

ABSTRACT

This study aims to enhance the performance of a React-based web application that utilizes Client-Side Rendering (CSR) by applying two optimization techniques: code splitting and lazy loading. The main issue addressed is the low Lighthouse performance scores, which indicate by slow initial loading times and large JavaScript initial bundle sizes, especially on pages with complex visual elements. To address these challenges, the research adopts a structured methodology consisting of initial performance analysis, identification of performance problem, implementation of code splitting and lazy loading strategies, and post-optimization testing. Development was conducted using Visual Studio Code, and performance evaluation was carried out with Google Lighthouse, Webpack Bundle Analyzer, and browser developer tools. The application used for this study includes several core pages such as Login, Data Analysis, Data Account, Data Income, and Data Menu. The results demonstrate significant improvements: over 80% reduction in bundle size, up to 80.66% faster loading times, and performance score increases ranging from 19% to 70.8% across lighthouse metrics including First Contentful Paint (FCP), Largest Contentful Paint (LCP), Total Blocking Time (TBT), and Speed Index. These findings confirm code splitting and lazy loading strategies can effectively enhance the responsiveness and efficiency of CSR-based web applications. This research contributes practical insights for front-end performance optimization and offers a scalable solution applicable to modern web development practices.

Keywords: *client-side rendering, code splitting, lazy loading, performance optimization, react, web application.*