# CROSS PLATFORM MOBILE APPLICATION DEVELOPMENT WITH FLUTTER

MINAB
IT SOLUTION

Hahu Jobs

# ??

# NEBIL MOHAMMED

DEVELOPER & UI/UX DEISGNER

https://nebas.design

# flutter 1 of 2 NOUN

flut·ter ˈflə-tər

fluttered; fluttering; flutters

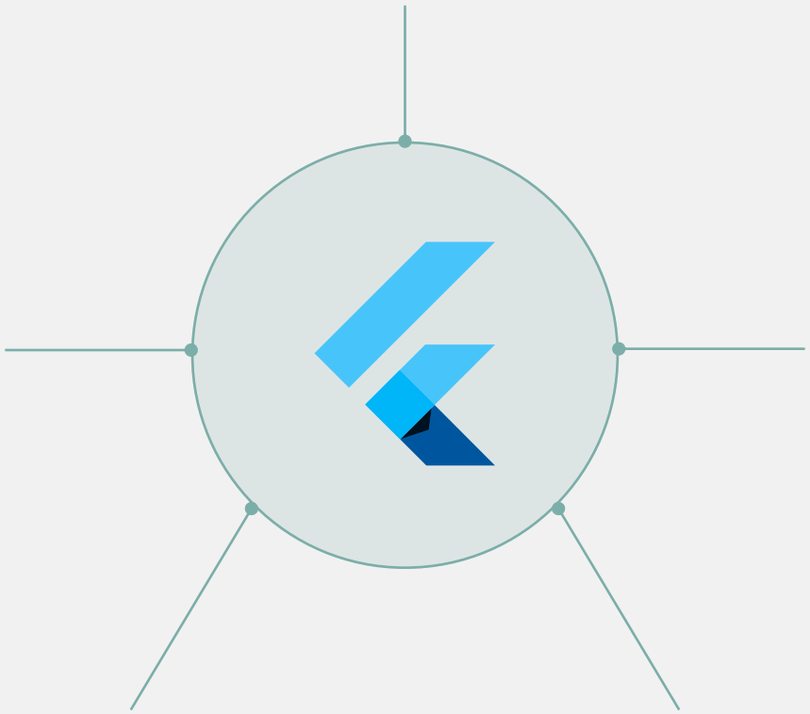
- Flutter is an open-source UI software development kit created by Google
- It allows developers to build high-performance, visually attractive, and natively compiled mobile applications for both iOS and Android platforms using a single codebase.

https://flutter.dev

# FEATURES!

## CROSS PLATFORM

Flutter enables developers to create applications for both iOS and Android platforms using a single codebase.



## NATIVE PERFORMANCE

Flutter applications are compiled to native code, ensuring they run as fast as native apps.

## FAST DEVELOPMENT

Flutter's hot reload feature allows developers to see the changes in the code instantly, speeding up the development process.

## OPEN-SOURCE

Flutter is an open-source framework, allowing developers to contribute to its development and access a wide range of community resources.

## FLEXIBLE UI

Flutter provides a rich set of pre-built and customizable widgets that help create beautiful and responsive user interfaces.

https://flutter.dev

# APPS

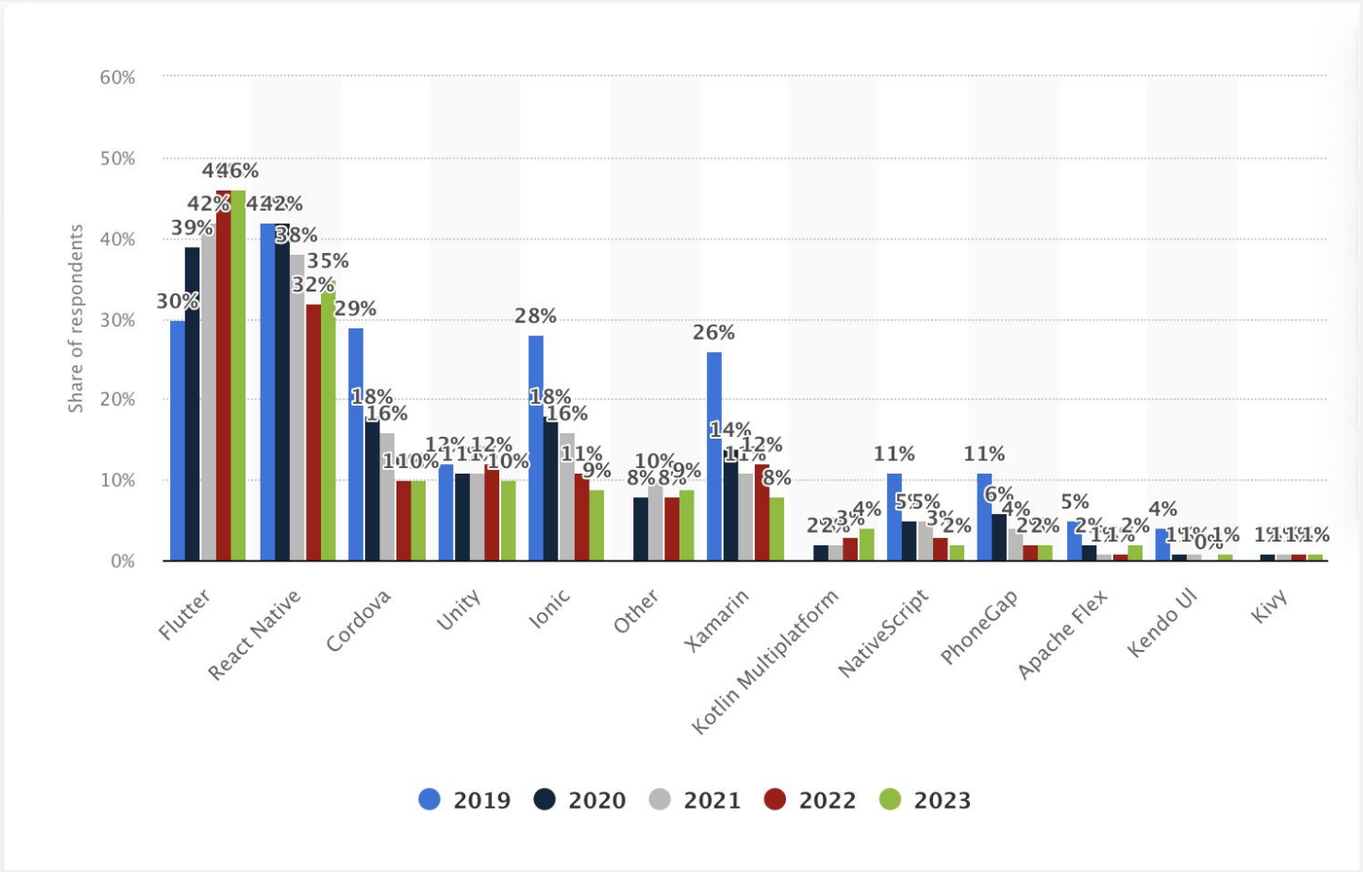
*As of January 2023, there are over 700,000 apps in the Play Store that are built with Flutter, and one in five new apps on the Play Store use Flutter, more than all other cross-platform frameworks put together*

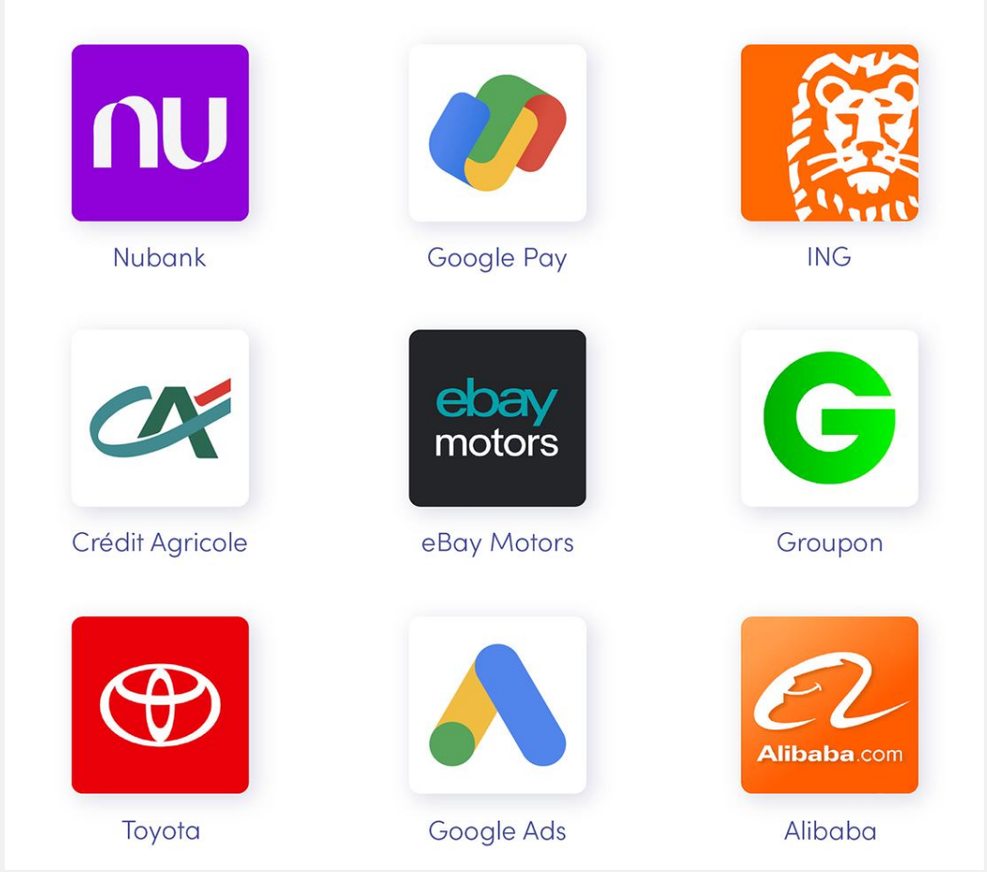
-verygood.ventures

**46%**

*of software developers used Flutter.*

- statista




https://www.miquido.com...

## EVERY THING IS A WIDGET!

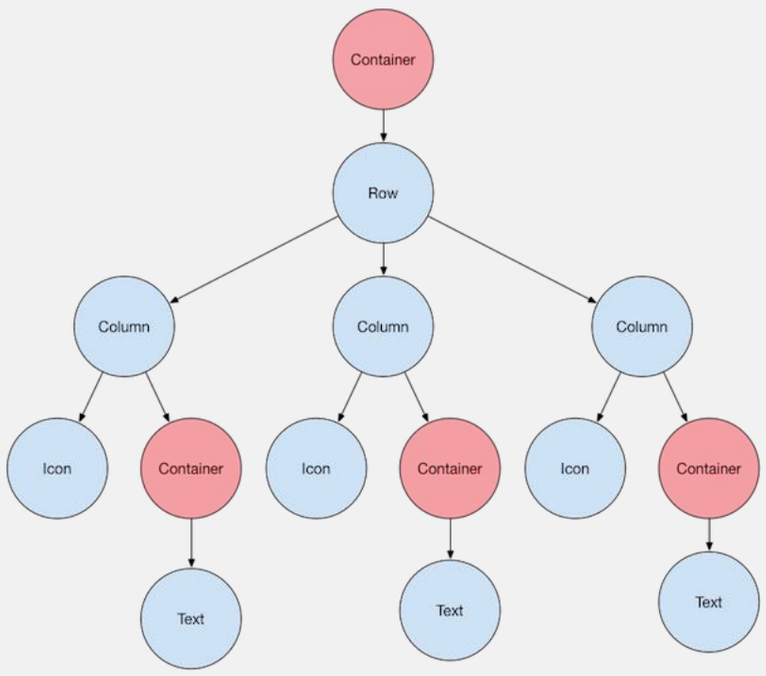
Widgets are used to describe the structure and layout of your application's UI elements, such as buttons, text fields, images, containers, and more.

# WIDGET?

## LAYOUT WIDGET

Flutter's layout widgets provide a flexible and powerful way to **organize** and **position** the **UI elements** within your application.

These widgets, such as Row, Column, Stack, and Expanded, allow you to create complex and responsive layouts with ease, enabling you to build visually appealing and intuitive user interfaces.




app.dart

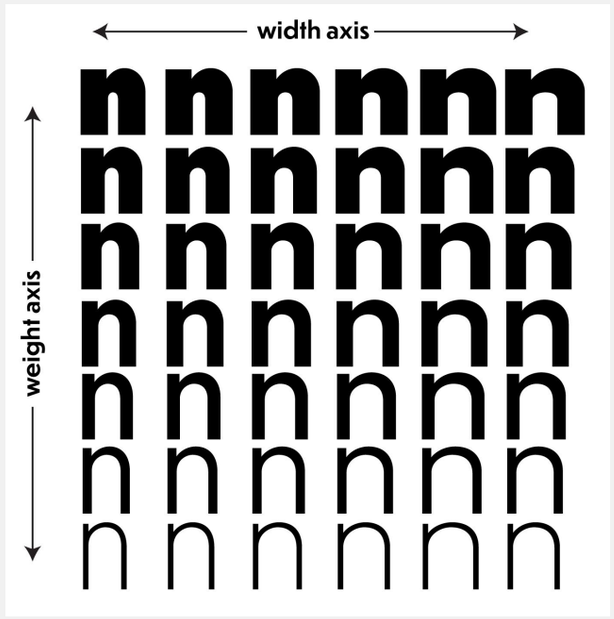
```dart
class MyApp extends StatelessWidget {
  const MyApp({super.key});
  @override
  Widget build(BuildContext context) {
    const String appTitle = 'Flutter layout demo';
    return MaterialApp(
      title: appTitle,
      home: Scaffold(
        appBar: AppBar(
          title: const Text(appTitle),
        ),
        body: const Center(
          child: Text('Hello World'),
        ),
      ),
    );
  }
}
```

# WIDGET?

## STYLE WIDGET

Flutter's style widgets, such as Text, Icon, and Container, allow you to customize the appearance of your UI elements.

These widgets provide a vast array of visual properties, including font styles, colors, sizes, and more, enabling you to create a cohesive and visually consistent user experience across your application.




app.dart

```
Theme(
  // Find and extend the parent theme using
`copyWith`.
  // To learn more, check out the section on
`Theme.of`.
  data: Theme.of(context).copyWith(
    colorScheme: ColorScheme.fromSeed(
      seedColor: Colors.pink,
    ),
  ),
  child: const FloatingActionButton(
    onPressed: null,
    child: Icon(Icons.add),
  ),
);
```

# WIDGET?

## ANIMATION WIDGET

Flutter's animation widgets, including AnimatedContainer, AnimatedOpacity, and AnimatedSize, enable you to create smooth and engaging transitions between UI states.

These widgets abstract away the complexity of manual animation implementation, allowing you to focus on defining the desired animation behavior and concentrate on building a delightful and responsive user interface.

app.dart

```
{
 home: Scaffold(
   appBar: AppBar(
     title: const Text(appTitle),
   ),
   body: const Center(
   Hero(
   tag : “Text”
     child: Text('Hello World'),
   ),
   ),
  ),
 );
}
```

# TYPE OF WIDGET

## STATELESS WIDGETS:

- Stateless widgets are immutable, meaning their properties and appearance cannot change during the lifetime of the widget.
- They are useful for widgets that do not require any internal state management, such as simple UI elements like icons, text, or images.
- Stateless widgets are lightweight and efficient, as they are built and rendered only once and do not require rebuilding during their lifetime.

app.dart

```
class HelloWorld extends StatelessWidget {
  const HelloWorld({super.key});

  @override
  Widget build(BuildContext context) {
    return Container();
  }
}
```

# TYPE OF WIDGET

## STATEFUL WIDGETS:

- Stateful widgets are mutable, meaning their properties and appearance can change dynamically in response to user interactions or other events.
- They maintain an internal state that can be updated, leading to a rebuild of the widget and its descendants.
- Stateful widgets are useful for widgets that need to handle user input, display real-time data, or manage complex UI states.
- Examples of stateful widgets include forms, buttons, and interactive components.

app.dart

```
class HelloWorld extends StatefulWidget {
  const HelloWorld({super.key});

  @override
  State<HelloWorld> createState() =>
_HelloWorldState();
}


class _HelloWorldState extends State<HelloWorld> {
  @override
  Widget build(BuildContext context) {
    return  Container();
  }
}
```

## RIVERPOD

Riverpod is a powerful state management solution for Flutter that provides a simple and scalable approach to managing application state. It is built on top of the Provider package and introduces several improvements, such as better testability, improved performance, and seamless integration with other Flutter packages.

*flutter_riverpod: ^2.5.1*

## GRAPHQL

GraphQL Flutter is a powerful library that allows you to seamlessly integrate GraphQL APIs into your Flutter applications. GraphQL is a modern, efficient, and flexible data query and manipulation language that provides an alternative to traditional REST APIs.

*graphql_flutter: ^5.1.2*


https://github.com/itismrx/rick_and_morty

**Assignment**

- Add a search functionality
  - Filter by species, status, Location
- Add a detail page

# THANK YOU