React Native for Web:
Extension of React Native that allows you to use React Native components and APIs to build web applications. It tries to make your React Native app run on web browsers using the same codebase. With Expo, you can leverage react-native-web to create a web version of your app. Expo has built-in support for this, making it relatively straightforward to deploy your app to the web. Downside is that not all native modules and features available in React Native/Expo are compatible with the web.

Details on React Native compatibility:
https://necolas.github.io/react-native-web/docs/react-native-compatibility/

How to Make Your React Native Apps Work on the Web:
https://retool.com/blog/how-to-make-your-react-native-apps-work-on-the-web

React Framework:
If we use an entirely new codebase we can use React. Using React and React Native allows you to maintain a consistent look, feel, and user experience across your web and mobile applications. Transitioning from React Native to React is fairly simple since they share similarities. It also includes hot reloading which is helpful when it comes to workflow.

Other Frameworks:
Other possible frameworks for a web app in this scenario can include Vue.js and Angular. Vue focuses on an easy-to-understand reactive data model and offers an ecosystem that can scale between a library and a full-featured framework. Vue's ecosystem provides tools and libraries to handle routing, state management, and building larger scale applications. Angular is often seen as a more robust solution, providing a strong opinionated framework that includes a wide range of features like dependency injection, templating, AJAX handling, and component-based architecture. This can make Angular appear more complex and have a steeper learning curve, but it’s well-suited for enterprise-scale applications due to its powerful tooling and integrated practices that address various development challenges.