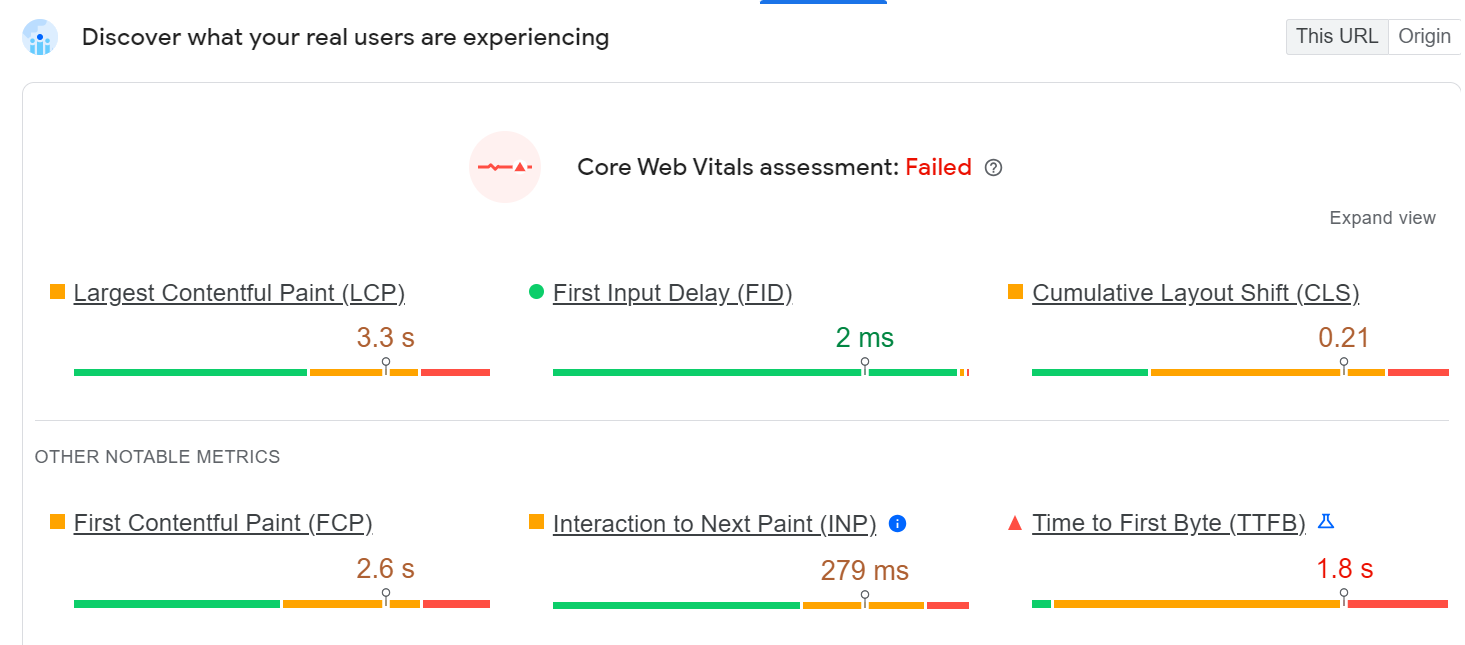


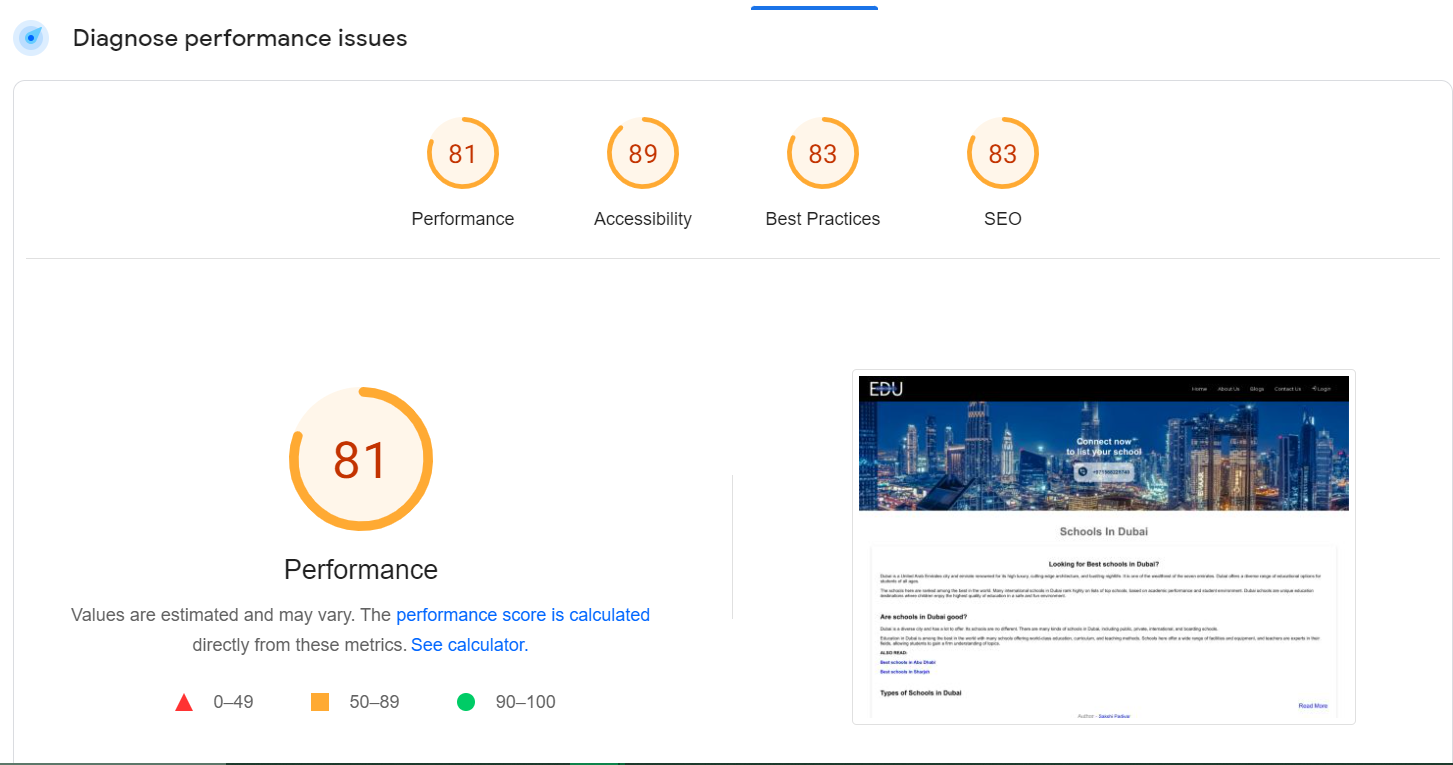


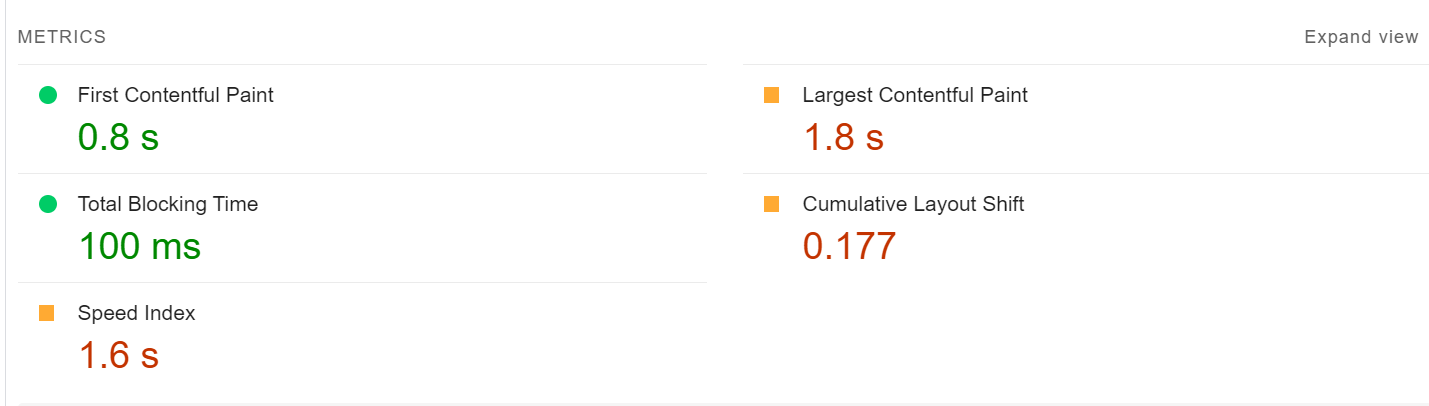


****

**IN DESKTOP**

****

****

****

Some diagnostics for both mobile and desktop are :

1. Reduce the impact the third-party code : Third-party code can significantly impact load performance. Limit the number of redundant third-party providers and try to load third-party code after your page has primarily finished loading.
2. Images elements do not have explicit width and height.
3. Minimise main-thread work.
4. Reduce JavaScript execution time.
5. Serve static assests with an efficient cache policy.
6. Avoid an execessive DOM size.
7. Avoid chaining critical requests.
8. User Timing marks and measures.
9. Keep request counts low and transfer sizes small.
10. Largest contentful paint element.
11. Avoid large layout shifts.
12. Avoid long main-thread tasks.