# Web Performance Audit: <https://oralb.com.au> Date: 20-07-2025

## Core Web Vitals Explained

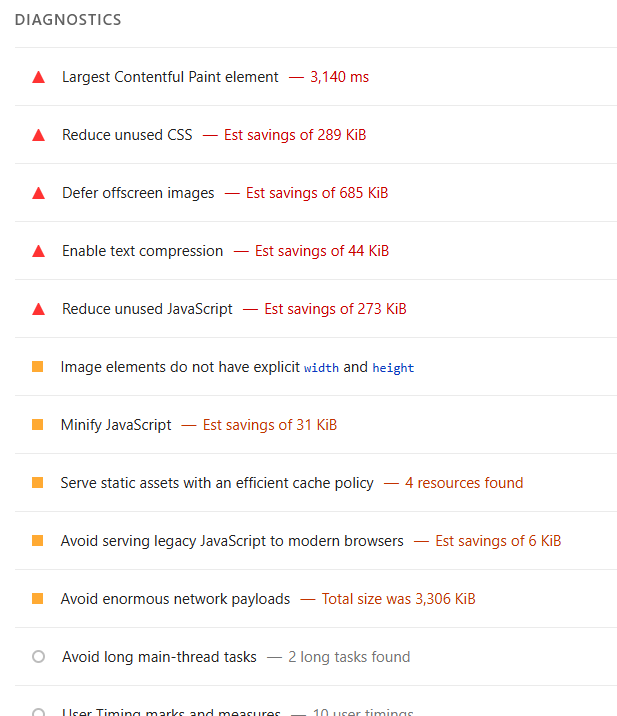
|  |  |
| --- | --- |
| Metric | Description |
| Largest Contentful Paint (LCP) | Measures loading performance. Ideal is < 2.5 seconds. |
| First Input Delay (FID) | Measures interactivity. Ideal is < 100 milliseconds. |
| Cumulative Layout Shift (CLS) | Measures visual stability. Ideal is < 0.1. |
| First Contentful Paint (FCP) | Time until the first text or image is painted. |
| Time to Interactive (TTI) | Time until the page is fully interactive. |
| Total Blocking Time (TBT) | Sum of all time periods between FCP and TTI when task length exceeded 50ms. |

Overall Report:  
It indicates the performance of application is fair but can improve with the recommendation implementation.

A screenshot of a website

AI-generated content may be incorrect.

## A screenshot of a computer AI-generated content may be incorrect.



## Identified Performance Issues

* High LCP indicating slow loading of main content.
* Elevated CLS due to layout shifts during page load.
* Long TTI suggesting heavy JavaScript execution.
* Large image sizes without optimization.
* Excessive use of third-party scripts.

## Recommendations to Improve Performance

* Optimize and compress images to reduce load time.
* Use `font-display: swap` to improve text rendering.
* Minimize and defer non-critical JavaScript.
* Implement lazy loading for offscreen images.
* Reduce third-party script usage and audit their impact.
* Use a Content Delivery Network (CDN) to serve assets faster.
* Preload key resources to improve LCP.

**Highest priority recommendation: Image optimization**

* **Why this matters:**
* **Largest Contentful Paint (LCP)** is often delayed due to large, unoptimized images.
* Images are typically the largest visible elements on a page, and slow loading directly affects user perception and engagement.
* Optimized images reduce bandwidth usage, improve load times, and enhance mobile performance.
* **How to implement:**
* Use modern formats like **WebP** or **AVIF** instead of JPEG/PNG.
* Apply **responsive image techniques** (srcset, sizes) to serve appropriately sized images for different devices.
* Implement **lazy loading** (loading="lazy") for offscreen images.
* Use tools like **ImageMagick**, **Squoosh**, or **Cloudinary** for compression.
* Serve images via a **Content Delivery Network (CDN)** to reduce latency.