

Audit Performance

Introduction:

The performance analysis was conducted on a competing application: todolistme.net. This application is difficult to compare with our application in the same state. The range of features differs from todolist.net to to-do-list-app. To-do-list-app is a basic application that consists of a single list of tasks to perform, whereas todolistme offers a system of multiple lists, categories and task planning. From a technical point of view, todolistme uses external libraries and includes third party resources, which is not the case with to-do-list-app. In addition, todolistme does not appear to have any updates in technology for the last six years, and therefore uses old outdated technologies and old APIs as well. Rather than scoring to-do-list-app against todolistme, we should rather audit todolistme's performance which will allow more precise view on what does not work and thus to formulate recommendations applicable to the to-do-list-app's application for its future developments.

Operation of the application:

Whether desktop or mobile, regardless of the quality of the device or connection, the performance of the site at first or second launch are problematic. The load far exceeds the window of 5 seconds. When the page is loaded, some events do not meet any responses within the 50 ms.

Recommendations:

Suggestions to improve the performance of the page:

1. The organization of the HTML document must be reviewed. `<Script>` tags are scattered throughout the document, which blocks the loading of the page. It would be wise to put them at the end of the document.
2. The number of nearly 70 requests is particularly high. We should reduce the overall number of requests. Several queries are devoted to loading images, but most of the images are used as icons but the use of icons in svg format would be much more relevant. Similarly, the background of the site is an image. The use of a solid colour through CSS properties would be more sensible.
3. It would also be possible to reduce the number of requests for loading. In fact, many of them involves advertisement that appear on the site. These have made the rendering path by monopolizing the main thread even before the main features of the site have been loaded and can be used. The use of the "defer" attribute in the `<script>` tag would allow the execution of these requests to be postponed at the end of the page loading.

4. Caching some resources would also improve the loading of the page during subsequent visits by the user.
5. The application uses iframes that have an impact on performance but also on security. In the case of share buttons on social networks, a clickable link and an asynchronous script are now recommended in the documentation.

These recommendations could improve performance but cannot be exhaustive. The implementation of Real User Monitoring tools, code-splitting, critical rendering path analysis, the use of a CDN are all tracks that could be dug to improve the performance of this site and be thought of in the future. development of our application.