**Flutter**

1. Flutter Framework

Flutter is an open-source UI toolkit developed by Google for building natively compiled applications for mobile, web, and desktop from a single codebase. It allows developers to create high-performance, visually appealing, and responsive applications using a single language (Dart) and a single codebase, which can be compiled to run on multiple platforms including Android, iOS, web, and desktop.

Some key features of Flutter framework include:

* Hot Reload:

Flutter offers a hot reload feature, which allows developers to see the changes made in the code in real-time without having to restart the entire application.

* Widgets:

Flutter uses a declarative UI programming model where the user interface is built using widgets, which are the basic building blocks of a Flutter application. Flutter provides a rich set of pre-built widgets for creating different types of UI elements such as buttons, text fields, images, etc., as well as the ability to create custom widgets.

* Fast Performance:

Flutter apps are compiled to native code, which enables them to achieve high performance close to native applications. Flutter also includes a built-in rendering engine, which allows for smooth animations and transitions.

* Cross-platform Development:

With Flutter, developers can write code once and run it on multiple platforms, including Android, iOS, web, and desktop. This makes it a cost-effective and efficient choice for cross-platform app development.

* Material Design and Cupertino Design:

Flutter provides two sets of pre-designed UI components - Material Design for Android apps and Cupertino Design for iOS apps - that help in creating apps with a consistent and native look-and-feel on respective platforms.

* Integration with Backend Services:

Flutter provides built-in libraries and plugins for integrating with various backend services such as Firebase, REST APIs, GraphQL, and more, making it easy to fetch data from external APIs and connect with other services.

* Supportive Community:

Flutter has a large and active community of developers, which provides extensive support, documentation, and resources for learning and troubleshooting. The community also contributes to the development of libraries and plugins, which extends the functionality of Flutter applications.

1. flutter installation and configuration
2. Widget

In Flutter, widgets are the basic building blocks of a user interface. They are used to define the layout, appearance, and behavior of the UI components in a Flutter application. Widgets in Flutter are highly customizable and can be combined together to create complex and dynamic UIs. There are two types of widgets in Flutter:

* Stateless Widgets:

These are immutable widgets that do not store any mutable state. They are typically used for parts of the UI that do not change over time, such as static UI elements like buttons, labels, and images. Stateless widgets are created using the StatelessWidget class and are implemented by overriding the build() method, which returns the UI components based on the given input properties.

* Stateful Widgets:

These are mutable widgets that store mutable state. They are typically used for parts of the UI that can change over time, such as user input fields, animations, and dynamic UI elements. Stateful widgets are created using the StatefulWidget class, which has an associated mutable state object created using the createState() method. The mutable state object is then used to store and manage the changing data for the widget.

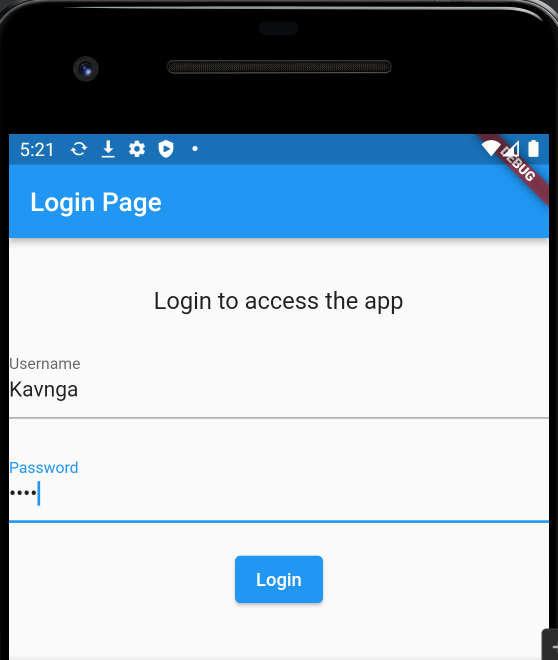
Chart, funnel chart

Description automatically generated

Flutter Widget Tree

1. pass data between components.
2. api bindings
3. Login/logout feature
4. Role based access control
5. Need a sample code for 3-6

Graphical user interface, application

Description automatically generated

Login as Kavinga Logged in as Kavinga

Graphical user interface, application

Description automatically generatedGraphical user interface, application

Description automatically generated

Login as Sam Logged in as Sam