

## **ABSTRACT**

*The choice of user-interface rendering strategy in Laravel web applications determines efficiency and user-experience quality, especially when the system must display large static datasets. This study evaluates performance degradation on the document page of the FRI Online Session (SOFI) application, which currently renders with Blade server-side rendering (SSR), where latency and interactivity decrease as session records grow to thousands. The objective is to empirically compare three strategies: Blade-SSR and React and Vue client-side rendering (CSR) via Inertia.js. A controlled experiment was applied to three backend-identical system versions under data scenarios ranging from 10 to 10 000 records. Google Lighthouse measured First Contentful Paint (FCP), Largest Contentful Paint (LCP), Speed Index (SI), and Total Blocking Time (TBT). Results show that Blade excels for small-to-medium datasets with FCP 0.9–1.7 s, but its TBT rises sharply to 2 790 ms at 10 000 records. React and Vue keep low TBT at large scales; Vue is marginally more efficient than React. Triplicate measurements in each scenario verified data reliability. Findings highlight a clear trade-off between initial load speed and interactive responsiveness. A hybrid scheme is recommended: SSR for the first paint, CSR for subsequent client interactions to preserve scalability and resources. The evidence guides higher-education institutions operating large academic systems in selecting a frontend architecture. Future work should investigate hybrid frameworks such as Next.js and Nuxt.js for further optimisation and enhanced user experience.*

**Keywords:** *Laravel, server-side rendering, client-side rendering, Blade, React, Vue*