Faculty of Electrical Engineering
Warsaw University of Technology

Flutter vs. React for Web Applications: A Comparative Benchmark Study of Performance Metrics and Developer Experience

Author: Jakub Witczak
Supervisor: dr hab. inż. Michał Śmiałek

INTRODUCTION

Amidst the rapid growth in device types, the need for cross-platform development tools has increased. This comparative study evaluates Flutter and React specifically for web application development, moving beyond Flutter's typical association with mobile platforms. The primary objective was to benchmark multiple performance metrics, such as build size and speed index, and analyse developer experience factors to determine if Flutter is a suitable and practical choice for building web applications. The research aimed to provide a comprehensive evaluation, considering both quantitative data and qualitative insights into the development process.

METHODOLOGY

For a valid comparison, two functionally identical applications were developed: one using Flutter and the other using React with TypeScript. Both applications featured a schema management panel with CRUD (Create, Read, Update, Delete) operations, a side navigation system, form handling, and consistent UI components inspired by the requirements of the DoorCE (Data Opener Central Europe) open data project led by the Warsaw University of Technology. Clean Code principles and framework-specific best practices were followed in both implementations. Quantitative evaluation focused on build size (total compiled application size in KB) and Speed Index (how quickly content becomes visible, measured in seconds). Google's Lighthouse tool was utilized for performance assessment, including Speed Index. Developer Experience (DX) was assessed qualitatively by comparing the setup process, perceived development velocity, UI construction (widgets vs. components), state management implementation, debugging effectiveness (including hot reload/restart), and the quality of documentation and community resources encountered. All tests were performed under controlled conditions on identical hardware and repeated three times to ensure reliability.

PERFORMANCE BENCHMARKS

The quantitative benchmarks revealed significant performance differences favouring React for web deployment in this context. The React application's average build size was substantially smaller at 1.23 MB compared to Flutter's 29.4 MB (Tab. 1). This disparity is primarily attributed to Flutter's architecture requiring the bundling of its rendering engine (like CanvasKit or the newer Skwasm) alongside the application code to draw Ul elements. In contrast, React uses a Virtual DOM to efficiently determine changes, but ultimately renders to the browser's native DOM for display. Although Flutter is developing technologies like Skwasm (using WebAssembly) to reduce bundle size, compatibility issues (requiring fallback renderers) still contribute to larger builds currently. Consequently, the Speed Index also favoured React, which displayed visual content considerably faster (average of 0.4 seconds) than the Flutter web application (average of 0.9 seconds) (Tab. 2). Results suggest a better-perceived loading experience for users accessing the React version.

Tab. 1. Build size comparison

Framework	Average Build Size
React	1.23 MB
Flutter (Web)	29.4 MB

Tab. 2. Speed Index comparison

Framework	Average Speed Index
React	0.4 s
Flutter (Web)	0.9 s

DEVELOPER EXPERIENCE (DX)

In a comparative study between Flutter and React for web apps, these factors were assessed based on the experience of building identical applications. React was found to feel familiar due to its alignment with web development concepts. Benefits were noted from extensive community resources and simpler state management with Zustand. In contrast, Flutter offered a more integrated debugging suite (DevTools) and built-in design systems. However, its widget-based approach and state management patterns (like BLoC) were found to represent a different development paradigm requiring adaptation. Ultimately, development efficiency and maintainability are significantly impacted by the Developer Experience.

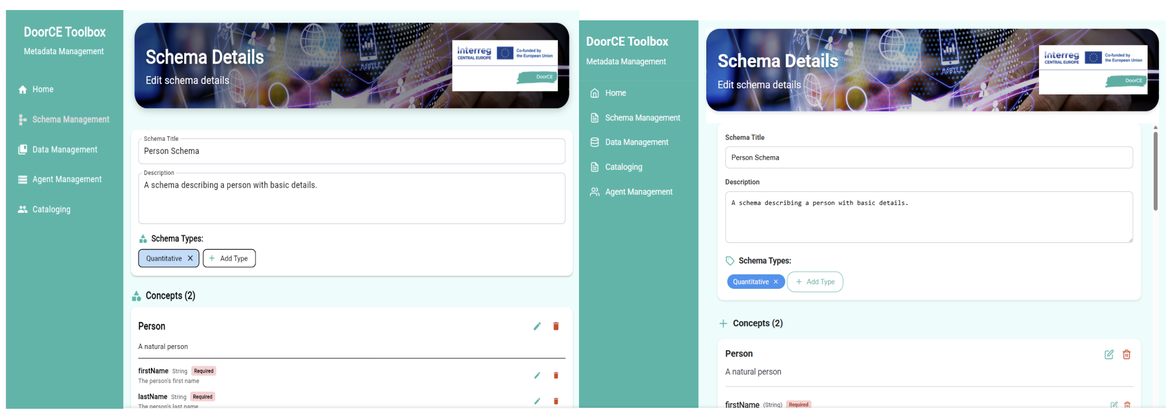

Fig. 1. Side-by-side comparison of the "Schema Details" view in the Flutter application (left) and the React application (right)

CONCLUSION

In conclusion, for the specific requirements of this web-based application, React exhibited superior performance characteristics critical for a positive user experience on the web, namely smaller build sizes and faster loading times. While Flutter offers the advantage of a unified codebase for cross-platform development, its current web implementation showed significant drawbacks in these benchmarks.