|  |
| --- |
| CODEBLOGGS / LightHouse Analysis / **Home Page** |
| **PERFORMANCE** suggestions for improvement |
| Current note : **66/100** |

### Largest Contentful Paint (LCP):

* Issue: The largest contentful element (img#headbar-logo-img) takes 2,620 ms to paint within the viewport.
* Solution: Optimize the loading of this image by compressing it and serving appropriately sized images. Consider lazy loading or preloading to improve LCP.

### Reduce Unused JavaScript:

* Issue: Significant potential savings of 946 KiB by reducing unused JavaScript, with the bulk coming from various modules.
* Solution: Review and eliminate unnecessary JavaScript modules. Defer loading scripts until they are required, improving overall performance and reducing network activity.

### Minify JavaScript:

* Issue: Potential savings of 361.9 KiB by minifying JavaScript, including files from extensions and plugins.
* Solution: Minify JavaScript files to reduce payload sizes and script parse time, contributing to faster page loads.

### Preconnect to Required Origins:

* Issue: Potential savings of 100 ms by preconnecting to Google Fonts CDN.
* Solution: Add preconnect or dns-prefetch hints for important third-party origins like Google Fonts to establish early connections and improve page load speed.

### Preload Largest Contentful Paint Image:

* Issue: Potential savings of 50 ms by preloading the LCP image.
* Solution: If the LCP element (img#headbar-logo-img) is dynamically added, preload the image to improve LCP and overall page performance.

### Properly Size Images:

* Issue: Potential savings of 10 KiB by serving appropriately sized images.
* Solution: Resize and serve images in the correct dimensions to save data and improve load times, especially for the identified image (img#headbar-logo-img).

### Minify CSS:

* Issue: Potential savings of 403 KiB by minifying CSS, including Bootstrap and custom styles.
* Solution: Minify CSS files to reduce network payload sizes, contributing to faster page rendering.

### Serve Static Assets with an Efficient Cache Policy:

* Issue: Two resources found without a specified cache TTL (Time To Live).
* Solution: Set an efficient cache policy for static assets (js/bundle.js and media/CodeBlogg…1627d07….png) to speed up repeat visits.

### Reduce Unused CSS:

* Issue: Potential savings of 483 KiB by reducing unused CSS, including Bootstrap and custom styles.
* Solution: Remove or defer unused CSS rules to reduce network activity and improve overall page performance.

### Avoid Serving Legacy JavaScript to Modern Browsers:

* Issue: No potential savings, but consider adopting a modern script deployment strategy.
* Solution: Use modern JavaScript deployment strategies with module/nomodule feature detection to ship less code to modern browsers while retaining support for legacy browsers.

### Page Prevented Back/Forward Cache Restoration:

* Issue: Failure to enter back/forward cache due to WebSocket usage.
* Solution: Consider optimizing WebSocket usage or be aware that pages with WebSocket may not enter back/forward cache. Check for updates on browser support for WebSocket in back/forward cache.