First of all, I want to show some key features of those cross-platform mobile application development:

Ionic:

- Uses web technologies like HTML, CSS, and JavaScript.
- You can base your project on such development network as React, Vue or Angular
- A lot of pre-build UI components which are cross-platform, and you don't need to rewrite them
- Possible to use native phone elements as camera, geolocation etc., but here is also disadvantage, because it can happen conflict between them.

React Native:

- Uses React framework and JavaScript code'
- You can work on native UI components on both iOS and Android
- It is possible to implement it to already created project, not only create application from scratch.
- Quick iteration cycles

Flutter:

- Uses Dart programming language developed by Google.
- Fully customizable UI components with rich widget set.
- Fast performance with native compilation.

Cordova:

- Uses HTML, CSS, and JavaScript.
- Plugins to access native device features like camera, GPS, etc.
- Applications built in Cordova can be a little slower if you need to add a lot of data and functionality.
  Hard to optimize.

Kirigami:

- Built on Qt framework for KDE desktop apps.
- Lightweight set of QtQuick components tailored for mobile.
- Seamless integration with KDE ecosystem.

In summary, Ionic, Flutter and React Native provide the most complete cross-platform frameworks with customizable UI components while Cordova offers a thinner wrapper for web apps. Kirigami is a niche option best for KDE integration.

My Choice of Development Framework

I will build my choice based between three most powerful frameworks, that uses big companies and that have a huge functionality: Ionic, Flutter and React Native.
Let's compare Community support (based on StackOverflow and Google Trends): on the first place is Flutter, second is React Native and the third is Ionic. For me it is especially important, because I will develop my first application, and if I need help, I will ask community about it. It touches also option about number of code examples and how many supporters' certain framework has. But it is important to mention that on GitHub flutter has more starts on the main repository. It is because flutter uses Dart, that is not used in such many fields as JavaScript.

Second criteria are performance. Now React Native has the same performance level as a Flutter then Ionic is a little bit slower. But I should also mention that React Native runs a risk of lower performance, because of JavaSript VM and the way it communicates with the Native Core.

I see on using of Ionic little bit unnecessary, because if I will use Ionic, I will build application based on React regardless, but if I want to use React as UI framework, then React Native is just a better alternative due to difference in performance and user interface. I see the point to use Ionic for that project, because Ionic is very easy to learn, you can use framework, that suits for you (for example if you want to build application on Angular or Vue), it is quite popular and other advantages, but for me person who will use React- React Native is better.

I would call third criteria as "Difficulty to start". For me personally will be easier to start developing on React, because I use it in my developers every day. I use his syntax in my Next development, so if compare language behind frameworks, I will choose JavaScript in React Native. I don't need to learn Dart, that will be quite new for me.

As I found, Flutter stays for more functionality, as React Native is simpler. For example, React Native has 25 components as building blocks, then Flutter has 21 components just for animations. But here is also important to tell, that React Native can provide beautiful UI using already built React libraries.

Difference in rendering solution: React uses iOS and Android native widgets, then Flutter uses Skia engine for UI rendering, that is used in Chrome, Firefox and Android. Skia gives more flexible and more precise UI.

I term of my project, that is quite simple: Sudoku with database and few extra features there is no need to make difficult, and we shouldn't create a some kind of a difficult UI here.

If it was difficult project with a lot of work from the UI side. And if optimization would be a huge point for this project (as I described earlier React Native can has some issues), I would use a Flutter, but Sudoku game is quite simple, and I think React Native passes for all requirements here very good. Based on the project requirements I have, I would use React Native as the framework to build the Sudoku app. Here are a few reasons why: I already have experience with React which will make it easier to learn React Native. The core concepts are very similar, and I know them. Development will move more quickly with React Native's robust ecosystem and abundance of reusable user interface components. It provides native performance and look/feel on both iOS and Android using the same JavaScript codebase. Flexible options for local storage and networking possibilities to connect to a server backend. Strong support for handling multiple languages/localizations. Lots of documentation and examples are available online. While Flutter is also a solid option, the familiarity with React makes React Native the lowest friction path to build this app. And it has all the capabilities needed like customizable native UI, offline storage, networking, multi-language support, etc. Here with React Native I can focus on functionality, but not on learning the whole new framework.