

Scenario 1: Product Administrator App

Considering the complexity and need for maintainability, a modern front-end framework like **React** or **Vue.js** would be appropriate because these frameworks provide component-based architecture, which makes the code more modular, easier to maintain, and scalable for adding new features.

React's extensive ecosystem would be particularly useful for managing multiple views and handling the large amount of data and forms present in the app.

Scenario 2: Huge Data Table in Web App

Given the requirement to handle hundreds of thousands of data records, a robust framework is essential to ensure performance and efficient data handling.

My choice would be Angular, because it would be a suitable choice due to its strong support for building complex, data-driven applications.

According to my research, Angular's powerful data binding, dependency injection, and comprehensive tooling ecosystem, combined with a data grid library like **AG Grid**, can handle large datasets efficiently and provide features such as virtual scrolling, ensuring smooth performance and a responsive user experience.

Scenario 3: Design System Prototype

For creating a prototype that showcases the product team's design system, a framework focused on rapid development and component reusability is essential.

Vue.js is an excellent choice due to its simplicity and flexibility in building component-based UIs. Vue's single-file components allow for modular and maintainable code, making it easy to showcase various design elements like colors, components, fonts, and styling.

From my research about it, using **Vue's ecosystem** tools, such as Vue CLI and Vue Devtools, can enhance the development experience and ensure a polished prototype.

Advantages on using React to build the myFlix App

Using React to build the myFlix app offers significant advantages, including a component-based architecture that promotes reusability and maintainability of code.

React's virtual DOM ensures efficient rendering and updating of the user interface, enhancing performance and user experience.

Additionally, React's rich ecosystem, including tools like React Router for state routing and Redux for state management, provides robust solutions for building dynamic and interactive single-page applications.