

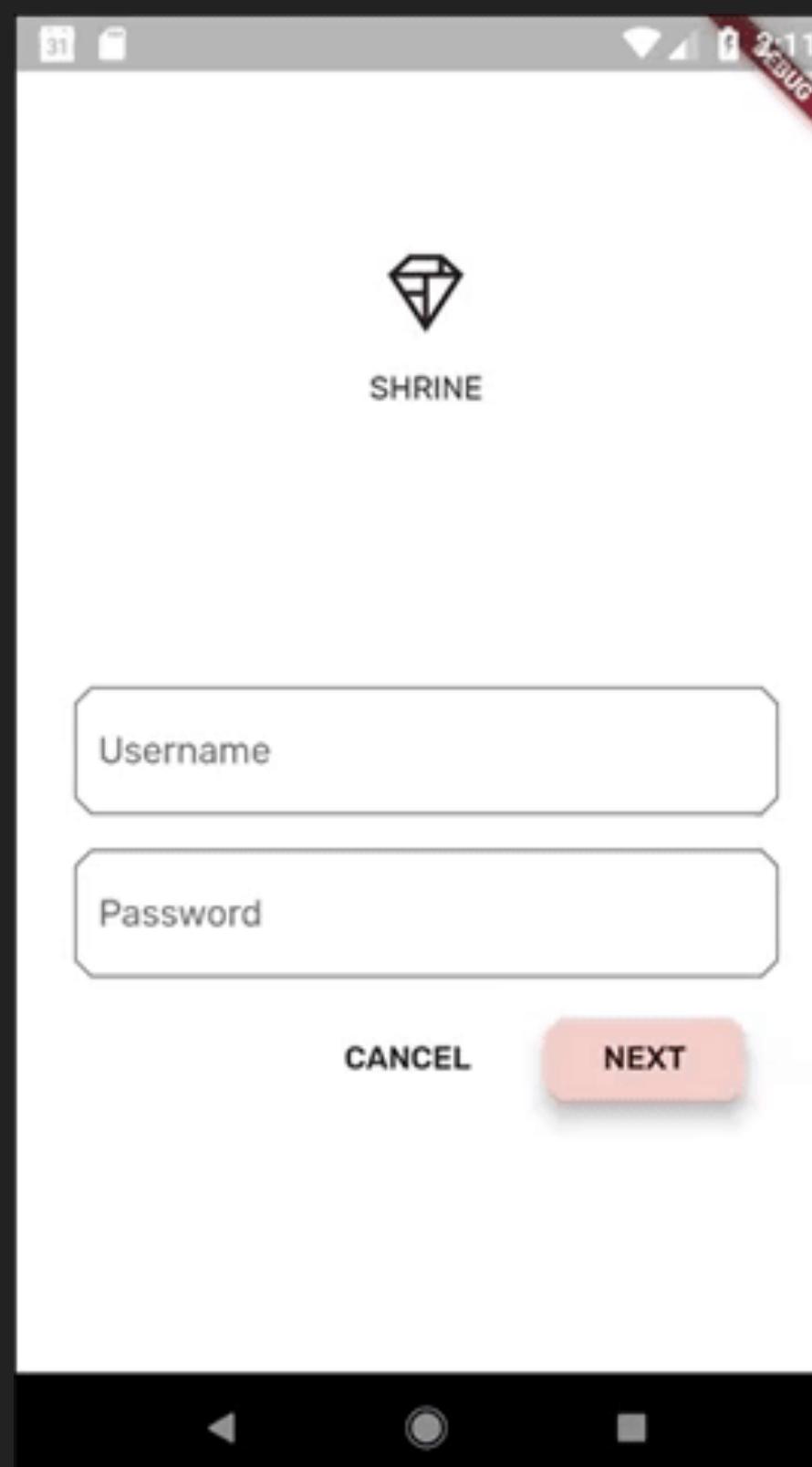
АЛЕКСЕЙ ПАНОВ

ТВОЕ ПЕРВОЕ ПРИЛОЖЕНИЕ
НА FLUTTER

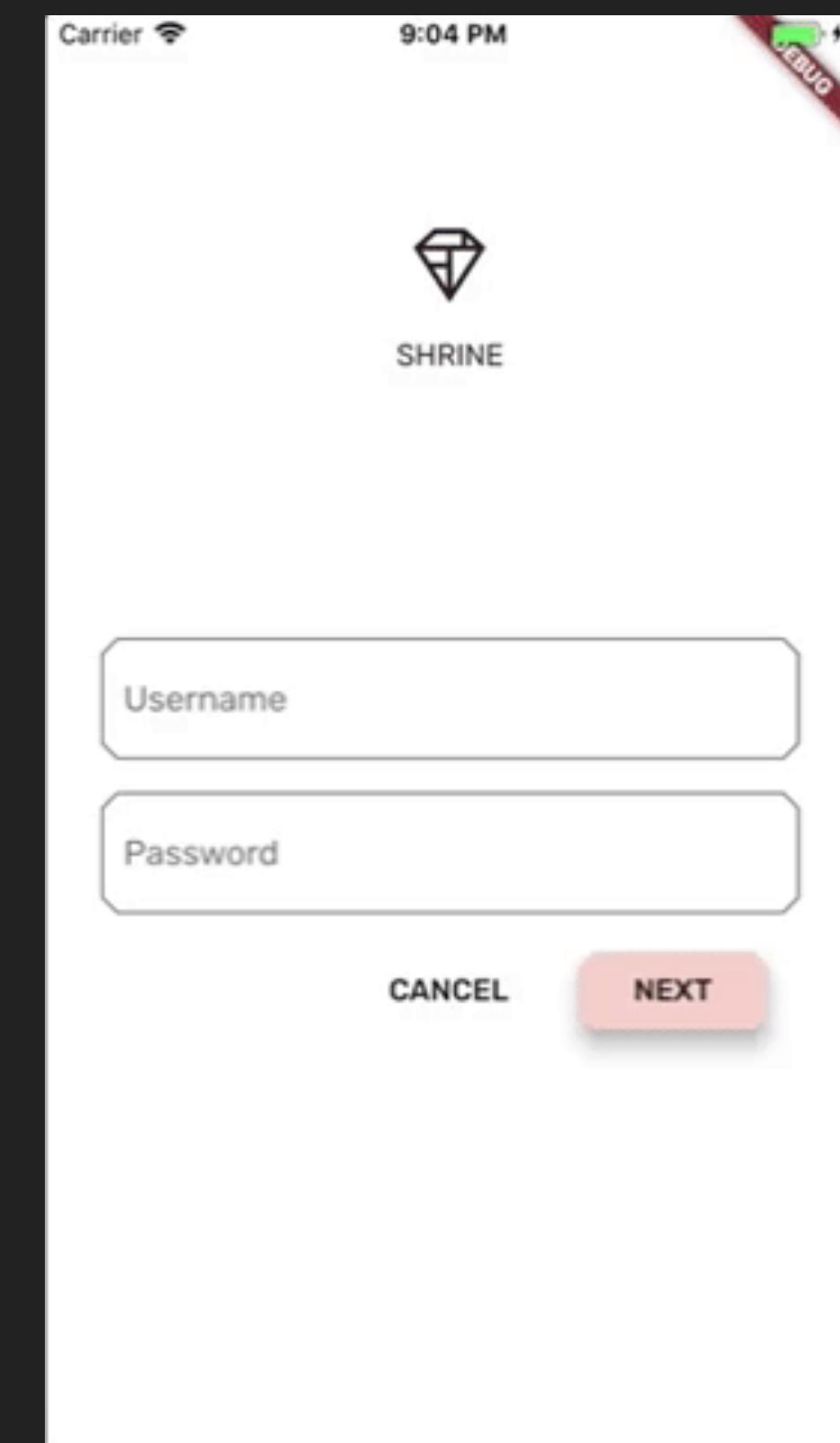
О ЧЕМ ДОКЛАД?

- ▶ Краткий обзор Флаттера
- ▶ Практическая часть

ЧТО ТАКОЕ FLUTTER?

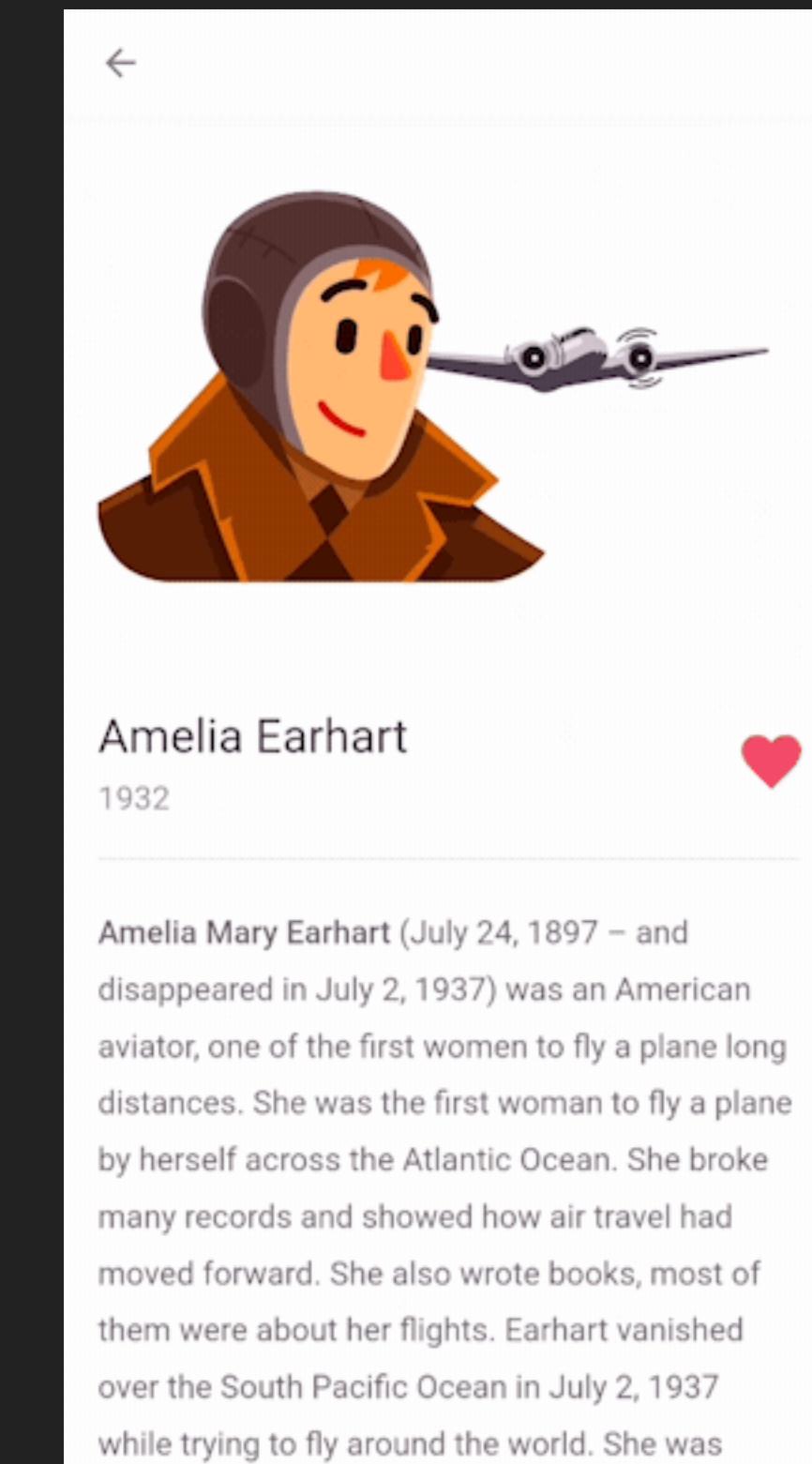
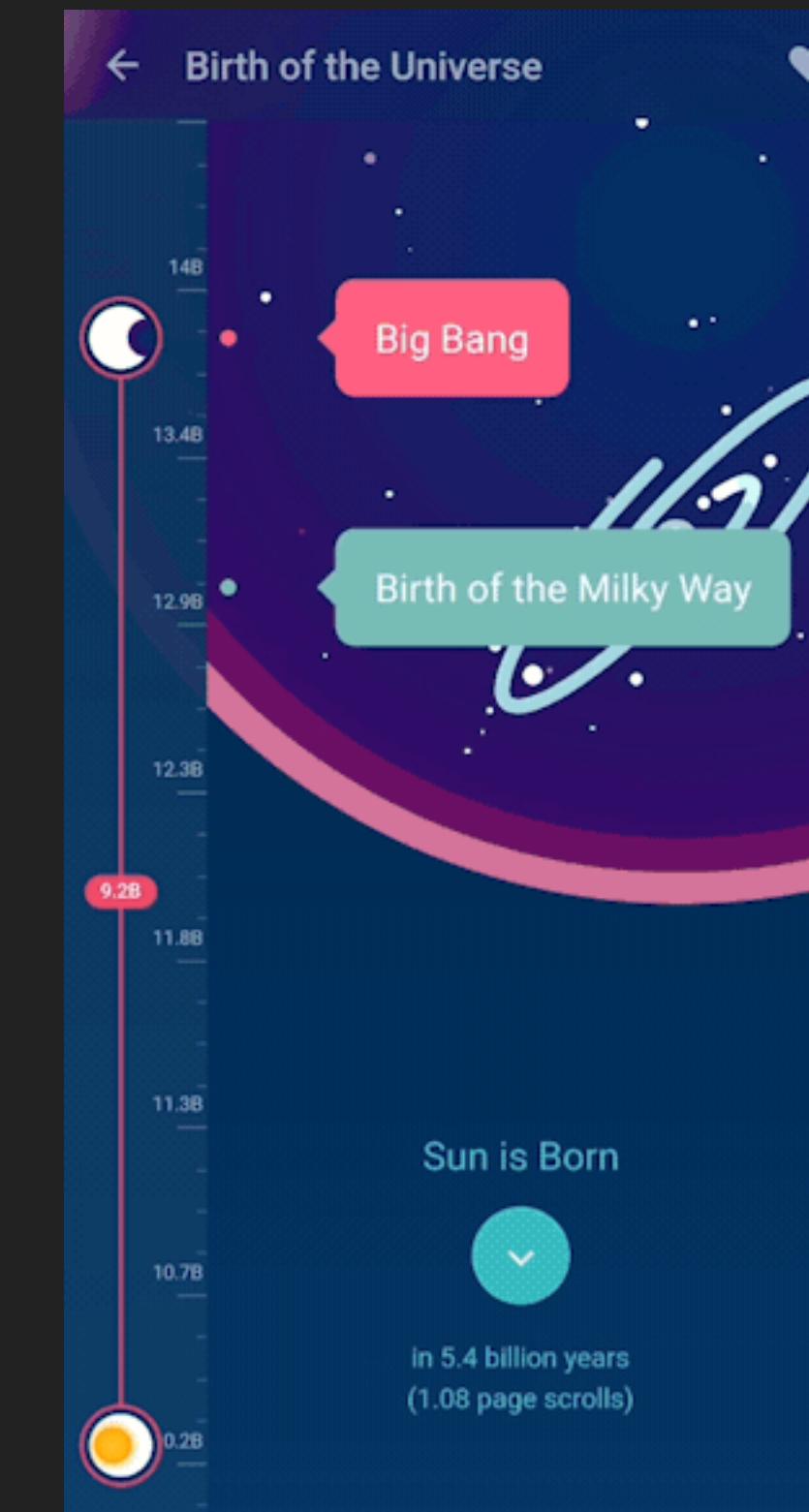
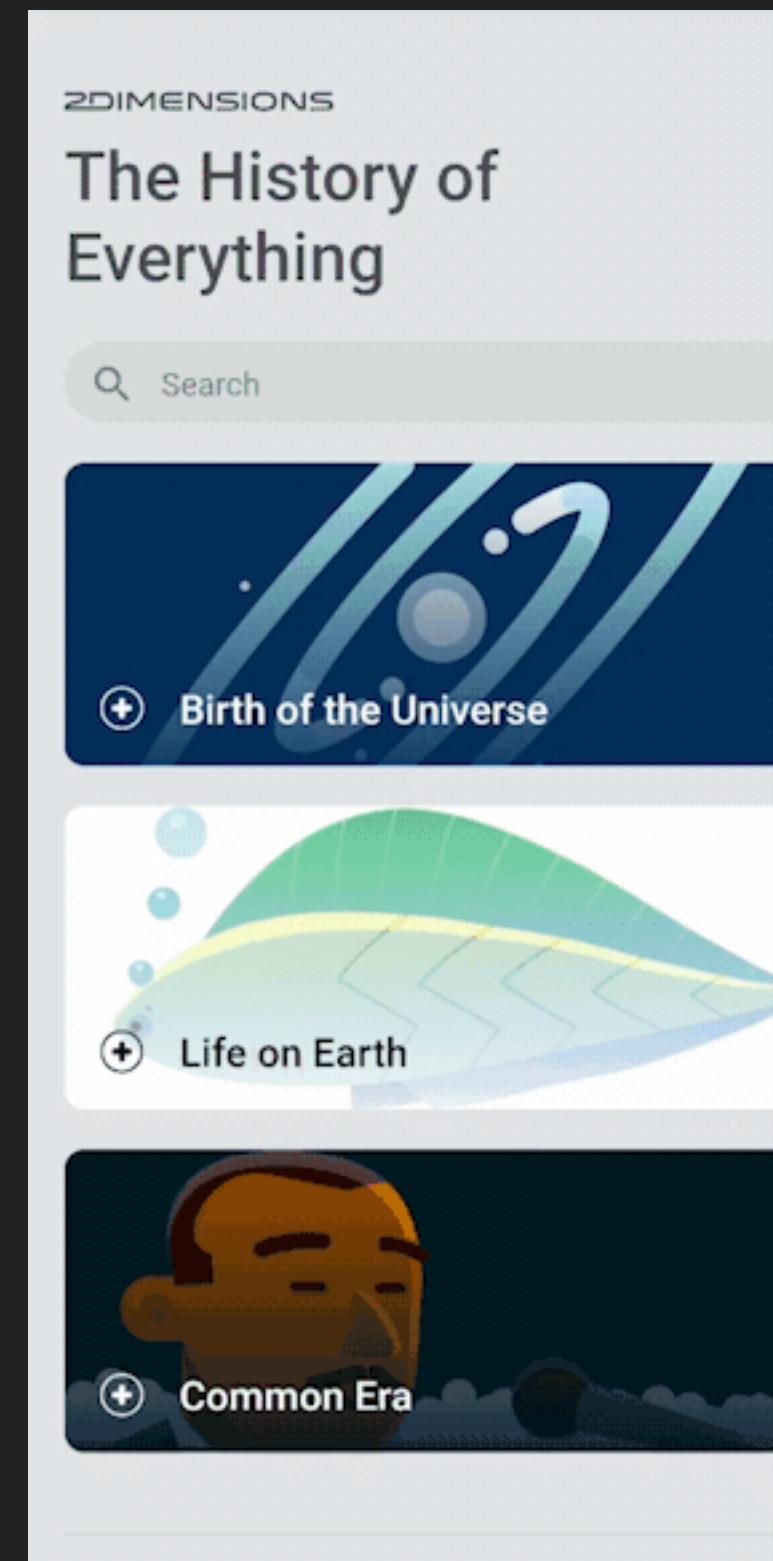


Android



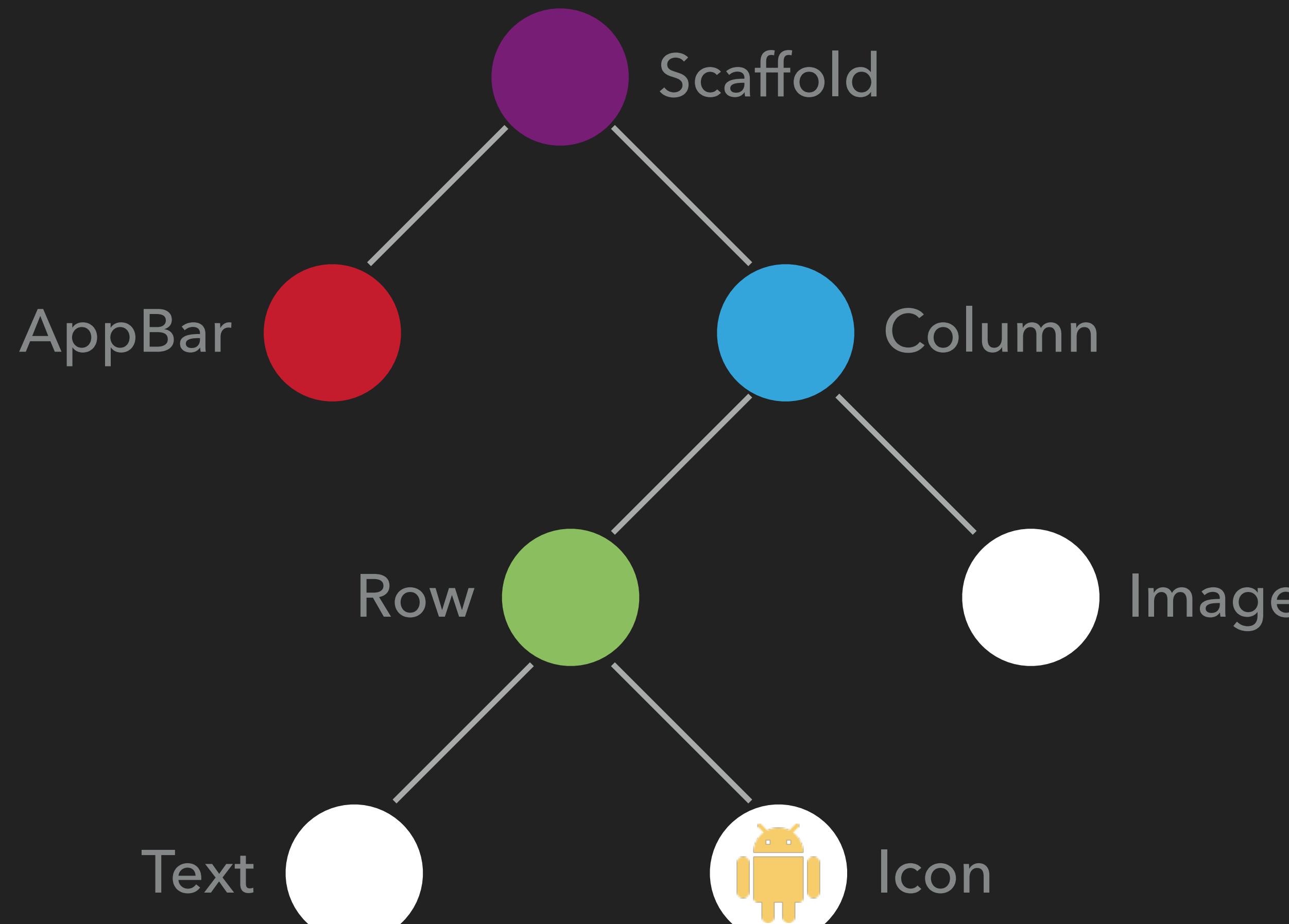
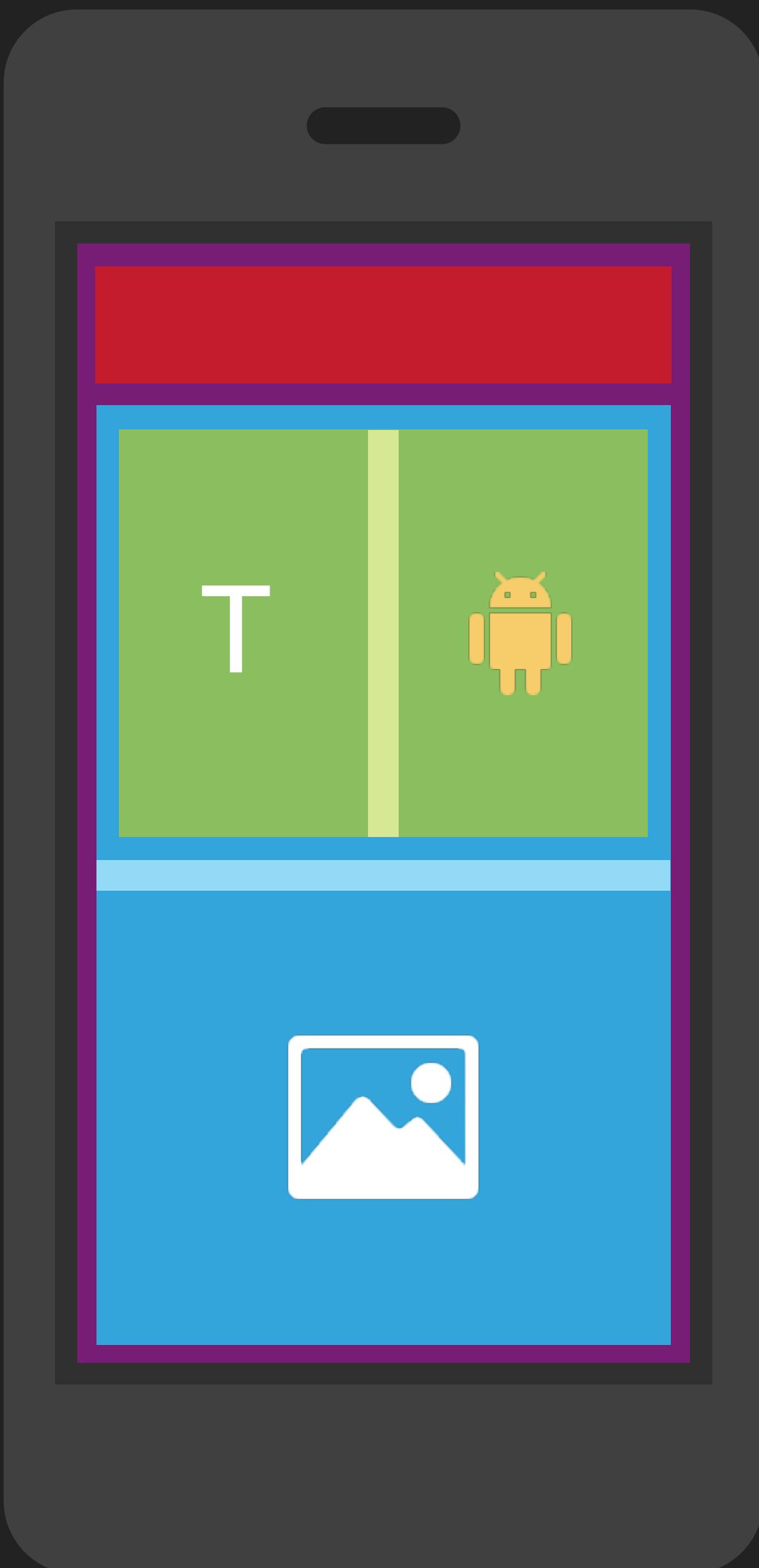
iOS

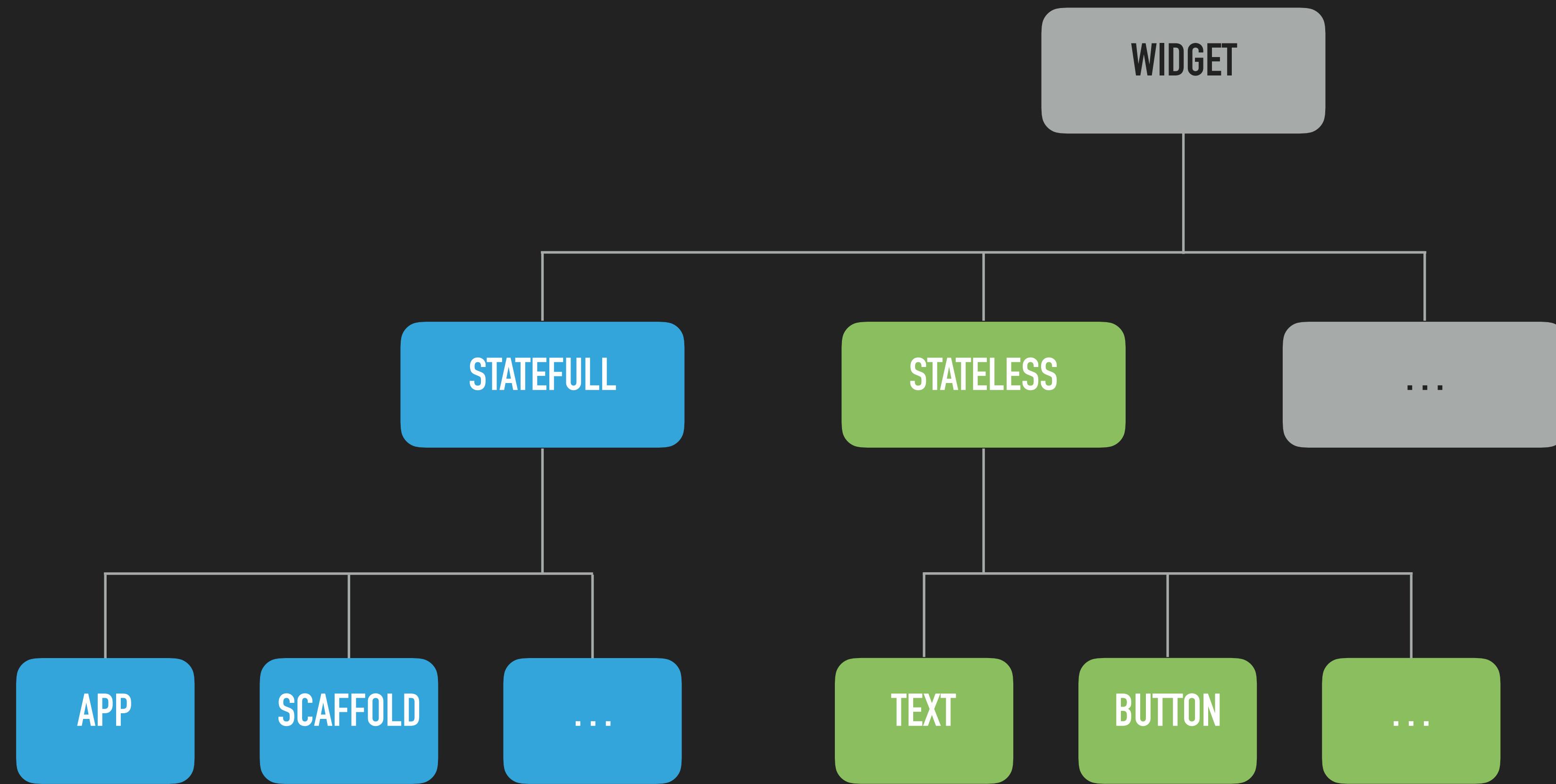
ПОЧЕМУ FLUTTER?



EVERYTHING IS A WIDGET

- ▶ Структурные элементы
- ▶ Стилистические элементы
- ▶ Вспомогательные элементы





```
class MyStatelessWidget extends StatelessWidget {  
    @override  
    Widget build(BuildContext context) {  
        return Container();  
    }  
}
```

```
class MyStatefulWidget extends StatefulWidget {
  @override
  _MyStatefulWidgetState createState() => _MyStatefulWidgetState();
}

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



```
class My StatefulWidget extends StatefulWidget {  
    @override  
    _My StatefulWidget createState() => _My StatefulWidget();  
}  
  
class _My StatefulWidget extends State<My StatefulWidget> {  
    @override  
    void initState() {  
        // TODO: implement initState  
        super.initState();  
    }  
  
    @override  
    void dispose() {  
        // TODO: implement dispose  
        super.dispose();  
    }  
  
    @override  
    Widget build(BuildContext context) {  
        return Container();  
    }  
}
```

```
class My StatefulWidget extends StatefulWidget {  
    @override  
    _My StatefulWidget createState() => _My StatefulWidget();  
}  
  
class _My StatefulWidget extends State<My StatefulWidget> {  
  
    void _updateMy StatefulWidget() {  
        setState(() {  
            //TODO: Update variables  
        });  
    }  
  
    @override  
    Widget build(BuildContext context) {  
        return Container();  
    }  
}
```

ЗАПУСК ПРИЛОЖЕНИЯ



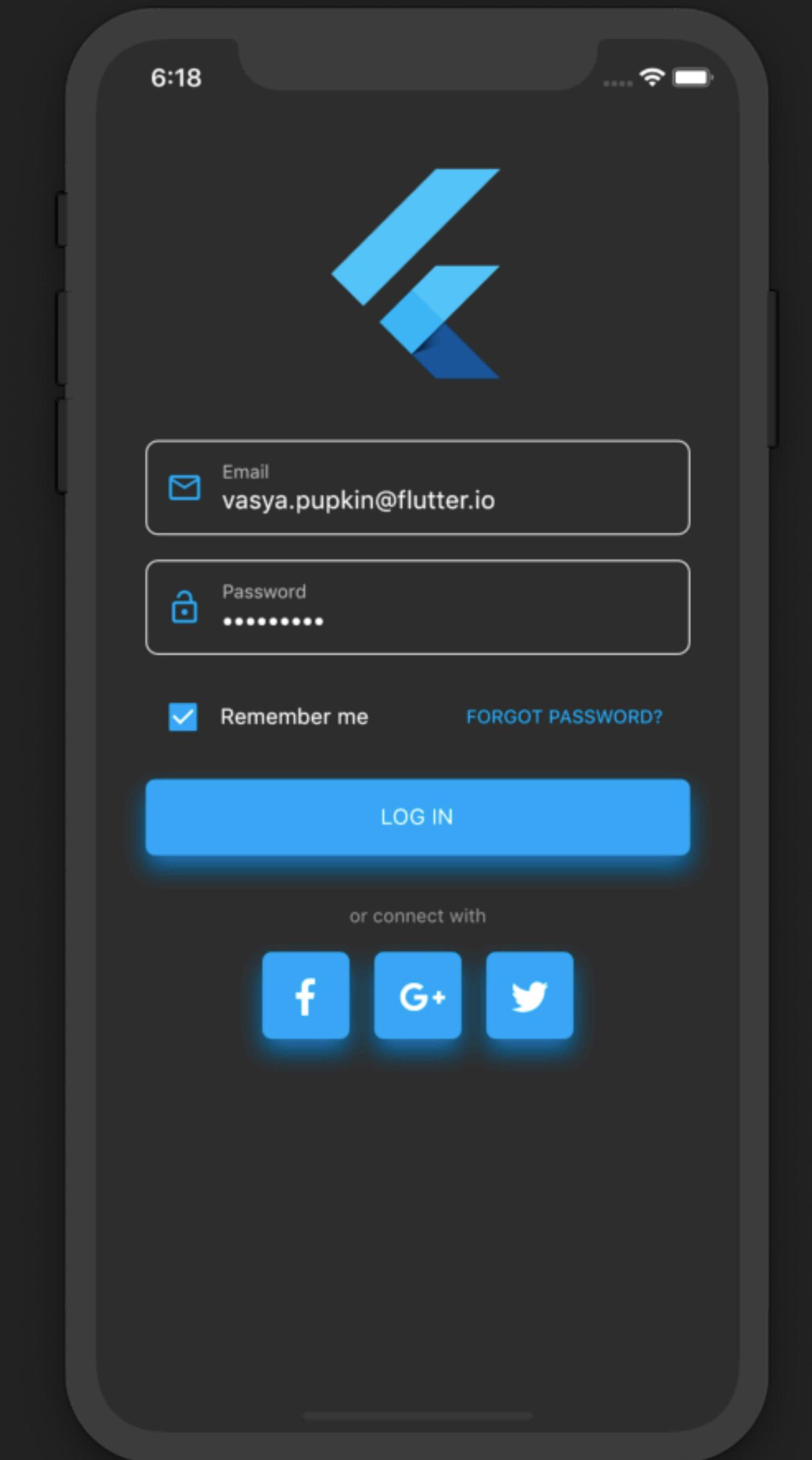
COLD START

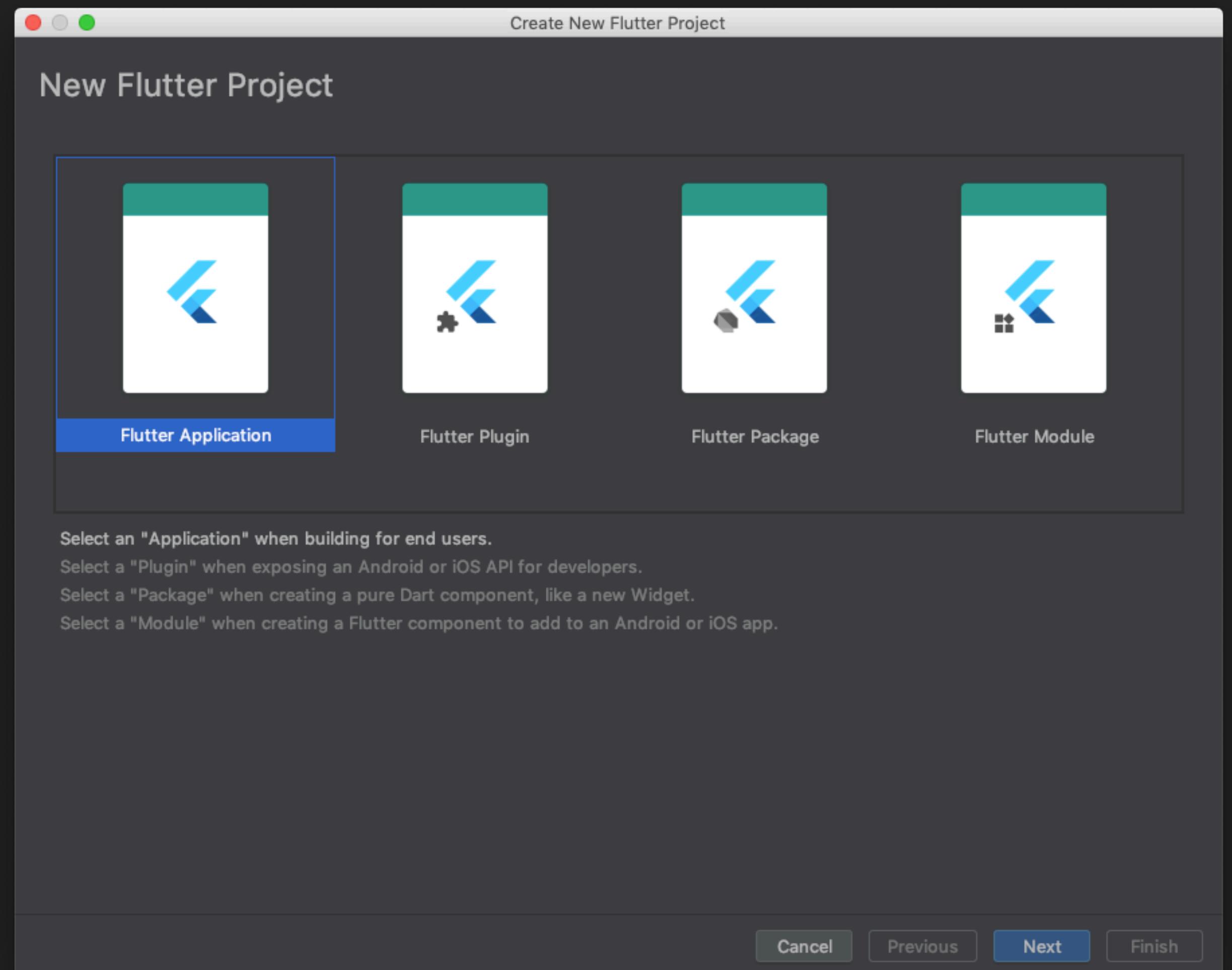


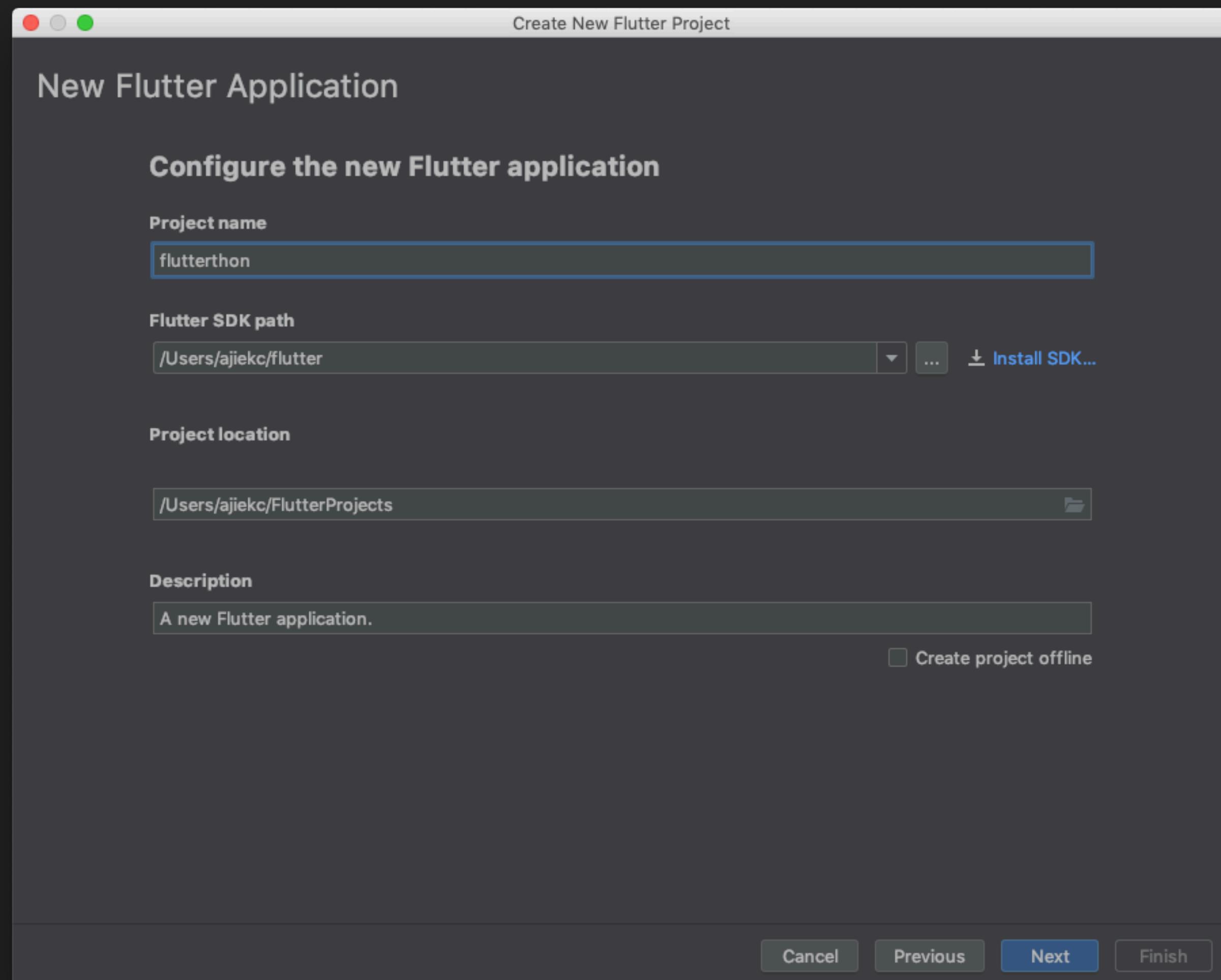
HOT RESTART

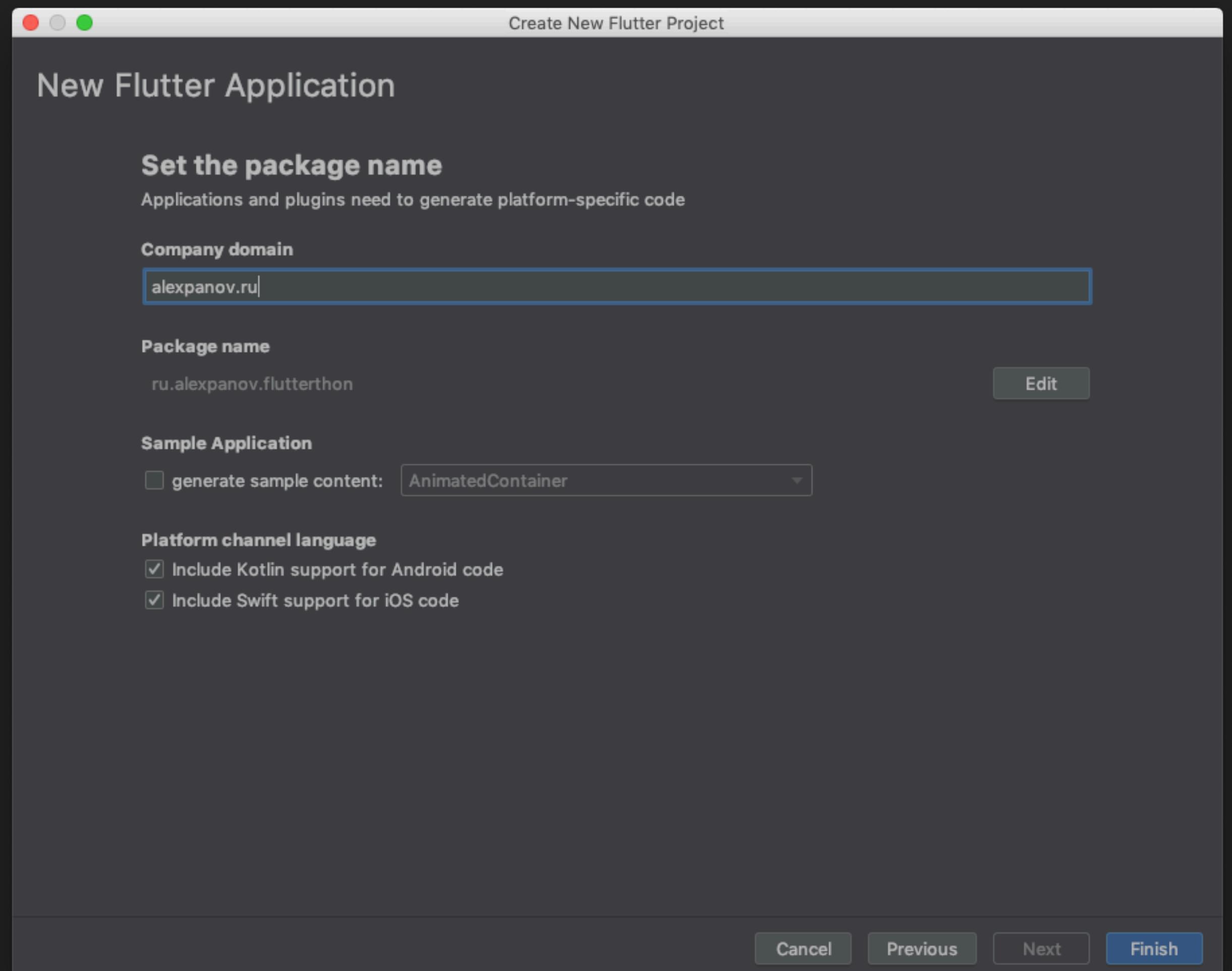


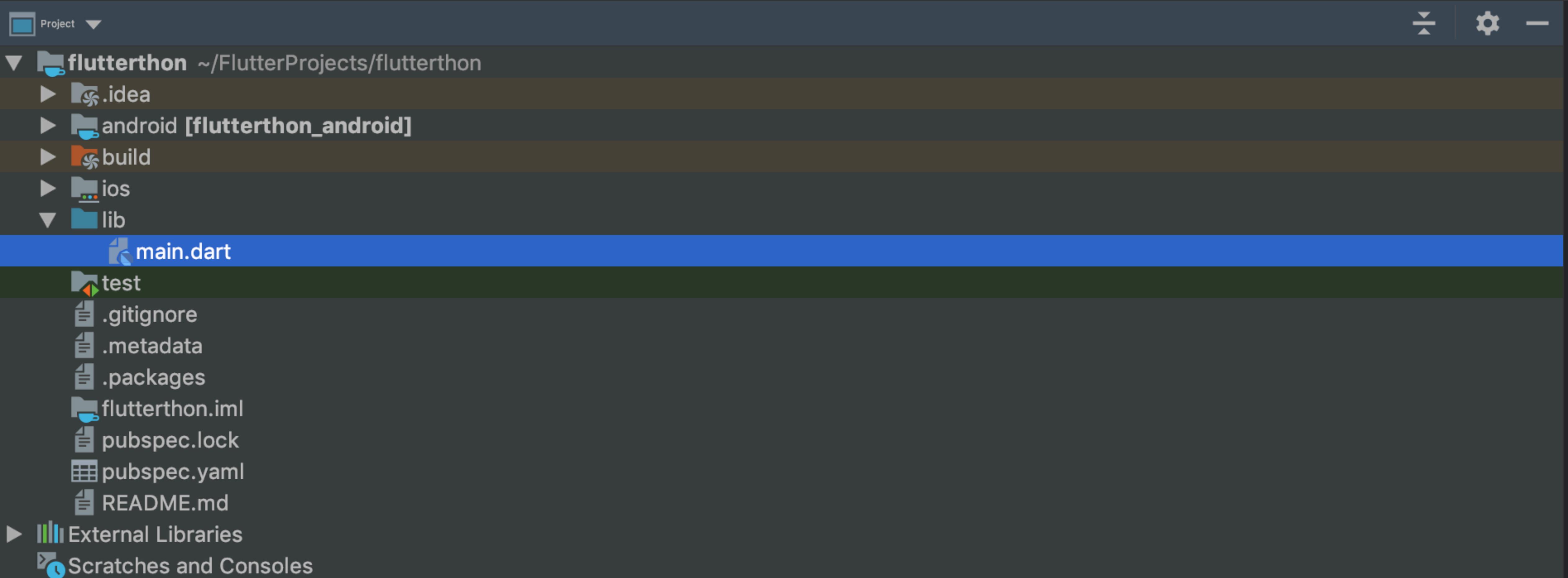
HOT RELOAD











A screenshot of an IDE interface, likely Android Studio, displaying a Dart file named `main.dart`. The code imports the `flutter/material.dart` package and defines the `main` function which runs the `MyApp` widget.

```
import 'package:flutter/material.dart';

void main() => runApp(MyApp());
```

The IDE features a dark theme with light-colored syntax highlighting. The top bar shows the project name `flutterthon`, the file path `~/FlutterProjects/flutterthon`, and the current file `main.dart`. The bottom navigation bar includes tabs for `Logcat`, `TODO`, `Terminal`, and `Dart Analysis`. On the right side, there are several floating toolbars: `Key Promoter X`, `Flutter Outline`, `Device File Explorer`, and `Event Log`. The bottom right corner displays the number `18`.

```
1 import 'package:flutter/material.dart';  
2  
3 void main() => runApp(MyApp());  
4  
5 |
```

A screenshot of an IDE interface, likely Android Studio, displaying a Dart file named `main.dart`. The code defines a simple Flutter application structure:

```
import 'package:flutter/material.dart';

void main() => runApp(MyApp());

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Container();
  }
}
```

The IDE features a dark theme with syntax highlighting for Dart. On the right side, there is a preview window showing a black iPhone X device. The bottom right corner of the preview shows a small portion of a landscape image. The bottom of the screen contains various toolbars and status indicators.

The screenshot shows an IDE interface with a Dart file named `main.dart` open. The code defines a simple Flutter application:

```
import 'package:flutter/material.dart';

void main() => runApp(MyApp());

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      theme: ThemeData.dark(),
      home: Container()
    ); // MaterialApp
  }
}
```

The code editor has syntax highlighting and a vertical gutter with icons indicating code flow or analysis results. To the right of the editor is a large preview window showing an iPhone X simulator running the application. The simulator screen displays a dark-themed Flutter app with a single white `Container` placeholder. The top of the simulator shows the text "Dart UI". The bottom of the screen shows a portion of a forest scene.

```
import 'package:flutter/material.dart';

void main() => runApp(MyApp());

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      theme: ThemeData.dark(),
      home: LoginScreen(),
    ); // MaterialApp
  }
}

class LoginScreen extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: Container(),
    ); // Scaffold
  }
}
```

The screenshot shows a development environment with the following details:

- Project Structure:** The project is named "flutterthon" and contains a "lib" directory with a "main.dart" file.
- Code Editor:** The main editor window displays the Dart code for the "main.dart" file, which defines the entry point of the application and two widget classes.
- IDE Features:** The interface includes various toolbars and panels:
 - Top bar: Includes file operations like Open, Save, and Print, along with tabs for "iPhone XR" and "main.dart".
 - Left sidebar: Shows "Resource Manager", "Project" (with a green checkmark), "Build Variants", "Layout Captures", "Structure", "Favorites", and "Logcat", "TODO", "Terminal", "Dart Analysis", and "Run" buttons.
 - Right sidebar: Includes "Key Promoter X", "Flutter Outline", "Flutter Inspector", "Device File Explorer", and "Event Log" and "Flutter Performance" tabs at the bottom.
- Simulator:** A large simulator window on the right shows a dark-themed mobile application interface with a red "DEBUG" indicator in the top right corner.
- Bottom Status Bar:** Shows the time as 20:6, encoding as UTF-8, and other system status indicators.

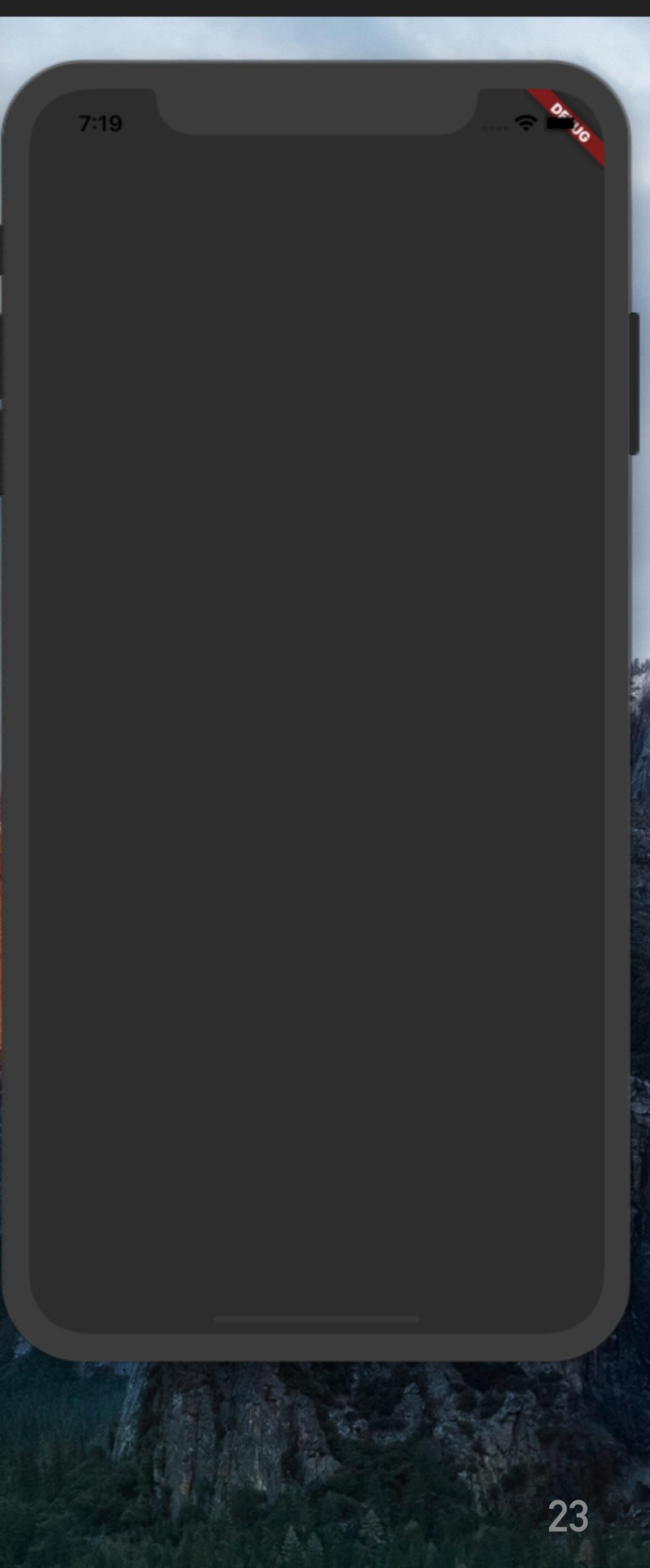
flutterthon [~/FlutterProjects/flutterthon] - .../pubspec.yaml [flutterthon]

Project Packages get Packages upgrade Flutter upgrade Flutter doctor

flutterthon ~/FlutterProjects/flutterthon .idea flutterthon_andro build images flutter.png ios lib main.dart test .gitignore .metadata .packages flutterthon.iml pubspec.lock pubspec.yaml README.md External Libraries Scratches and Consoles

```
34
35     # The following section is specific to Flutter.
36     flutter:
37
38         # The following line ensures that the Material Icons font is
39         # included with your application, so that you can use the icons
40         # the material Icons class.
41         uses-material-design: true
42
43         # To add assets to your application, add an assets section, like
44         # assets:
45         #   - images/a_dot_burr.jpeg
46         #   - images/a_dot_ham.jpeg
47
48         # An image asset can refer to one or more resolution-specific
49         # https://flutter.dev/assets-and-images/#resolution-aware.
50
51         # For details regarding adding assets from package dependencies,
52         # https://flutter.dev/assets-and-images/#from-packages
53
54         # To add custom fonts to your application, add a fonts section
55         # in this "flutter" section. Each entry in this list should have
56         # "family" key with the font family name, and a "fonts" key with
57         # list giving the asset and other descriptors for the font. For
58         # example:
59         # fonts:
60         #   - family: Schyler
61         #     fonts:
62         #       - asset: fonts/Schyler-Regular.ttf
63         #       - asset: fonts/Schyler-Italic.ttf
64         #         style: italic
65         #   - family: Trajan Pro
66         #     fonts:
```

Document 1/1 > flutter:



flutterthon [~/FlutterProjects/flutterthon] - .../pubspec.yaml [flutterthon]

Project Packages get Packages upgrade Flutter upgrade Flutter doctor

flutterthon ~/FlutterP .idea android [flutterthon_andro build images flutter.png ios lib main.dart test .gitignore .metadata .packages flutterthon.iml pubspec.lock pubspec.yaml README.md External Libraries Scratches and Consoles

Install referenced packages

ing section is specific to Flutter.

36 flutter:

37 # The following line ensures that the Material Icons font is
38 # included with your application, so that you can use the icon
39 # the material Icons class.
40 uses-material-design: true

41

42 # To add assets to your application, add an assets section, li
43 assets:
44 - images/flutter.png

45

46 # An image asset can refer to one or more resolution-specific
47 # https://flutter.dev/assets-and-images/#resolution-aware.

48

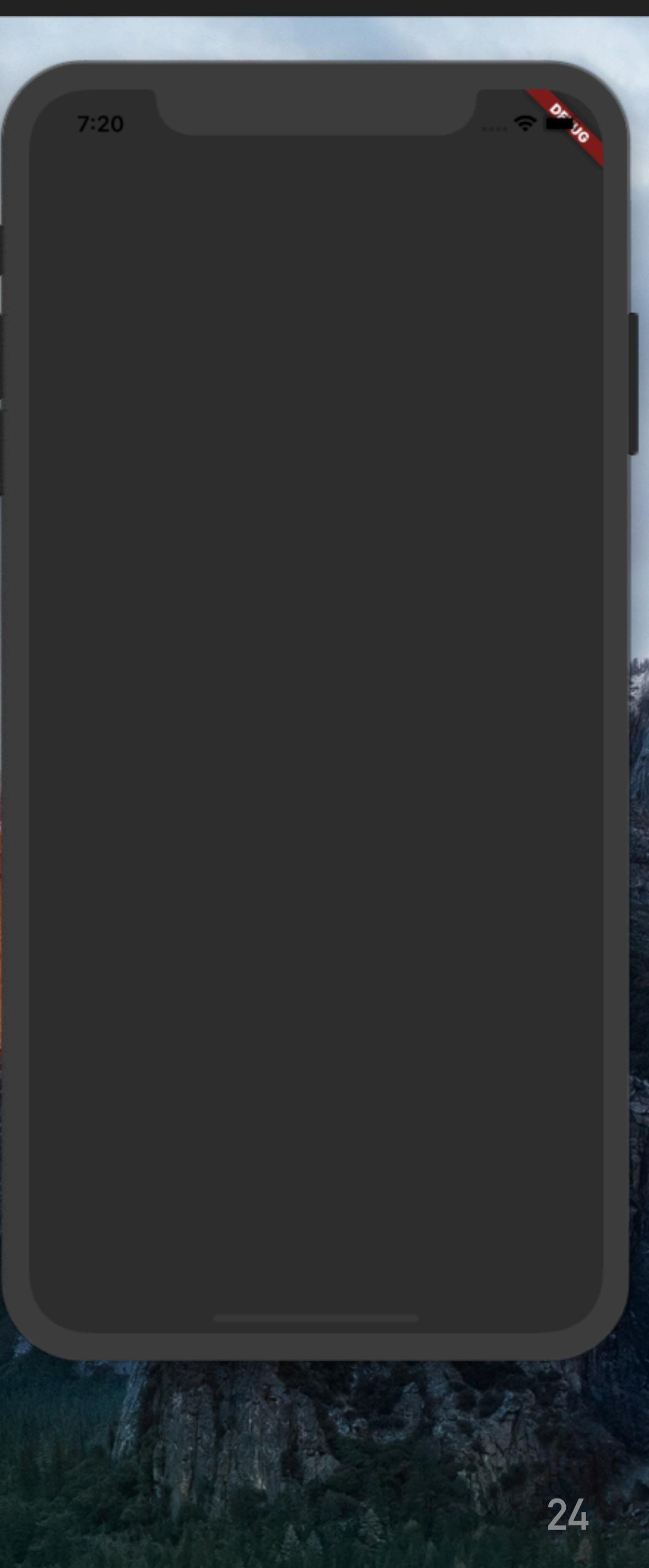
49

50 # For details regarding adding assets from package dependencie
51 # https://flutter.dev/assets-and-images/#from-packages

52

53 # To add custom fonts to your application, add a fonts section
54 # in this "flutter" section. Each entry in this list should ha
55 # "family" key with the font family name, and a "fonts" key wi
56 # list giving the asset and other descriptors for the font. Fo
57 # example:
58 # fonts:
59 # - family: Schyler
60 # fonts:
61 # - asset: fonts/Schyler-Regular.ttf
62 # - asset: fonts/Schyler-Italic.ttf
63 # style: italic
64 # - family: Trajan Pro
65 # fonts:
66 # - asset: fonts/TrajanPro.ttf

Document 1/1 > flutter:



The screenshot shows a development environment for a Flutter application named "flutterthon".

Code Editor:

- The main file is `lib/main.dart`.
- The code defines a `MaterialApp` with a dark theme and a `LoginScreen` as the home screen.
- The `LoginScreen` class extends `StatelessWidget` and contains a `Scaffold` with a large `Image.asset` placeholder.

```
9   theme: ThemeData.dark(),
10  home: LoginScreen(),
11 ); // MaterialApp
12 }
13 }
14
15 class LoginScreen extends StatelessWidget {
16   @override
17   Widget build(BuildContext context) {
18     return Scaffold(
19       body: Image.asset(
20         'images/flutter.png',
21         height: 150.0,
22         width: 150.0,
23       ), // Image.asset
24     ); // Scaffold
25   }
26 }
27
```

IDE Features:

- Project tree on the left shows files like `main.dart`, `pubspec.yaml`, and `assets/flutter.png`.
- Toolbars at the bottom include Logcat, TODO, Terminal, Dart Analysis, Run, Messages, Event Log, and Flutter Performance.
- Right-hand sidebar includes Key Promoter X, Flutter Outline, Flutter Inspector, and Device File Explorer.
- Bottom status bar shows the device is an iPhone XR, the time is 7:23, and the battery level is 23%.

flutterthon [~/FlutterProjects/flutterthon] - .../lib/main.dart [flutterthon]

main.dart

```
9   theme: ThemeData.dark(),
10  home: LoginScreen(),
11 ); // MaterialApp
12 }
13 }
14
15 class LoginScreen extends StatelessWidget {
16   @override
17   Widget build(BuildContext context) {
18     return Scaffold(
19       body: SafeArea(
20         child: Image.asset(
21           'images/flutter.png',
22           height: 150.0,
23           width: 150.0,
24         ), // Image.asset
25       ), // SafeArea
26     ); // Scaffold
27   }
28 }
29
```

Resource Manager

1: Project

Build Variants

Layout Captures

Z: Structure

2: Favorites

6: Logcat

TODO

Terminal

Dart Analysis

Run

0: Messages

Event Log

Flutter Performance

25:8 LF UTF-8 2 spaces ☺ ☹ ☻

7:24

Key Promoter X

Flutter Outline

Flutter Inspector

Device File Explorer

The image shows a developer's workspace for a Flutter application. On the left, the code editor displays the main.dart file with Dart code defining a dark-themed MaterialApp and a LoginScreen widget that contains a centered image of the Flutter logo. The right side of the interface features a mobile device emulator for an iPhone XR, showing the running application with the blue and white Flutter logo on a black background. The top of the screen shows the project name 'flutterthon' and the current file 'main.dart'. Various toolbars and panels are visible along the top and right edges of the IDE.

```
9   theme: ThemeData.dark(),
10  home: LoginScreen(),
11 ); // MaterialApp
12 }
13 }
14
15 class LoginScreen extends StatelessWidget {
16   @override
17   Widget build(BuildContext context) {
18     return Scaffold(
19       body: SafeArea(
20         child: Image.asset(
21           'images/flutter.png',
22           height: 150.0,
23           width: 150.0,
24         ), // Image.asset
25       ), // SafeArea
26     ); // Scaffold
27 }
28 }
29
```

The screenshot shows a development environment for Flutter. On the left, the code editor displays `main.dart` with the following code:

```
11 ); // MaterialApp
12 }
13 }
14
15 class LoginScreen extends StatelessWidget {
16   @override
17   Widget build(BuildContext context) {
18     return Scaffold(
19       body: SafeArea(
20         child: Column(
21           children: <Widget>[
22             Image.asset(
23               'images/flutter.png',
24               height: 150.0,
25               width: 150.0,
26             ), // Image.asset
27             TextField(
28               decoration: InputDecoration(
29                 labelText: 'Email',
30                 border: OutlineInputBorder(),
31               ), // InputDecoration
32             ), // TextField
33           ], // <Widget>[]
34         ), // Column
35       );
36     }
37 }
```

On the right, a preview window shows an iPhone XR displaying the Flutter logo at the top and a text input field labeled "Email" below it. The status bar indicates the time is 7:30 and there is a "DEBUG" indicator.



A screenshot of an IDE (Android Studio) displaying a Flutter application. The code editor shows `main.dart` with the following code:

```
19 body: SafeArea(  
20   child: Column(  
21     children: <Widget>[  
22       Image.asset(  
23         'images/flutter.png',  
24         height: 150.0,  
25         width: 150.0,  
26       ), // Image.asset  
27       TextField(  
28         decoration: InputDecoration(  
29           labelText: 'Email',  
30           labelStyle: TextStyle(color: Colors.grey[400]),  
31           border: InputBorder.none,  
32         ), // InputDecoration  
33       ), // TextField  
34     ], // <Widget>[]  
35   ), // Column  
36 ), // SafeArea  
37 ); // Scaffold  
38 }  
39 }  
40 }
```

The right side of the interface shows a preview of the app running on an iPhone XR. The screen displays a large blue Flutter logo at the top, followed by a text input field labeled "Email". The bottom of the screen shows a blurred landscape.

The screenshot shows a development environment for Flutter. On the left, the code editor displays `main.dart` with the following code:

```
19 body: SafeArea(  
20   child: Column(  
21     children: <Widget>[  
22       Image.asset(  
23         'images/flutter.png',  
24         height: 150.0,  
25         width: 150.0,  
26       ), // Image.asset  
27       TextField(  
28         decoration: InputDecoration(  
29           labelText: 'Email',  
30           labelStyle: TextStyle(color: Colors.grey[400]),  
31           border: InputBorder.none,  
32           prefixIcon: Icon(  
33             Icons.mail_outline,  
34             color: Colors.lightBlue,  
35           ), // Icon  
36           ), // InputDecoration  
37         ), // TextField  
38       ], // <Widget>[]  
39     ), // Column  
40   ), // SafeArea  
41 ); // Scaffold  
42 }
```

The right side of the interface shows a preview of the application running on an iPhone XR. The screen displays a large blue Flutter logo at the top, followed by a text input field labeled "Email". The bottom of the screen shows the status bar with "7:34" and "iPhone XR — 12.2".

Android Studio File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help

flutterthon [~/FlutterProjects/flutterthon] - .../lib/main.dart [flutterthon]

Resource Manager

1: Project

main.dart x

```
19 body: SafeArea(
20   child: Column(
21     children: <Widget>[
22       Image.asset(
23         'images/flutter.png',
24         height: 150.0,
25         width: 150.0,
26       ), // Image.asset
27       TextField(
28         decoration: InputDecoration(
29           labelText: 'Email',
30           labelStyle: TextStyle(color: Colors.grey[400]),
31           border: InputBorder.none,
32           prefixIcon: Icon(
33             Icons.mail_outline,
34             color: Colors.lightBlue,
35           ), // Icon
36           ), // InputDecoration
37         ), // TextField
38       ], // <Widget>[]
39     ), // Column
40   ), // SafeArea
41 ); // Scaffold
42 }
```

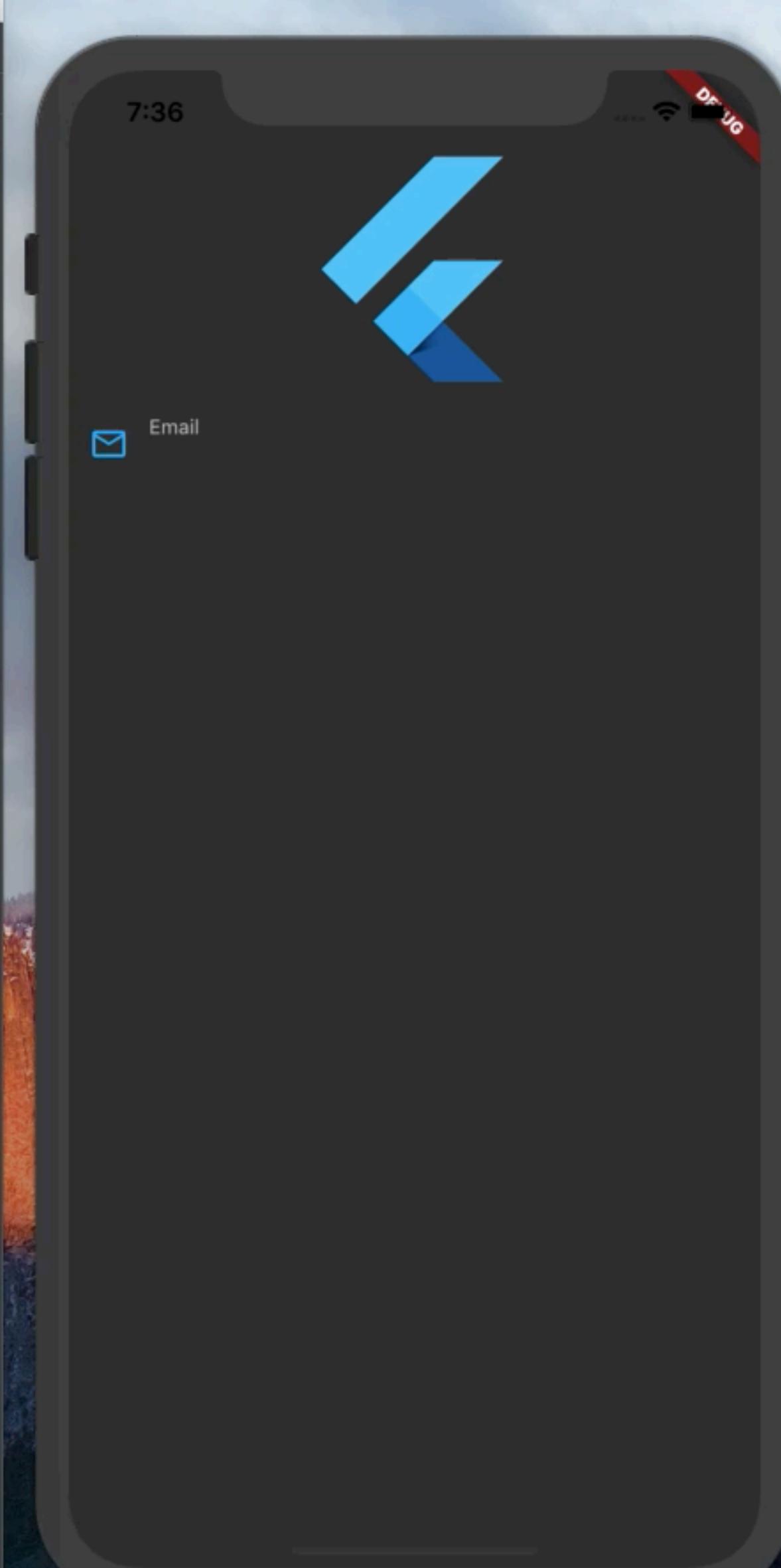
6: Logcat 7: Structure 8: Favorites 2: Favorites 9: Layout Captures 10: Build Variants 11: Favorites 12: Resource Manager

Event Log Flutter Performance

27:18 LF UTF-8 2 spaces 32

7:36 Key Promoter X Flutter Outline Flutter Inspector Device File Explorer

32

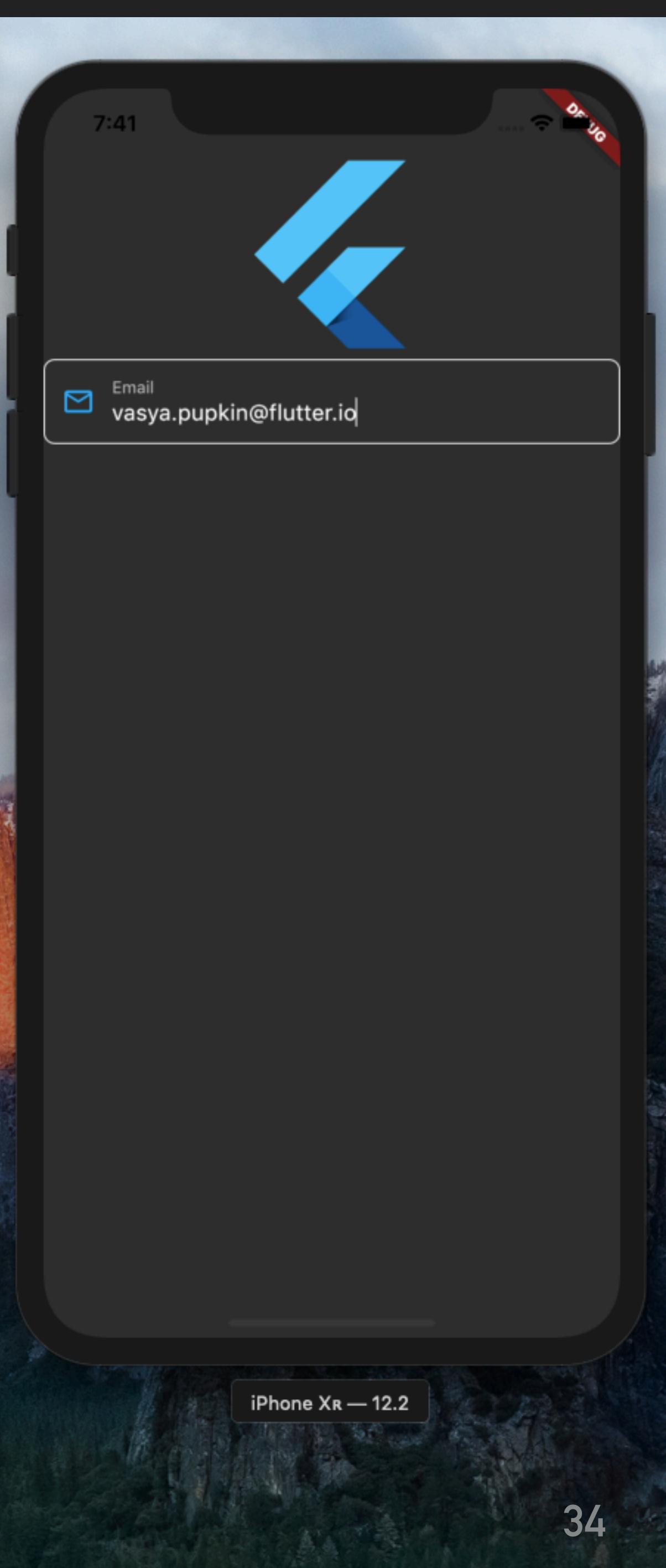


The screenshot shows a development environment for a Flutter application named "flutterthon". The main window displays the Dart code for the `main.dart` file, which contains the following code:

```
18 return Scaffold(  
19   body: SafeArea(  
20     child: Column(  
21       children: <Widget>[  
22         Image.asset(  
23           'images/flutter.png',  
24           height: 150.0,  
25           width: 150.0,  
26         ), // Image.asset  
27         Container(  
28           decoration: ShapeDecoration(  
29             shape: RoundedRectangleBorder(  
30               borderRadius: BorderRadius.all(Radius.circular(8.0)),  
31               side: BorderSide(color: Colors.grey[100]),  
32             ), // RoundedRectangleBorder  
33           ), // ShapeDecoration  
34           child: TextField(  
35             decoration: InputDecoration(  
36               labelText: 'Email',  
37               labelStyle: TextStyle(color: Colors.grey[400]),  
38               border: InputBorder.none,  
39               prefixIcon: Icon(  
40                 Icons.mail_outline,  
41                 color: Colors.lightBlue,
```

The right side of the interface shows a preview of the application running on an iPhone XR. The screen displays a large blue Flutter logo at the top, followed by a card with an envelope icon and the email address "vasya.pupkin@flutter.io". The bottom of the screen shows a navigation bar with icons for home, back, forward, and search.

```
flutterthon [~/FlutterProjects/flutterthon] - .../lib/main.dart [flutterthon]
main.dart
Resource Manager
18 return Scaffold(
19   body: SafeArea(
20     child: Column(
21       children: <Widget>[
22         Image.asset(
23           'images/flutter.png',
24           height: 150.0,
25           width: 150.0,
26         ), // Image.asset
27         Container(
28           decoration: ShapeDecoration(
29             shape: RoundedRectangleBorder(
30               borderRadius: BorderRadius.all(Radius.circular(8.0)),
31             side: BorderSide(color: Colors.grey[100]),
32           ), // RoundedRectangleBorder
33           child: TextField(
34             cursorColor: Colors.white,
35             cursorWidth: 1.0,
36             decoration: InputDecoration(
37               labelText: 'Email',
38               labelStyle: TextStyle(color: Colors.grey[400]),
39               border: InputBorder.none,
40               prefixIcon: Icon(
41             
```



A screenshot of an IDE (Android Studio) displaying a Flutter application. The code editor shows the main.dart file with the following code:

```
17 // ignore_for_file: prefer_const_constructors, prefer_const_literals_in_to_string
18
19 return Scaffold(
20   body: SafeArea(
21     child: Padding(
22       padding: const EdgeInsets.symmetric(horizontal: 32.0),
23       child: Column(
24         children: <Widget>[
25           Padding(
26             padding: const EdgeInsets.symmetric(vertical: 32.0),
27             child: Image.asset(
28               'images/flutter.png',
29               height: 150.0,
30               width: 150.0,
31             ), // Image.asset
32           ), // Padding
33           Container(
34             decoration: ShapeDecoration(
35               shape: RoundedRectangleBorder(
36                 borderRadius: BorderRadius.all(Radius.circular(8.0)),
37                 side: BorderSide(color: Colors.grey[100]),
38               ), // RoundedRectangleBorder
39             ), // ShapeDecoration
40             child: TextField(
41               cursorColor: Colors.white,
42               cursorWidth: 1.0,
```

The right side of the interface shows a preview of the app running on an iPhone XR. The screen displays the Flutter logo at the top and a large blue button labeled "Email" below it. The status bar shows the time as 7:43 and the battery level.

The screenshot shows a development environment for Flutter. On the left, the code editor displays `main.dart` with the following code:

```
51 ), // TextField
52 ), // Container
53 Container(
54   decoration: ShapeDecoration(
55     shape: RoundedRectangleBorder(
56       borderRadius: BorderRadius.all(Radius.circular(8.0)),
57     ),
58   ),
59   side: BorderSide(color: Colors.grey[100]),
60   ), // RoundedRectangleBorder
61   child: TextField(
62     cursorColor: Colors.white,
63     cursorWidth: 1.0,
64     decoration: InputDecoration(
65       labelText: 'Email',
66       labelStyle: TextStyle(color: Colors.grey[400]),
67       border: InputBorder.none,
68       prefixIcon: Icon(
69         Icons.mail_outline,
70         color: Colors.lightBlue,
71       ),
72     ),
73   ),
74 ), // Container
75 ), // <Widget>[]
```

The right side of the interface shows a preview of the application running on an iPhone XR. The screen displays a dark-themed UI with two email input fields. The top field has a blue rounded rectangle border and a light blue placeholder 'Email'. The bottom field is plain white with a grey placeholder 'Email'. The status bar at the top of the phone screen shows the time as 7:45 and a 'DEBUG' indicator.

The screenshot shows an IDE interface with a Flutter project named "flutterthon". The main editor window displays the Dart file "main.dart" containing the following code:

```
51 ), // TextField
52 ), // Container
53 Container(
54   decoration: ShapeDecoration(
55     shape: RoundedRectangleBorder(
56       borderRadius: BorderRadius.all(Radius.circular(8.0))
57     ),
58     side: BorderSide(color: Colors.grey[100]),
59   ), // RoundedRectangleBorder
60   child: TextField(
61     obscureText: true,
62     cursorColor: Colors.white,
63     cursorWidth: 1.0,
64     decoration: InputDecoration(
65       labelText: 'Password',
66       labelStyle: TextStyle(color: Colors.grey[400]),
67       border: InputBorder.none,
68       prefixIcon: Icon(
69         Icons.lock_open,
70         color: Colors.lightBlue,
71       ), // Icon
72     ), // InputDecoration
73   ), // TextField
74 ) // Container
```

The right side of the interface shows a preview of the Flutter application running on an iPhone XR. The app displays a form with two fields: "Email" (containing "vasya.pupkin@flutter.io") and "Password" (containing a redacted password). The Flutter logo is visible at the top of the screen.

The screenshot shows a development environment for Flutter. On the left, the code editor displays `main.dart` with the following code:

```
51 ), // TextField
52 ), // Container
53 Container(
54   decoration: ShapeDecoration(
55     shape: RoundedRectangleBorder(
56       borderRadius: BorderRadius.
57       side: BorderSide(color: Col
58     ), // RoundedRectangleBorder
59   ), // ShapeDecoration
60   child: TextField(
61     obscureText: true,
62     cursorColor: Colors.white,
63     cursorWidth: 1.0,
64     decoration: InputDecoration(
65       labelText: 'Password',
66       labelStyle: TextStyle(color:
67       border: InputBorder.none,
68       prefixIcon: Icon(
69         Icons.lock_open,
70         color: Colors.lightBlue,
71       ), // Icon
72     ), // InputDecoration
73   ), // TextField
74 ) // Container
```

The right side of the interface includes the following components:

- Flutter Outline:** A tree view showing the widget hierarchy:
 - main() → void
 - MyApp
 - MyApp
 - build(BuildContext context) → Widget
 - MaterialApp theme: ThemeData.dark()
 - LoginScreen
 - LoginScreen
 - build(BuildContext context) → Widget
 - Scaffold
 - SafeArea
 - Padding padding: const EdgeInsets.symmetric(horizontal: 15.0, vertical: 10.0)
 - Column
 - Padding padding: const EdgeInsets.symmetric(vertical: 10.0)
 - Image name: 'images/flutter.png', height: 150.0
 - Container decoration: BoxDecoration
 - TextField cursorColor: Colors.white, cursorWidth: 1.0
 - Container
 - Text
- Flutter Inspector:** A context menu for a `Container` widget, with the "Extract widget..." option highlighted.
- Device:** An iPhone XR displaying the Flutter application. The screen shows a login form with fields for Email (containing `vasya.pupkin@flutter.io`) and Password (containing `.....`). The Flutter logo is visible at the top of the screen.

The screenshot shows a development environment for Flutter. On the left, the code editor displays `main.dart` with the following code:

```
52     ), // Container
53     new LoginField()
54   ], // <Widget>[]
55   ), // Column
56   ), // Padding
57   ), // SafeArea
58 ); // Scaffold
59 }
60 }
61
62 class LoginField extends StatelessWidget {
63   const LoginField({
64     Key key,
65   }) : super(key: key);
66
67   @override
68   Widget build(BuildContext context) {
69     return Container(
70       decoration: ShapeDecoration(
71         shape: RoundedRectangleBorder(
72           borderRadius: BorderRadius.all(Radius.circular(8.0)),
73           side: BorderSide(color: Colors.grey[100]),
74         ), // RoundedRectangleBorder
75       ), // ShapeDecoration
76     );
77   }
78 }
```

The right side of the interface shows a preview of the application running on an iPhone XR. The screen displays a login form with a large blue K logo at the top. Below it are two input fields: one for Email with the value "vasya.pupkin@flutter.io" and one for Password with the value ".....". The status bar shows the time as 7:48 and the battery level.

The screenshot shows a development environment for Flutter. On the left, the code editor displays `main.dart` with the following code:

```
52     ), // Container
53     LoginField()
54   ], // <Widget>[]
55   ),
56   ),
57   ),
58   );
59 }
60 }
61
62 class LoginField extends StatelessWidget {
63   LoginField({this.icon, this.label, this.obscureText = false});
64
65   final IconData icon;
66   final String label;
67   final bool obscureText;
68
69   @override
70   Widget build(BuildContext context) {
71     return Container(
72       decoration: ShapeDecoration(
73         shape: RoundedRectangleBorder(
74           borderRadius: BorderRadius.all(Radius.circular(8.0)),
75           side: BorderSide(color: Colors.grey[100]),
```

The `main.dart` file imports `main.dart` from the `lib` directory. The `main` function contains a `Scaffold` widget with a `SafeArea`, `Padding`, and a `Column` containing two `LoginField` widgets. The `LoginField` class is defined with `icon`, `label`, and `obscureText` parameters. It returns a `Container` with rounded rectangle borders and grey sides.

On the right, a simulated iPhone XR device displays the application. The screen shows the Flutter logo at the top, followed by two input fields: one for "Email" with the value "vasya.pupkin@flutter.io" and one for "Password" with the value ".....". The device status bar indicates the time is 7:59 and shows a "DEBUG" mode indicator.

The screenshot shows a development environment for Flutter. On the left, the code editor displays `main.dart` with the following code:

```
69  
70 @override  
71 Widget build(BuildContext context) {  
72   return Container(  
73     decoration: ShapeDecoration(  
74       shape: RoundedRectangleBorder(  
75         borderRadius: BorderRadius.all(Radius.circular(8.0)),  
76         side: BorderSide(color: Colors.grey[100]),  
77       ), // RoundedRectangleBorder  
78     ), // ShapeDecoration  
79     child: TextField(  
80       obscureText: obscureText,  
81       cursorColor: Colors.white,  
82       cursorWidth: 1.0,  
83       decoration: InputDecoration(  
84         labelText: label,  
85         labelStyle: TextStyle(color: Colors.grey[400]),  
86         border: InputBorder.none,  
87         prefixIcon: Icon(  
88           icon,  
89           color: Colors.lightBlue,  
90         ), // Icon  
91       ), // InputDecoration  
92     ), // TextField  
93   ); // Container
```

The right side of the interface shows a preview of the application running on an iPhone XR. The app displays a login screen with a large blue Flutter logo at the top. Below it are two text input fields: one for 'Email' containing 'vasya.pupkin@flutter.io' and one for 'Password' containing a masked value '.....'. The status bar indicates the time is 7:57 and shows a 'DEBUG' mode indicator.

flutterthon [~/FlutterProjects/flutterthon] - .../lib/main.dart [flutterthon]

main.dart

```
27     'images/flutter.png',
28     height: 150.0,
29     width: 150.0,
30   ), // Image.asset
31   ), // Padding
32   LoginField(
33     icon: Icons.mail_outline,
34     label: 'Email',
35   ), // LoginField
36   LoginField(
37     icon: Icons.lock_open,
38     label: 'Password',
39     obscureText: true,
40   ) // LoginField
41   ], // <Widget>[]
42   ), // Column
43   ), // Padding
44   ), // SafeArea
45 ); // Scaffold
46 }
47 }
48
49 class LoginField extends StatelessWidget {
50   LoginField({this.icon, this.label, this.obscureText = false});
```

Resource Manager

Project

Build Variants

Layout Captures

Structure

Favorites

Logcat

TODO

Terminal

Dart Analysis

Run

Messages

Event Log

Flutter Performance

40:16 LF UTF-8 2 spaces ☺ ☹ ☻

8:00

Key Promoter X

Flutter Outline

Flutter Inspector

Device File Explorer

8:00

✉ Email vasya.pupkin@flutter.io

🔒 Password

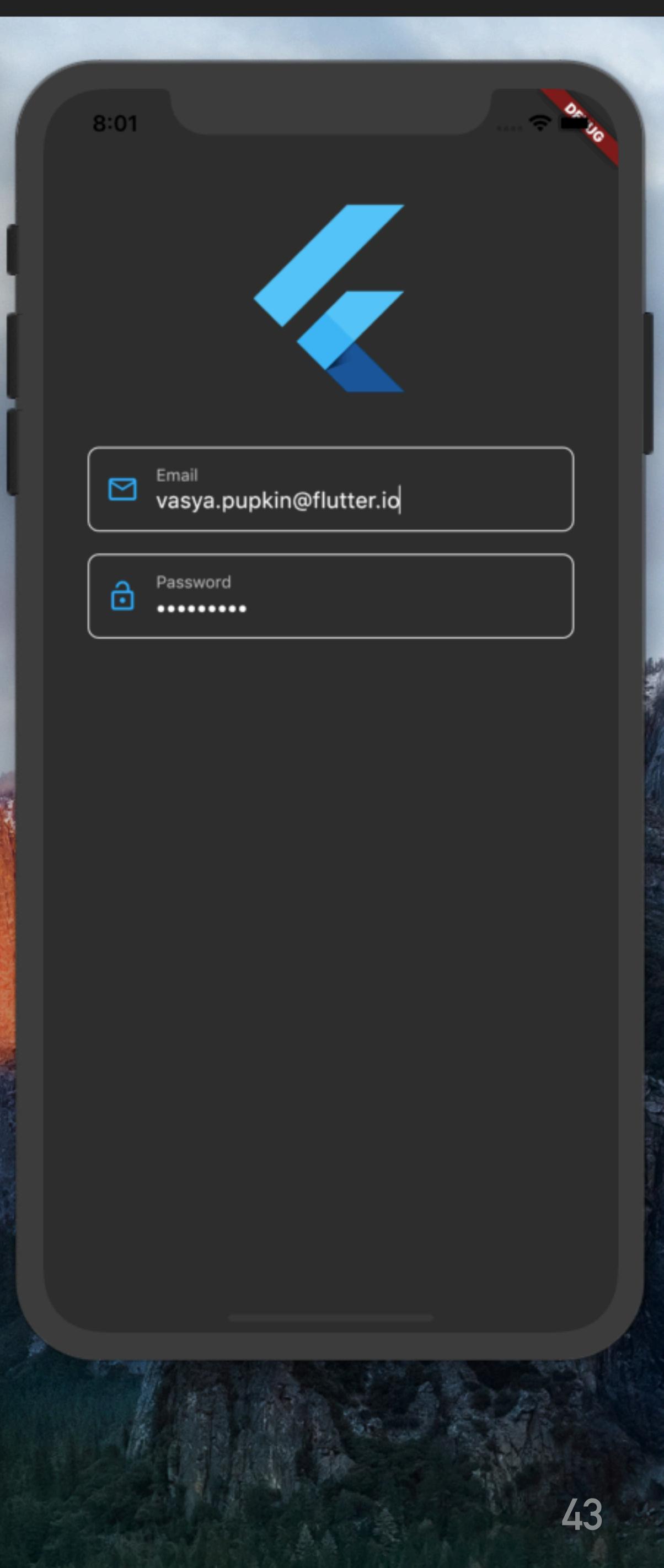
42

The screenshot shows the Android Studio interface with the following details:

- Title Bar:** flutterthon [~/FlutterProjects/flutterthon] - .../lib/main.dart [flutterthon]
- Toolbar:** Includes icons for back, forward, search, and various build options.
- Side Navigation:** Shows the project structure (1: Project, 2: Favorites, Z: Structure), Build Variants, Layout Captures, and Device File Explorer.
- Code Editor:** The main.dart file is open, displaying the following Dart code:

```
27     'images/flutter.png',
28     height: 150.0,
29     width: 150.0,
30   ), // Image.asset
31   ), // Padding
32   LoginField(
33     icon: Icons.mail_outline,
34     label: 'Email',
35   ), // LoginField
36   SizedBox(height: 16.0),
37   LoginField(
38     icon: Icons.lock_open,
39     label: 'Password',
40     obscureText: true,
41   ) // LoginField
42   ], // <Widget>[]
43   ), // Column
44   ), // Padding
45   ), // SafeArea
46 ); // Scaffold
47 }
48 }
49
50 class LoginField extends StatelessWidget {
```

The code defines a `SafeArea` widget with a `Column` containing two `LoginField` widgets and a centered `Image.asset`. The `LoginField` widget has a placeholder for an email icon and a password field with obscuring.



flutterthon [~/FlutterProjects/flutterthon] - .../lib/main.dart [flutterthon]

main.dart

```
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
```

```
    ), // Image.asset
    ), // Padding
LoginField(
  icon: Icons.mail_outline,
  label: 'Email',
), // LoginField
SizedBox(height: 16.0),
LoginField(
  icon: Icons.lock_open,
  label: 'Password',
  obscureText: true,
), // LoginField
SizedBox(height: 16.0),
Checkbox(
  value: false,
  onChanged: (newValue) {},
) // Checkbox
], // <Widget>[]
), // Column
), // Padding
), // SafeArea
); // Scaffold
}
```

Resource Manager

Project

Build Variants

Layout Captures

Structure

Favorites

Logcat

TODO

Terminal

Dart Analysis

Run

Messages

Event Log

Flutter Performance

46:16 LF UTF-8 2 spaces ☺ ☹ ☻

8:03

Key Promoter X

Flutter Outline

Flutter Inspector

Device File Explorer

44

The screenshot shows a development environment for a Flutter project named "flutterthon".

Code Editor:

- The main editor window displays the Dart file `main.dart`.
- The code defines a `LoginScreen` class that extends `StatefulWidget`. It includes an `_LoginScreenState` state class.
- The `_LoginScreenState` class has a `bool isRememberMeChecked = false;` field.
- The `build` method creates a `Scaffold` with a `SafeArea` and a `Padding` widget containing a `Column` of children.
- The first child of the `Column` is a `Padding` with horizontal padding of 32.0 and vertical padding of 32.0, containing an `Image.asset` with the path `'images/flutter.png'` and dimensions `height: 150.0, width: 150.0`.

IDE Features:

- Left sidebar: Project, Resource Manager, Build Variants, Layout Captures, Z: Structure, Favorites.
- Bottom navigation bar: Logcat, TODO, Terminal, Dart Analysis, Run, Messages, Event Log, Flutter Performance.
- Right sidebar: Key Promoter X, Flutter Outline, Flutter Inspector, Device File Explorer.

Preview:

- A preview of the Flutter application is shown on an iPhone XR device.
- The screen shows the large blue Flutter logo at the top.
- Below it are two input fields: one for "Email" and one for "Password".
- The bottom of the screen shows a blurred landscape image.

The screenshot shows a development environment for a Flutter application named "flutterthon". The main window displays the Dart code for the application's entry point, `main.dart`. The code defines a `Scaffold` with a `SafeArea`, `Padding`, and a `Column` containing three `TextFormFields` and a `Checkbox`. The `TextFormFields` have labels "Email" and "Password" with corresponding icons. The `Checkbox` has a value binding to a variable `_isRememberMeChecked`. The `main.dart` file is located in the `lib` directory of the project.

```
40 icon: Icons.mail_outline,
41 label: 'Email',
42 ), // LoginField
43 SizedBox(height: 16.0),
44 LoginField(
45   icon: Icons.lock_open,
46   label: 'Password',
47   obscureText: true,
48 ), // LoginField
49 SizedBox(height: 16.0),
50 Checkbox(
51   value: _isRememberMeChecked,
52   onChanged: (newValue) {
53     _isRememberMeChecked = !_isRememberMeChecked;
54   },
55   ), // Checkbox
56   ], // <Widget>[]
57   ), // Column
58   ), // Padding
59   ), // SafeArea
60 ); // Scaffold
61 }
62 }
63 }
```

The right side of the interface shows a preview of the application running on an iPhone XR. The screen displays a dark-themed login form with fields for "Email" and "Password". The Flutter logo is visible at the top of the screen. The bottom of the phone shows a landscape scene.

The screenshot shows a development environment for Flutter. On the left, the code editor displays `main.dart` with the following code:

```
40 icon: Icons.mail_outline,
41 label: 'Email',
42 ), // LoginField
43 SizedBox(height: 16.0),
44 LoginField(
45   icon: Icons.lock_open,
46   label: 'Password',
47   obscureText: true,
48 ), // LoginField
49 SizedBox(height: 16.0),
50 Checkbox(
51   value: _isRememberMeChecked,
52   onChanged: (newValue) {
53     setState(() {
54       _isRememberMeChecked = !_isRememberMeChecked;
55     });
56   },
57 ), // Checkbox
58 ], // <Widget>[]
59 ), // Column
60 ), // Padding
61 ), // SafeArea
62 ); // Scaffold
63 }
```

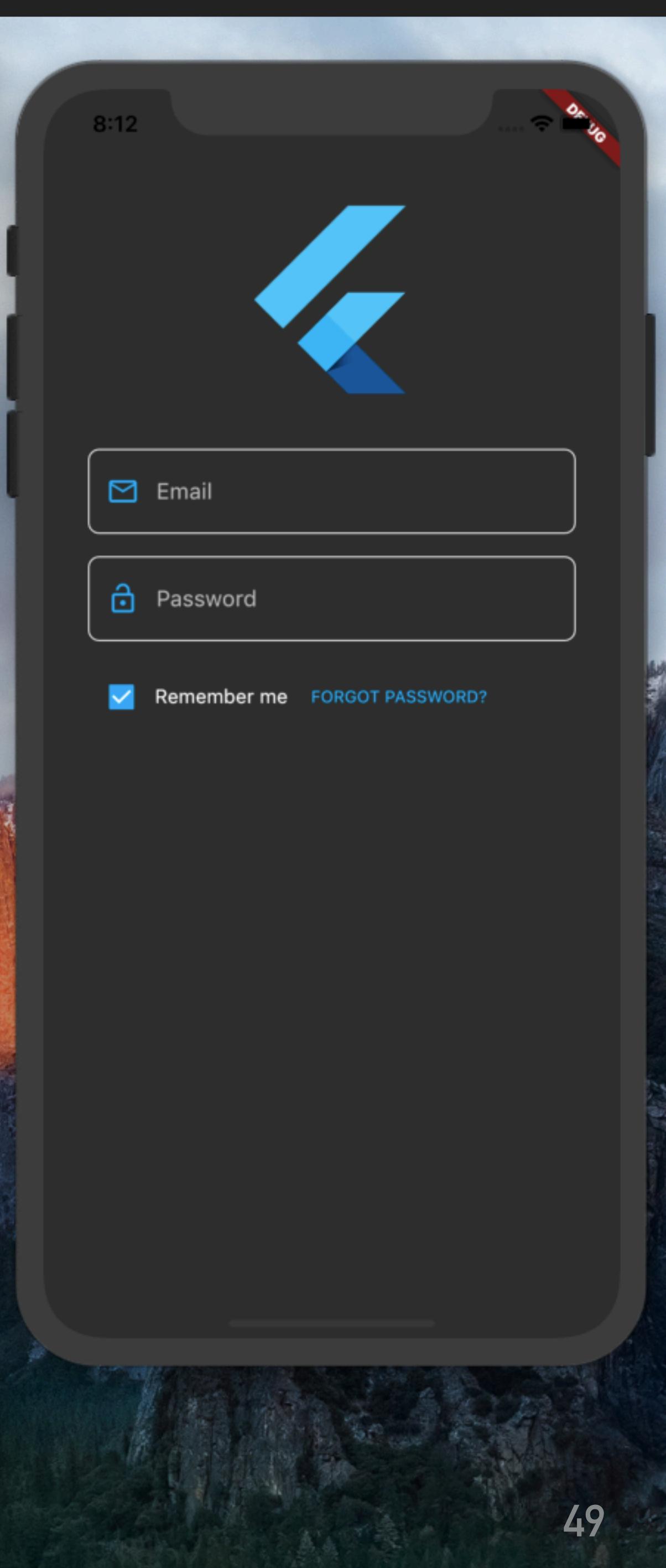
The right side of the interface shows a preview of the application running on an iPhone XR. The screen displays a login form with two text fields: "Email" and "Password". A checkbox labeled "Remember Me" is checked. The Flutter logo is visible at the top of the screen.

The screenshot shows a development environment for Flutter. On the left, the code editor displays `main.dart` with the following code:

```
44 LoginField(  
45   icon: Icons.lock_open,  
46   label: 'Password',  
47   obscureText: true,  
48 ), // LoginField  
49 SizedBox(height: 16.0),  
50 Row(  
51   children: <Widget>[  
52     Checkbox(  
53       activeColor: Colors.lightBlue,  
54       value: _isRememberMeChecked,  
55       onChanged: (newValue) {  
56         setState(() {  
57           _isRememberMeChecked = !_isRememberMeChecked;  
58         });  
59       },  
60     ), // Checkbox  
61     Text('Remember me')  
62   ], // <Widget>[]  
63 ) // Row  
64 ], // <Widget>[]  
65 ), // Column  
66 ), // Padding  
67 ), // SafeArea
```

The right side of the interface shows a preview of the Flutter application running on an iPhone XR. The screen displays a login form with two text fields ('Email' and 'Password'), a checked checkbox labeled 'Remember me', and the Flutter logo at the top.

```
49     SIZENONTEXTUALYMAPSIZETO, 10.0, 10.0
50   Row(
51     children: <Widget>[
52       Checkbox(
53         activeColor: Colors.lightBlue,
54         value: _isRememberMeChecked,
55         onChanged: (newValue) {
56           setState(() {
57             _isRememberMeChecked = !_isRememberMeChecked;
58           });
59         },
60       ), // Checkbox
61       Text('Remember me'),
62       FlatButton(
63         child: Text(
64           'Forgot password?'.toUpperCase(),
65           style: TextStyle(color: Colors.lightBlue, fontSize: 12.0),
66         ), // Text
67         onPressed: () {},
68       ) // FlatButton
69     ], // <Widget>[]
70   ) // Row
71 , // <Widget>[]
72 // Column
73 Padding
```



flutterthon [~/FlutterProjects/flutterthon] - .../lib/main.dart [flutterthon]

main.dart

```
49   SizedBox(height: 16.0),  
50   Row(  
51     mainAxisAlignment: MainAxisAlignment.spaceBetween,  
52     children: <Widget>[  
53       Checkbox(  
54         activeColor: Colors.lightBlue,  
55         value: _isRememberMeChecked,  
56         onChanged: (newValue) {  
57           setState(() {  
58             _isRememberMeChecked = !_isRememberMeChecked;  
59           });  
60         },  
61       ), // Checkbox  
62       Text('Remember me'),  
63       FlatButton(  
64         child: Text(  
65           'Forgot password?'.toUpperCase(),  
66           style: TextStyle(color: Colors.lightBlue, fontSize: 12.0  
67         ), // Text  
68         onPressed: () {},  
69       ) // FlatButton  
70     ], // <Widget>[]  
71   ) // Row  
72 ] // <Widget>[]
```

1: Project 2: Favorites 3: Z: Structure 4: Layout Captures 5: Build Variants

Resource Manager

6: Logcat 7: TODO 8: Terminal 9: Dart Analysis 10: Run 11: Messages

Event Log 12: Flutter Performance

50 chars 51:17 LF 13: UTF-8 14: 2 spaces 15: 16: 17: 18: 19: 20: 21: 22: 23: 24: 25: 26: 27: 28: 29: 30: 31: 32: 33: 34: 35: 36: 37: 38: 39: 40: 41: 42: 43: 44: 45: 46: 47: 48: 49: 50: 51: 52: 53: 54: 55: 56: 57: 58: 59: 60: 61: 62: 63: 64: 65: 66: 67: 68: 69: 70: 71: 72:

Key Promoter X Flutter Outline Flutter Inspector Device File Explorer

8:13

✉ Email

🔒 Password

Remember me FORGOT PASSWORD?

50

flutterthon [~/FlutterProjects/flutterthon] - .../lib/main.dart [flutterthon]

main.dart

```
Row(  
  children: <Widget>[  
    Checkbox(  
      activeColor: Colors.lightBlue,  
      value: _isRememberMeChecked,  
      onChanged: (newValue) {  
        setState(() {  
          _isRememberMeChecked = !_isRememberMeChecked;  
        });  
      },  
    ), // Checkbox  
    Text('Remember me'),  
    Expanded(  
      child: Container(),  
    ), // Expanded  
    FlatButton(  
      child: Text(  
        'Forgot password?'.toUpperCase(),  
        style: TextStyle(color: Colors.lightBlue, fontSize: 12.0),  
      ), // Text  
      onPressed: () {},  
    ), // FlatButton  
  ], // <Widget>[]  
) // Row
```

Resource Manager

Project

Build Variants

Layout Captures

Z: Structure

Favorites

Logcat

TODO

Terminal

Dart Analysis

Run

Messages

Event Log

Flutter Performance

64:20 LF UTF-8 2 spaces 😊 😄

8:14

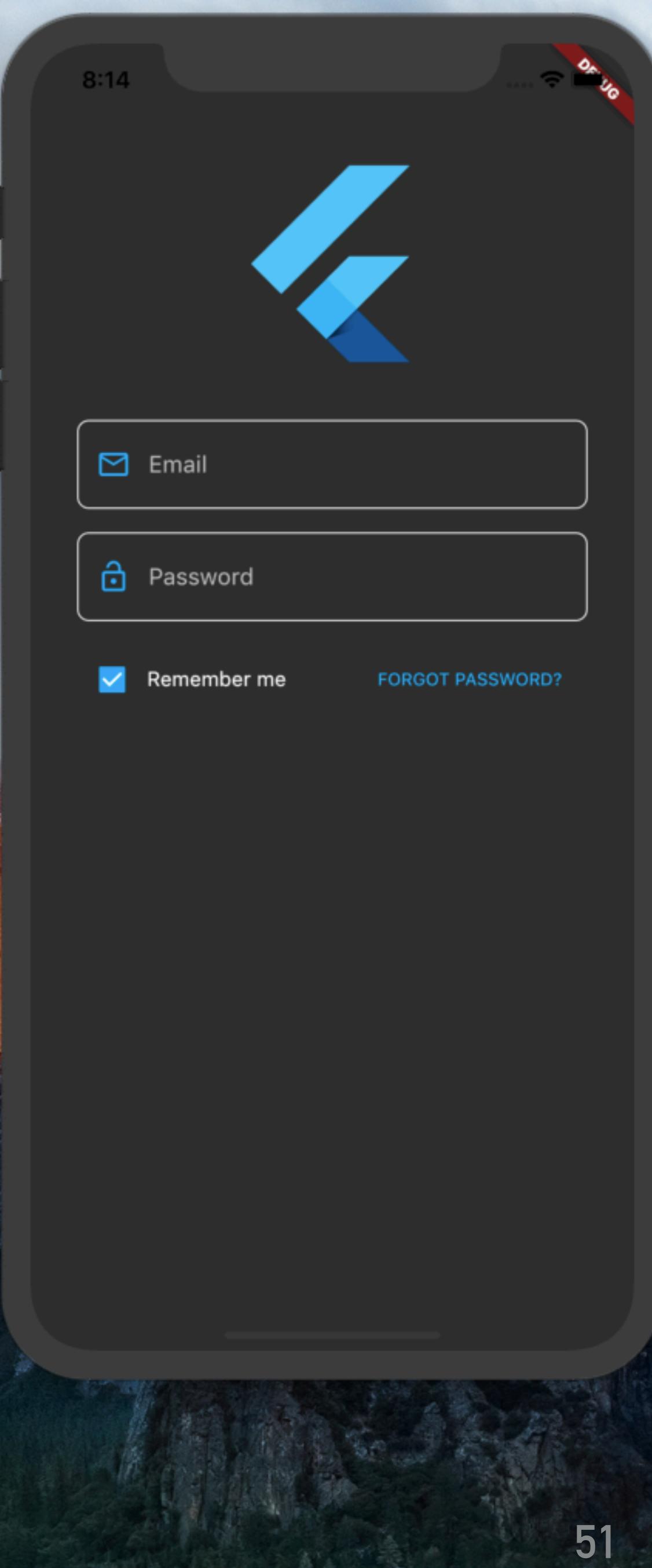
Key Promoter X

Flutter Outline

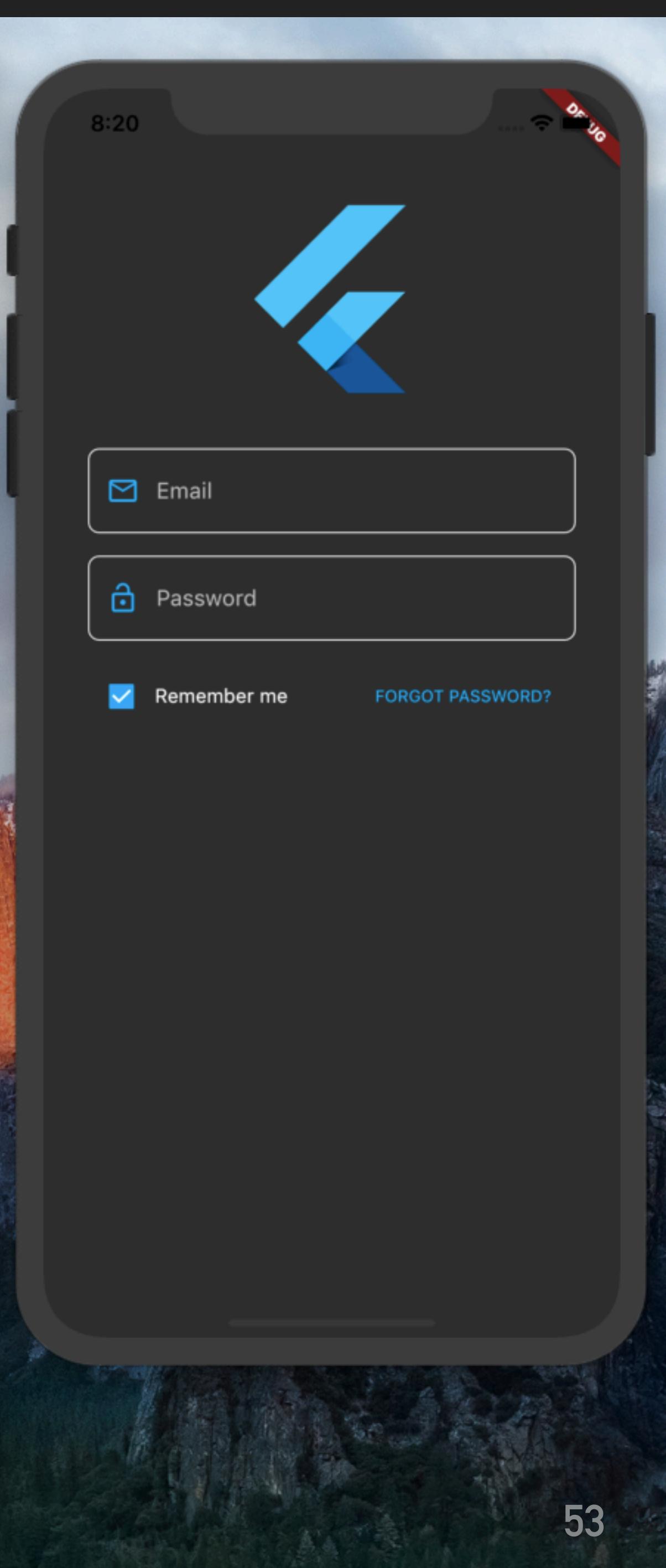
Flutter Inspector

Device File Explorer

FORGOT PASSWORD?







flutterthon [~/FlutterProjects/flutterthon] - .../lib/main.dart [flutterthon]

main.dart

```
80    ); // Scaffold
81
82  }
83
84 class LoginButton extends StatefulWidget {
85   @override
86   _LoginButtonState createState() => _LoginButtonState();
87 }
88
89 class _LoginButtonState extends State<LoginButton> {
90   @override
91   Widget build(BuildContext context) {
92     return Container(
93       decoration: BoxDecoration(color: Colors.lightBlue),
94       child: Text(
95         'LOG IN',
96         style: TextStyle(color: Colors.white),
97       ), // Text
98     ); // Container
99   }
100 }
101
102 class LoginField extends StatelessWidget {
103   LoginField({this.icon, this.label, this.obscureText = false});
```

Resource Manager

Project

Build Variants

Layout Captures

Z: Structure

Favorites

Logcat

TODO

Terminal

Dart Analysis

Run

Messages

Event Log

Flutter Performance

Key Promoter X

Flutter Outline

Flutter Inspector

Device File Explorer

8:21

Email

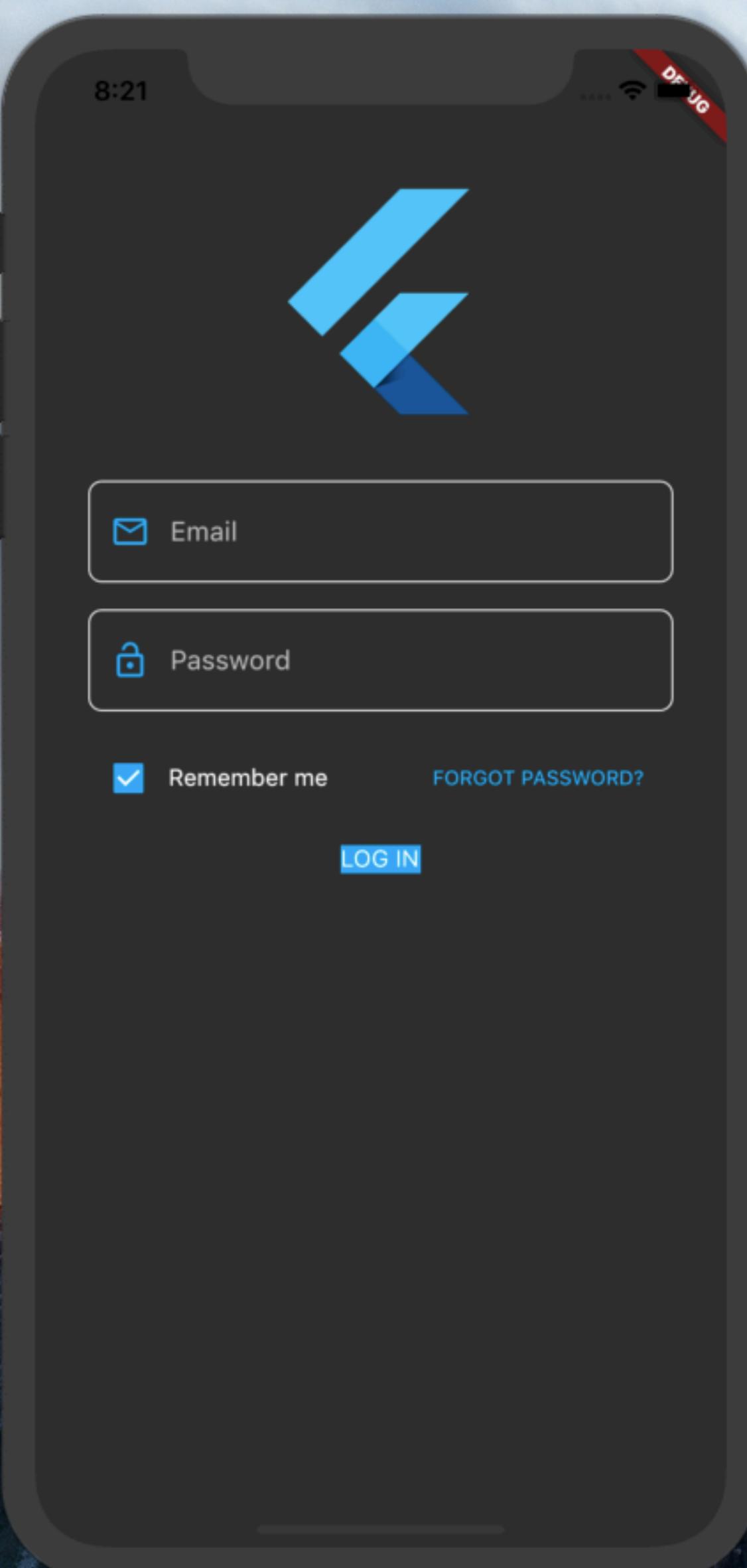
Password

Remember me

FORGOT PASSWORD?

LOG IN

54



The screenshot shows a development environment with a Dart code editor and a preview window.

Code Editor (main.dart):

```
82 }  
83  
84 class LoginButton extends StatefulWidget {  
85   @override  
86   _LoginButtonState createState() => _LoginButtonState();  
87 }  
88  
89 class _LoginButtonState extends State<LoginButton> {  
90   @override  
91   Widget build(BuildContext context) {  
92     return Container(  
93       padding: const EdgeInsets.all(16.0),  
94       decoration: BoxDecoration(  
95         color: Colors.lightBlue,  
96         borderRadius: BorderRadius.all(Radius.circular(6.0)),  
97       ), // BoxDecoration  
98       child: Text(  
99         'LOG IN',  
100        style: TextStyle(color: Colors.white),  
101      ), // Text  
102    ); // Container  
103 }  
104  
105 }
```

Preview Window:

The preview shows a Flutter mobile application running on an iPhone XR. The screen displays a login form with a large blue button labeled "LOG IN". The application also includes fields for "Email" and "Password", a "Remember me" checkbox, and a "FORGOT PASSWORD?" link.

Bottom Status Bar:

iPhone XR — 12.2

8:22

Key Promoter X

Flutter Outline

Flutter Inspector

Device File Explorer

Event Log

Flutter Performance

Logcat

TODO

Terminal

Dart Analysis

Run

Messages

97.7 LF UTF-8 2 spaces 😊 😐

55

The screenshot shows a development environment with a Dart file open in the main editor and a running Flutter application on a simulated device.

Main.dart Code:

```
83
84 class LoginButton extends StatefulWidget {
85   @override
86   _LoginButtonState createState() => _LoginButtonState();
87 }
88
89 class _LoginButtonState extends State<LoginButton> {
90   @override
91   Widget build(BuildContext context) {
92     return Container(
93       padding: const EdgeInsets.all(16.0),
94       decoration: BoxDecoration(
95         boxShadow: [
96           BoxShadow(
97             color: Colors.lightBlue[800],
98             offset: Offset(0.0, 8.0),
99             blurRadius: 12.0,
100            ), // BoxShadow
101          ],
102          color: Colors.lightBlue,
103          borderRadius: BorderRadius.all(Radius.circular(6.0)),
104        ), // BoxDecoration
105        child: Text(
106          'LOG IN',
107          style: TextStyle(color: Colors.white),
108        ),
109      );
110    }
111 }
```

Flutter Application Preview:

The application displays a login screen with the following elements:

- A large blue Flutter logo at the top.
- Two input fields: "Email" and "Password".
- A "Remember me" checkbox.
- A "FORGOT PASSWORD?" link.
- A prominent blue "LOG IN" button at the bottom right.

The status bar indicates the time is 8:27 and the battery level is high.

The screenshot shows a development environment for Flutter. On the left, the code editor displays `main.dart` with the following code:

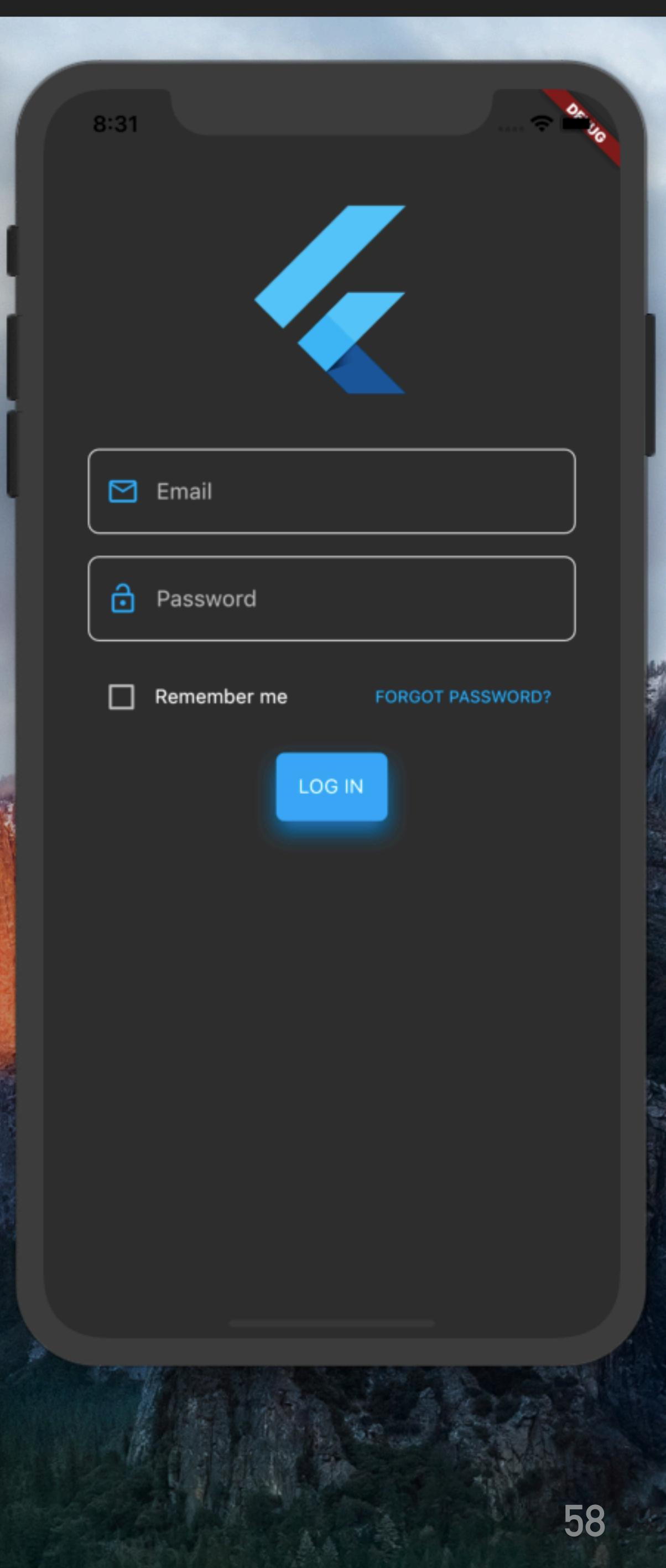
```
89
90 class _LoginButtonState extends State<LoginButton> {
91   @override
92   Widget build(BuildContext context) {
93     return GestureDetector(
94       onTapDown: (details) {},
95       onTapUp: (details) {},
96       onTapCancel: () {},
97       child: Container(
98         padding: const EdgeInsets.all(16.0),
99         decoration: BoxDecoration(
100           boxShadow: [
101             BoxShadow(
102               color: Colors.lightBlue[800],
103               offset: Offset(0.0, 8.0),
104               blurRadius: 12.0,
105             ), // BoxShadow
106           ],
107           color: Colors.lightBlue,
108           borderRadius: BorderRadius.all(Radius.circular(6.0)),
109         ), // BoxDecoration
110         child: Text(
111           'LOG IN',
112           style: TextStyle(color: Colors.white),
113         ), // Text
114       ), // Container
115     ); // GestureDetector
116   } // build
117 } // _LoginButtonState
118
119
120
121
122
123
```

The right side of the interface shows a preview of the application running on an iPhone XR. The screen displays a login form with fields for Email and Password, a Remember me checkbox, a FORGOT PASSWORD? link, and a large blue LOG IN button. The Flutter logo is visible at the top of the screen.

```
flutterthon [~/FlutterProjects/flutterthon] - .../lib/main.dart [flutterthon]
main.dart

83
84 class LoginButton extends StatelessWidget {
85     @override
86     _LoginButtonState createState() => _LoginButtonState();
87 }
88
89 class _LoginButtonState extends State<LoginButton> {
90     int _buttonShade = 500;
91     int _shadowShade = 800;
92
93     void _setButtonState({@required bool isPressed}) {
94         setState(() {
95             _buttonShade = isPressed ? 600 : 500;
96             _shadowShade = isPressed ? 900 : 800;
97         });
98     }
99
100    @override
101    Widget build(BuildContext context) {
102        return GestureDetector(
103            onTapDown: (details) {},
104            onTapUp: (details) {},
105            onTapCancel: () {},
106            child: Container(

```



The screenshot shows a development environment for Flutter. On the left, the code editor displays `main.dart` with the following code:

```
102 return GestureDetector(
103   onTapDown: (details) {},
104   onTapUp: (details) {},
105   onTapCancel: () {},
106   child: Container(
107     padding: const EdgeInsets.all(16.0),
108     decoration: BoxDecoration(
109       boxShadow: [
110         BoxShadow(
111           color: Colors.lightBlue[_shadowShade],
112           offset: Offset(0.0, 8.0),
113           blurRadius: 12.0,
114         ), // BoxShadow
115       ],
116       color: Colors.lightBlue[_buttonShade],
117       borderRadius: BorderRadius.all(Radius.circular(6.0)),
118     ), // BoxDecoration
119     child: Text(
120       'LOG IN',
121       style: TextStyle(color: Colors.white),
122     ), // Text
123   ), // Container
124 ); // GestureDetector
125
126 }
```

The code uses `BoxShadow` and `BorderRadius` to style a button. The `color` properties for both the shadow and the button itself are highlighted in blue, indicating they are selected or being edited.

On the right, a preview window shows the resulting UI on an iPhone XR. The screen displays a login form with fields for Email and Password, a Remember me checkbox, a FORGOT PASSWORD? link, and a large blue **LOG IN** button. The background features a blurred landscape image. The top right corner of the phone screen shows the word "DEBUG".

The screenshot shows a development environment for Flutter. On the left, the code editor displays `main.dart` with the following code:

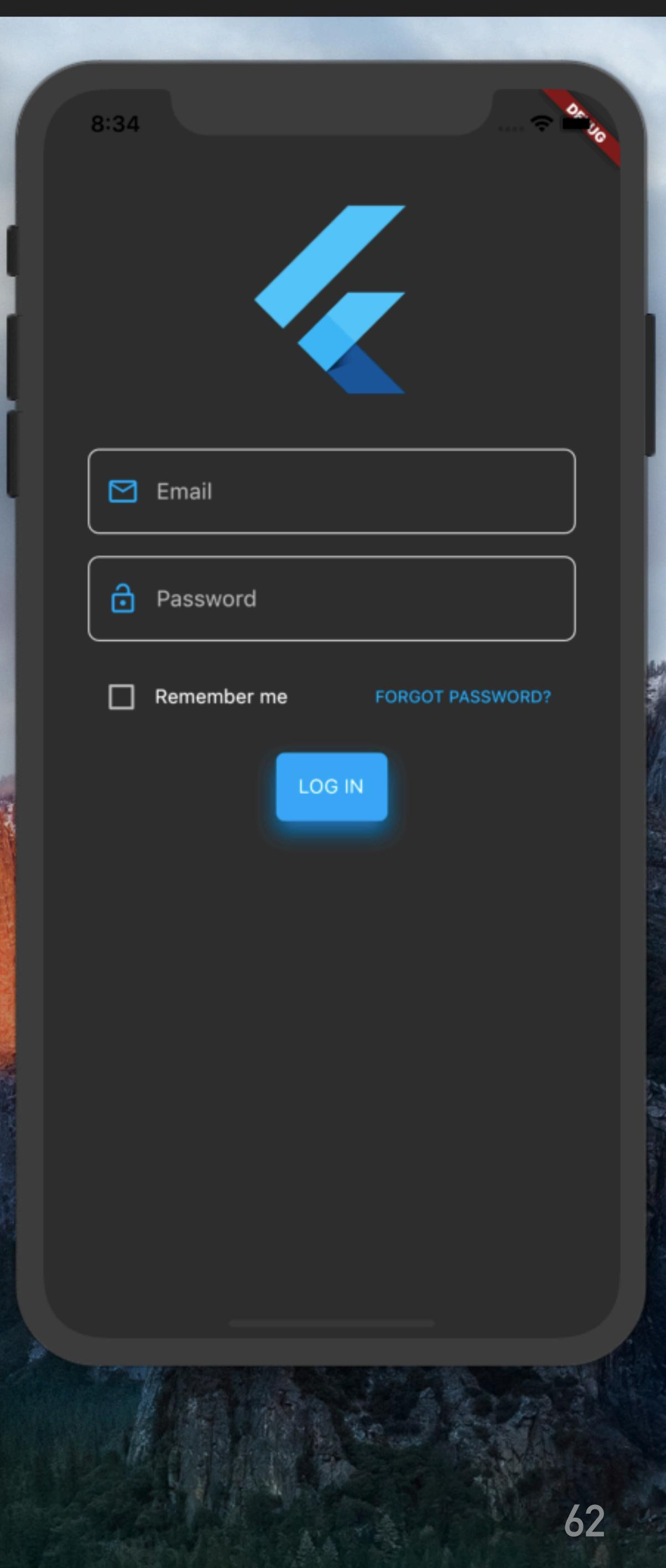
```
98 }  
99  
100 @override  
101 Widget build(BuildContext context) {  
102   return GestureDetector(  
103     onTapDown: (details) {  
104       _setButtonState(isPressed: true);  
105     },  
106     onTapUp: (details) {  
107       _setButtonState(isPressed: false);  
108     },  
109     onTapCancel: () {  
110       _setButtonState(isPressed: false);  
111     },  
112     child: Container(  
113       padding: const EdgeInsets.all(16.0),  
114       decoration: BoxDecoration(  
115         boxShadow: [  
116           BoxShadow(  
117             color: Colors.lightBlue[_shadowShade],  
118             offset: Offset(0.0, 8.0),  
119             blurRadius: 12.0,  
120           ), // BoxShadow  
121         ],  
122       ),  
123     ),  
124   );  
125 }  
126  
127 void _setButtonState({  
128   required bool isPressed,  
129 }  
130 ) {  
131 }
```

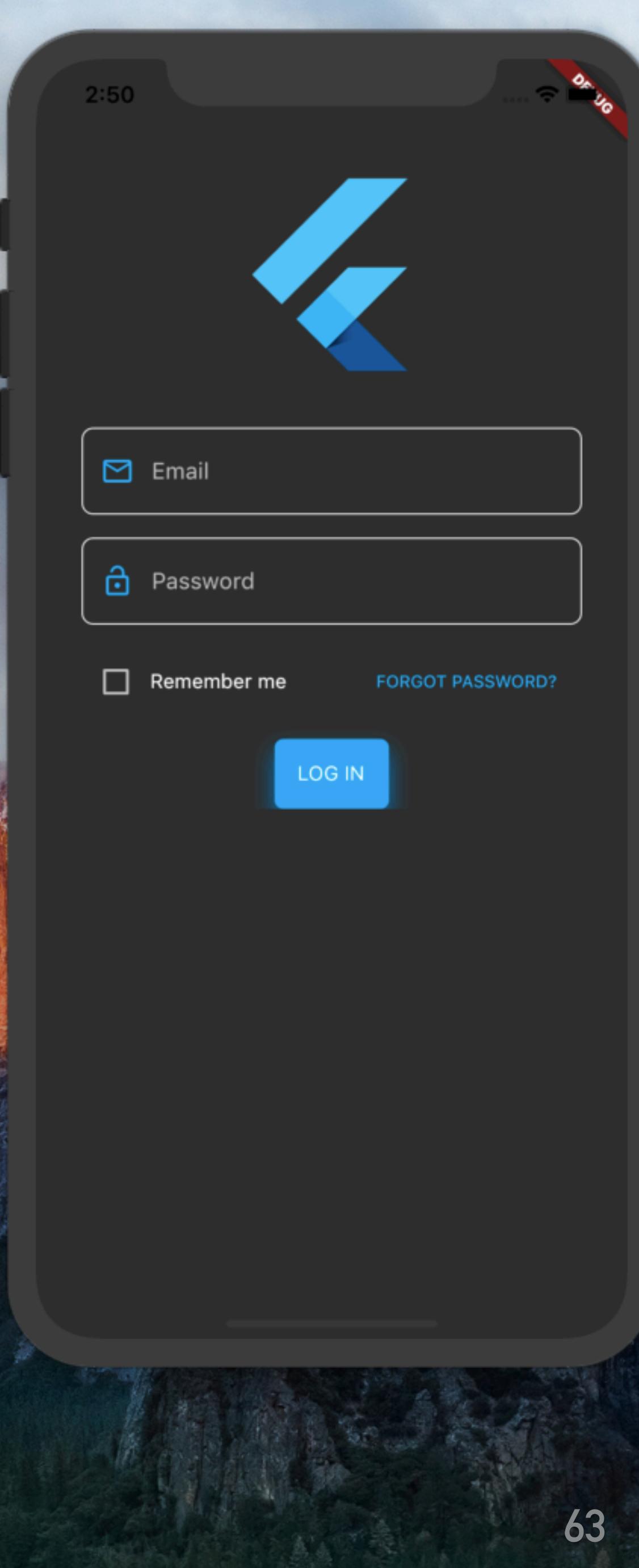
On the right, a mobile application is running on an iPhone XR. The screen shows a login form with fields for Email and Password, a Remember me checkbox, a FORGOT PASSWORD? link, and a blue LOG IN button. The Flutter logo is visible at the top of the screen.



```
flutterthon [~/FlutterProjects/flutterthon] - .../lib/main.dart [flutterthon]
main.dart
Resource Manager
Project
1: Project
main.dart x
Key Promoter X
Flutter Outline
Flutter Inspector
Build Variants
Layout Captures
Structure
2: Favorites
2: Favorites
Device File Explorer
6: Logcat
TODO
Terminal
Dart Analysis
Run
0: Messages
Event Log
Flutter Performance
89:3 LF UTF-8 2 spaces
main.dart
78     ), // Padding
79     ), // SafeArea
80   ); // Scaffold
81 }
82 }
83
84 class LoginButton extends StatelessWidget {
85   LoginButton({@required this.onPressed, @required this.child});
86
87   final VoidCallback onPressed;
88   final Widget child;
89
90   @override
91   _LoginButtonState createState() => _LoginButtonState();
92 }
93
94 class _LoginButtonState extends State<LoginButton> {
95   int _buttonShade = 500;
96   int _shadowShade = 800;
97
98   void _setButtonState({@required bool isPressed}) {
99     setState(() {
100       _buttonShade = isPressed ? 600 : 500;
101       _shadowShade = isPressed ? 900 : 800;

```





A screenshot of a Flutter application running on an iPhone XR device. The application displays a login screen with the following interface elements:

- A large blue Flutter logo at the top.
- Two input fields: "Email" and "Password".
- A "Remember me" checkbox and a "FORGOT PASSWORD?" link.
- A prominent blue "LOG IN" button.

The code for this screen is visible in the main.dart file of the project:

```
123 } ,  
124 onTapCancel: () {  
125     _setButtonState(isPressed: false);  
126 },  
127 onTap: onPressed,|  
128 child: Container(  
129     padding: const EdgeInsets.all(16.0),  
130     decoration: BoxDecoration(  
131         boxShadow: [  
132             BoxShadow(  
133                 color: Colors.lightBlue[_shadowShade],  
134                 offset: Offset(0.0, 8.0),  
135                 blurRadius: 12.0,  
136             ), // BoxShadow  
137         ],  
138         color: Colors.lightBlue[_buttonShade],  
139         borderRadius: BorderRadius.all(Radius.circular(6.0)),  
140     ), // BoxDecoration  
141     child: child,  
142 ), // Container  
143 ); // GestureDetector  
144 }  
145 }  
146 }
```

The code uses a Gesture Detector to handle tap events on the container, which contains the input fields and the main body of the screen. The container is styled with a box shadow and rounded corners.

The screenshot shows a development environment for Flutter. On the left, the code editor displays `main.dart` with the following code:

```
116 },  
117     _setButtonState(isPressed: false),  
118 },  
119     onTapCancel: () {  
120         _setButtonState(isPressed: false);  
121     },  
122     onTap: widget.onPressed,  
123     child: Container(  
124         padding: const EdgeInsets.all(16.0),  
125         decoration: BoxDecoration(  
126             boxShadow: [  
127                 BoxShadow(  
128                     color: Colors.lightBlue[_shadowShade],  
129                     offset: Offset(0.0, 8.0),  
130                     blurRadius: 12.0,  
131                 ), // BoxShadow  
132             ],  
133             color: Colors.lightBlue[_buttonShade],  
134             borderRadius: BorderRadius.all(Radius.circular(6.0)),  
135         ), // BoxDecoration  
136         child: widget.child,  
137     ), // Container  
138 ); // GestureDetector  
139 }  
140 }  
141 }  
142 }
```

The right side of the interface shows a preview of the application running on an iPhone XR. The screen displays a login form with fields for Email and Password, a Remember me checkbox, a FORGOT PASSWORD? link, and a blue LOG IN button. The Flutter logo is visible at the top of the screen.

The screenshot shows an IDE interface with a Flutter project named 'flutterthon'. The main.dart file is open, displaying the Dart code for a login screen. The code includes a Row with two TextInputs for Email and Password, a Remember me checkbox, a FORGOT PASSWORD? link, and a large blue LOGIN button. The Flutter Outline and Flutter Inspector panels are visible on the right, and a Device File Explorer panel is at the bottom right. The bottom navigation bar includes tabs for Logcat, TODO, Terminal, Dart Analysis, Run, Messages, Event Log, and Flutter Performance. The status bar at the bottom shows the time as 81:16, encoding as LF, encoding as UTF-8, and font size as 2 spaces.

```
64      ), // Expanded
65      FlatButton(
66        child: Text(
67          'Forgot password?'.toUpperCase(),
68          style: TextStyle(color: Colors.lightBlue, font
69        ), // Text
70        onPressed: () {},
71      ), // FlatButton
72    ], // <Widget>[]
73  ), // Row
74  SizedBox(height: 16.0),
75  LoginButton(
76    child: Text(
77      'LOG IN',
78      style: TextStyle(color: Colors.white),
79    ), // Text
80    onPressed: () {},
81  ) // LoginButton
82  ], // <Widget>[]
83  ), // Column
84  ), // Padding
85  ), // SafeArea
86 ); // Scaffold
87 }
```

The screenshot shows a development environment for Flutter. On the left, the code editor displays `main.dart` with the following code:

```
20
21 class _LoginScreenState extends State<LoginScreen> {
22   bool _isRememberMeChecked = false;
23
24   @override
25   Widget build(BuildContext context) {
26     return Scaffold(
27       body: SafeArea(
28         child: Padding(
29           padding: const EdgeInsets.symmetric(horizontal: 32.0),
30           child: Column(
31             crossAxisAlignment: CrossAxisAlignment.stretch,
32             children: <Widget>[
33               Padding(
34                 padding: const EdgeInsets.symmetric(vertical: 32.0),
35                 child: Image.asset(
36                   'images/flutter.png',
37                   height: 150.0,
38                   width: 150.0,
39                 ), // Image.asset
40               ), // Padding
41               LoginField(
42                 icon: Icons.mail_outline,
43                 label: 'Email',
44               ), // LoginField
45             ],
46           ),
47         ),
48       ),
49     );
50   }
51 }
```

The right side of the interface shows a preview of the Flutter application running on an iPhone XR. The screen displays a login form with a large Flutter logo at the top, followed by fields for Email and Password, a Remember me checkbox, a FORGOT PASSWORD? link, and a blue LOG IN button.

The screenshot shows a development environment for Flutter. On the left, the code editor displays `main.dart` with the following code:

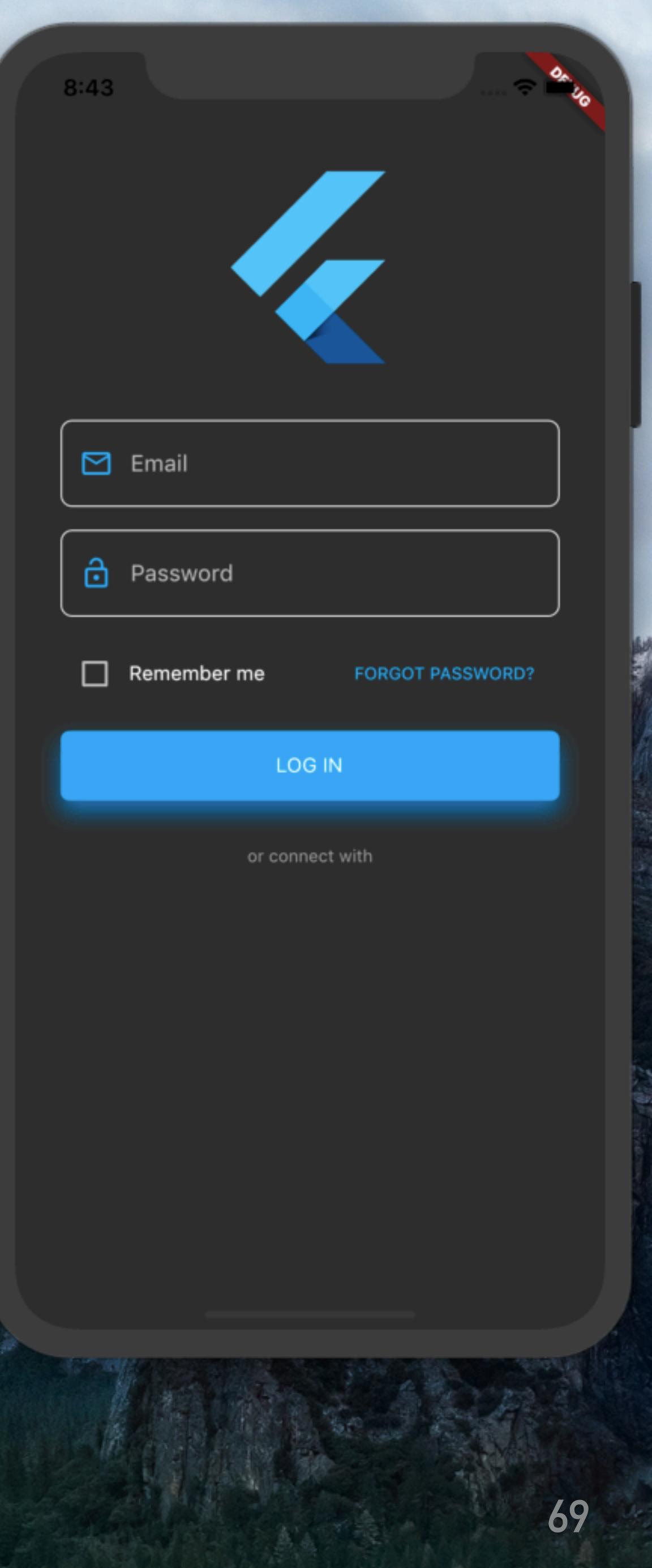
```
125: Container(
126:   padding: const EdgeInsets.all(16.0),
127:   decoration: BoxDecoration(
128:     boxShadow: [
129:       BoxShadow(
130:         color: Colors.lightBlue[_shadowShade],
131:         offset: Offset(0.0, 8.0),
132:         blurRadius: 12.0,
133:       ), // BoxShadow
134:     ],
135:     color: Colors.lightBlue[_buttonShade],
136:     borderRadius: BorderRadius.all(Radius.circular(6.0)),
137:   ), // BoxDecoration
138:   child: Center(child: widget.child),
139: ), // Container
140: ); // GestureDetector
141:
142:
143:
144: class LoginField extends StatelessWidget {
145:   LoginField({this.icon, this.label, this.obscureText = false});
146:
147:   final IconData icon;
148:   final String label;
149:   final bool obscureText;
```

The right side of the interface shows a preview of the application running on an iPhone XR. The screen displays a login form with fields for Email and Password, a Remember me checkbox, a FORGOT PASSWORD? link, and a large blue LOG IN button. The Flutter logo is visible at the top of the screen.

The screenshot shows a development environment for a Flutter application named "flutterthon". The main window displays the Dart code for the application's entry point, `main.dart`. The code defines a `Scaffold` with a `SafeArea`, `Padding`, and a `Column` containing a `LoginButton` and a `Center` widget. The `LoginButton` has a child `Text` with the text "LOG IN". The `Center` widget contains a `Text` with the text "or connect with". The right side of the interface shows a preview of the application running on an iPhone XR. The app features a dark theme with a large blue "LOG IN" button at the bottom. The top of the screen shows the Flutter logo and the word "DEBUG". The bottom of the screen shows a "FORGOT PASSWORD?" link and a "Remember me" checkbox.

```
75     SizedBox(height: 16.0),  
76     LoginButton(  
77       child: Text(  
78         'LOG IN',  
79         style: TextStyle(color: Colors.white),  
80       ), // Text  
81       onPressed: () {},  
82     ), // LoginButton  
83     SizedBox(  
84       height: 32.0,  
85     ), // SizedBox  
86     Center(  
87       child: Text(  
88         'or connect with',  
89         style: TextStyle(color: Colors.grey, fontSize: 12.0),  
90       ), // Text  
91     ), // Center  
92   ], // <Widget>[]  
93   ), // Column  
94   ), // Padding  
95   ), // SafeArea  
96 ); // Scaffold  
97  
98  
99
```

A screenshot of an IDE (Android Studio) displaying a Flutter application. The left side shows the code editor with 'main.dart' open, containing Dart code for a login screen. A tooltip is visible over the 'Icons.face' icon. The right side shows a preview of the app running on an iPhone XR, displaying a login form with fields for Email and Password, a Remember me checkbox, a FORGOT PASSWORD? link, and a large blue LOG IN button. The Flutter logo is visible in the top right corner of the app's header.



The screenshot shows the pub.dev website, specifically the 'Flutter Packages' section. At the top, there's a navigation bar with icons for back, forward, search, and refresh, followed by the URL 'https://pub.dev/flutter'. The main header is 'Flutter Packages' with a blue icon. To the right are 'Getting Started' dropdowns for 'Flutter' and 'Web & Server'. Below the header is a search bar with placeholder text 'Search Flutter-compatible packages' and a magnifying glass icon. Underneath the search bar are three tabs: 'FLUTTER' (selected), 'WEB', and 'ALL'. A promotional message below the tabs reads: 'Flutter makes it easy and fast to build beautiful mobile apps for iOS and Android.' The main content area is titled 'Top Flutter-compatible packages' and lists ten packages in two columns:

| Package | Type | Description |
|--|---------|--|
| path_provider | FLUTTER | Flutter plugin for getting commonly used locations on the Android & iOS file systems, such as the temp and app data directories. |
| sqflite | FLUTTER | Flutter plugin for SQLite, a self-contained, high-reliability, embedded, SQL database engine. |
| provider | FLUTTER | A dependency injection system built with widgets for widgets. provider is mostly syntax sugar for InheritedWidget, to make common use-cases straightforward. |
| package_info | FLUTTER | Flutter plugin for querying information about the application package, such as CFBundleVersion on iOS or versionCode on |
| url_launcher | FLUTTER | Flutter plugin for launching a URL on Android and iOS. Supports web, phone, SMS, and email schemes. |
| cached_network_image | FLUTTER | Flutter library to load and cache network images. Can also be used with placeholder and error widgets. |
| scoped_model | FLUTTER | A Widget that passes a Reactive Model to all of its children |
| flutter_webview_plugin | FLUTTER | Plugin that allows Flutter to communicate with a native Webview. |
| firebase_storage | FLUTTER | Flutter plugin for Firebase Cloud Storage, a powerful, simple, and cost-effective object storage service for Android and iOS. |
| device_info | FLUTTER | Flutter plugin providing detailed information about the device (make, model, etc.), and Android or iOS version the app is running on. |

At the bottom right corner, there's a page number '70'.

pub.dev

Search Flutter-compatible packages

font_awesome_flutter 8.4.0

Published Feb 4, 2019 FLUTTER

Readme Changelog Example Installing Versions 98

Use this package as a library

1. Depend on it

Add this to your package's pubspec.yaml file:

```
dependencies:  
  font_awesome_flutter: ^8.4.0
```

2. Install it

You can install packages from the command line:

with Flutter:

```
$ flutter pub get
```

Alternatively, your editor might support `flutter pub get`. Check the docs for your editor to learn more.

3. Import it

Now in your Dart code, you can use:

```
import 'package:font_awesome_flutter/font_awesome_flutter.dart';
```

About
The Font Awesome Icon pack available as Flutter Icons
[Repository \(GitHub\)](#)
[View/report issues](#)
[API reference](#)

Author
 [Brian Egan](#)
Uploader
 brian@brianegan.com
License
[MIT \(LICENSE.md\)](#)
Dependencies
[flutter](#)
More
[Packages that depend on font_awesome_flutter](#)

71

flutterthon [~/FlutterProjects/flutterthon] - .../pubspec.yaml [flutterthon]

Project flutterthon 1: pubspec.yaml

Resource Manager

1: Project 1: pubspec.yaml

Flutter commands

Packages get Packages upgrade | Flutter upgrade | Flutter doctor

Key Promoter X

Flutter Inspector

Flutter Outline

Device File Explorer

19 dependencies:

20 flutter:

21 sdk: flutter

22

23 *# The following adds the Cupertino Icons font to your application.*

24 *# Use with the CupertinoIcons class for iOS style icons.*

25 cupertino_icons: ^0.1.2

26 fontAwesome_flutter: ^8.4.0

27

28 dev_dependencies:

29 flutter_test:

30 sdk: flutter

31

32

33 *# For information on the generic Dart part of this file, see the*

34 *# following page: <https://www.dartlang.org/tools/pub/pubspec>*

35

36 *# The following section is specific to Flutter.*

37 flutter:

38

39 *# The following line ensures that the Material Icons font is*

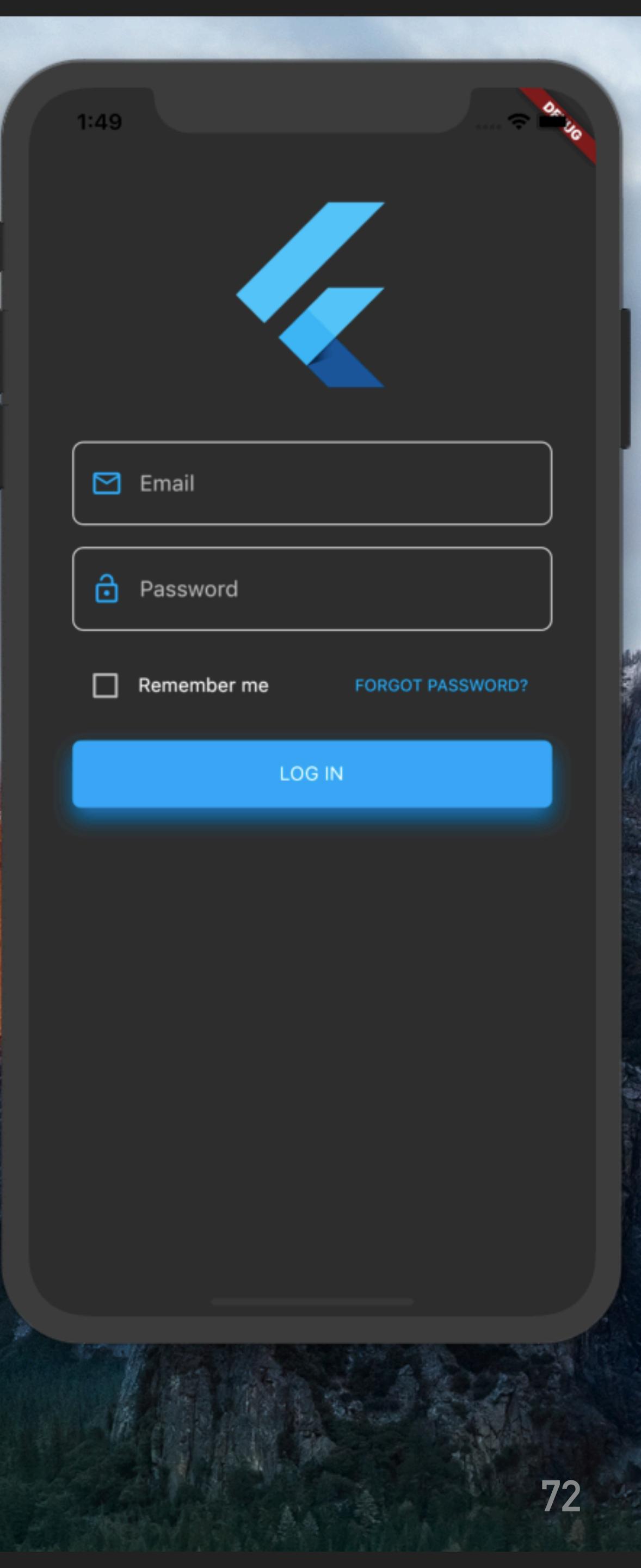
40 *# included with your application, so that you can use the icons in*

41 *# the material Icons class.*

Document 1/1 > dependencies: > fontAwesome_flutter:

Dart Analysis 4: Run 5: TODO 6: Logcat 7: Terminal 8: Version Control Event Log Flutter Performance

28 chars 26:3 LF 8: UTF-8 No JSON schema 2 spaces Git: master



The screenshot shows a development environment for Flutter. On the left, the code editor displays `main.dart` with the following content:

```
import 'package:flutter/material.dart';
import 'package:font_awesome_flutter/font_awesome_flutter.dart';

void main() => runApp(MyApp());

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      theme: ThemeData.dark(),
      home: LoginScreen(),
    ); // MaterialApp
  }
}

class LoginScreen extends StatefulWidget {
  @override
  _LoginScreenState createState() => _LoginScreenState();
}

class _LoginScreenState extends State<LoginScreen> {
  bool _isRememberMeChecked = false;

  @override
```

The code editor interface includes tabs for `main.dart`, `Resource Manager`, `Project`, `Build Variants`, `Layout Captures`, `Structure`, `Favorites`, and `Messages`. The bottom status bar shows `6: Logcat`, `TODO`, `Terminal`, `Dart Analysis`, `Run`, `0: Messages`, `Event Log`, `Flutter Performance`, and system status like `2:65 LF`, `UTF-8`, `2 spaces`.

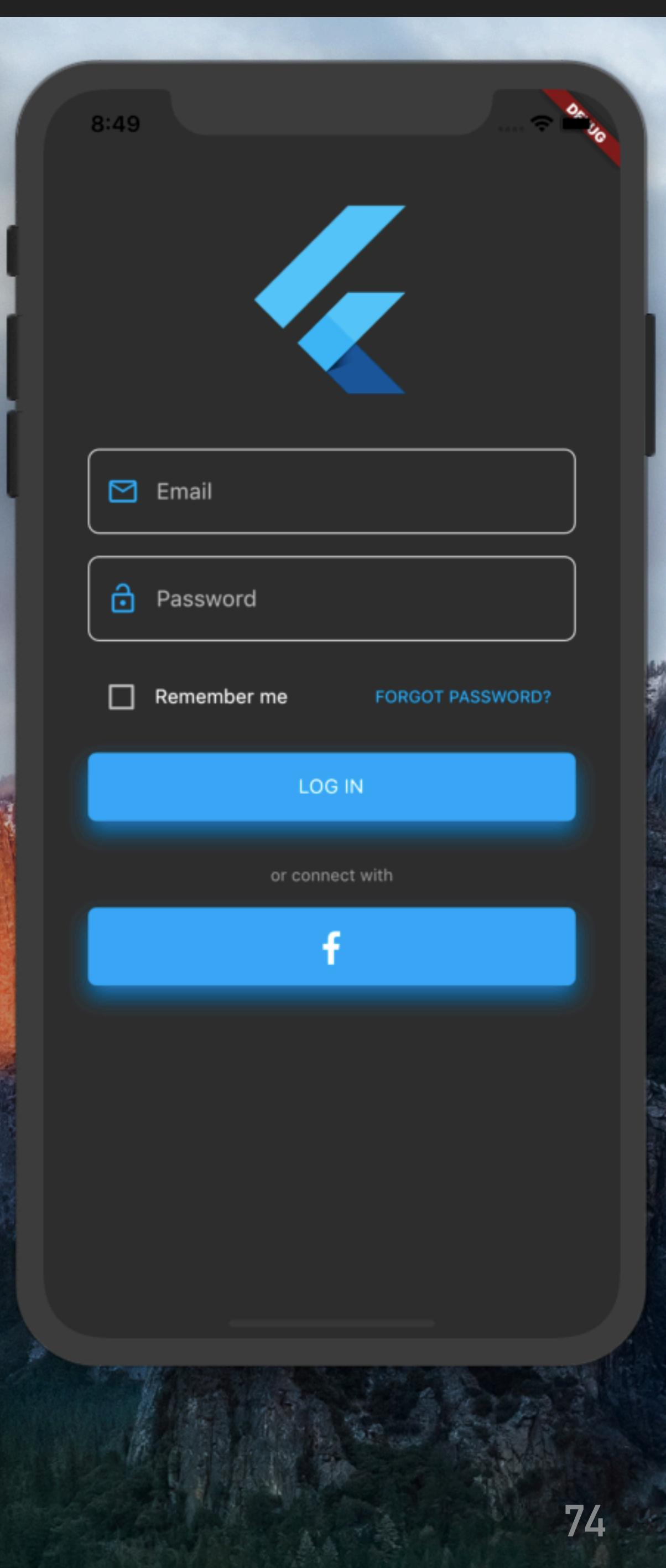
On the right, a preview window shows the `LoginScreen` on an `iPhone XR`. The screen features a large blue `K` logo at the top. Below it are two input fields: one for `Email` and one for `Password`. There is a `Remember me` checkbox, a `FORGOT PASSWORD?` link, and a large blue `LOG IN` button. A note at the bottom says `or connect with`.

```
flutterthon [~/FlutterProjects/flutterthon] - .../lib/main.dart [flutterthon]
main.dart
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116

  ),
  // center
  SizedBox(
    height: 16.0,
  ),
  // SizedBox
  LoginButton(
    child: Icon(FontAwesomeIcons.facebookF),
    onPressed: () {},
  ),
  // LoginButton
  SizedBox(
    height: 16.0,
  ),
  // SizedBox
  ],
  // <Widget>[]
),
// Column
),
// Padding
),
// SafeArea
);
// Scaffold
}

class LoginButton extends StatelessWidget {
LoginButton({@required this.onPressed, @required this.child});

final VoidCallback onPressed;
final Widget child;
```



The screenshot shows a development environment for a Flutter application named 'flutterthon'. The main window displays the Dart code for the application's entry point, `main.dart`. The code defines a UI structure using `Column`, `Row`, `SizedBox`, and `LoginButton` widgets. The `LoginButton` widget uses icons from the `FontAwesomeIcons` library to represent social media logins (Facebook, Google Plus, Twitter). The code is annotated with line numbers from 92 to 116.

```
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
```

The right side of the interface shows a preview of the mobile application running on an iPhone XR. The app features a dark theme with a large blue Flutter logo at the top. Below it is a form with fields for 'Email' and 'Password', a 'Remember me' checkbox, and a 'FORGOT PASSWORD?' link. A large blue 'LOG IN' button is centered below the form. At the bottom, there are three social media connection buttons labeled 'f', 'G+', and 'twitter'.

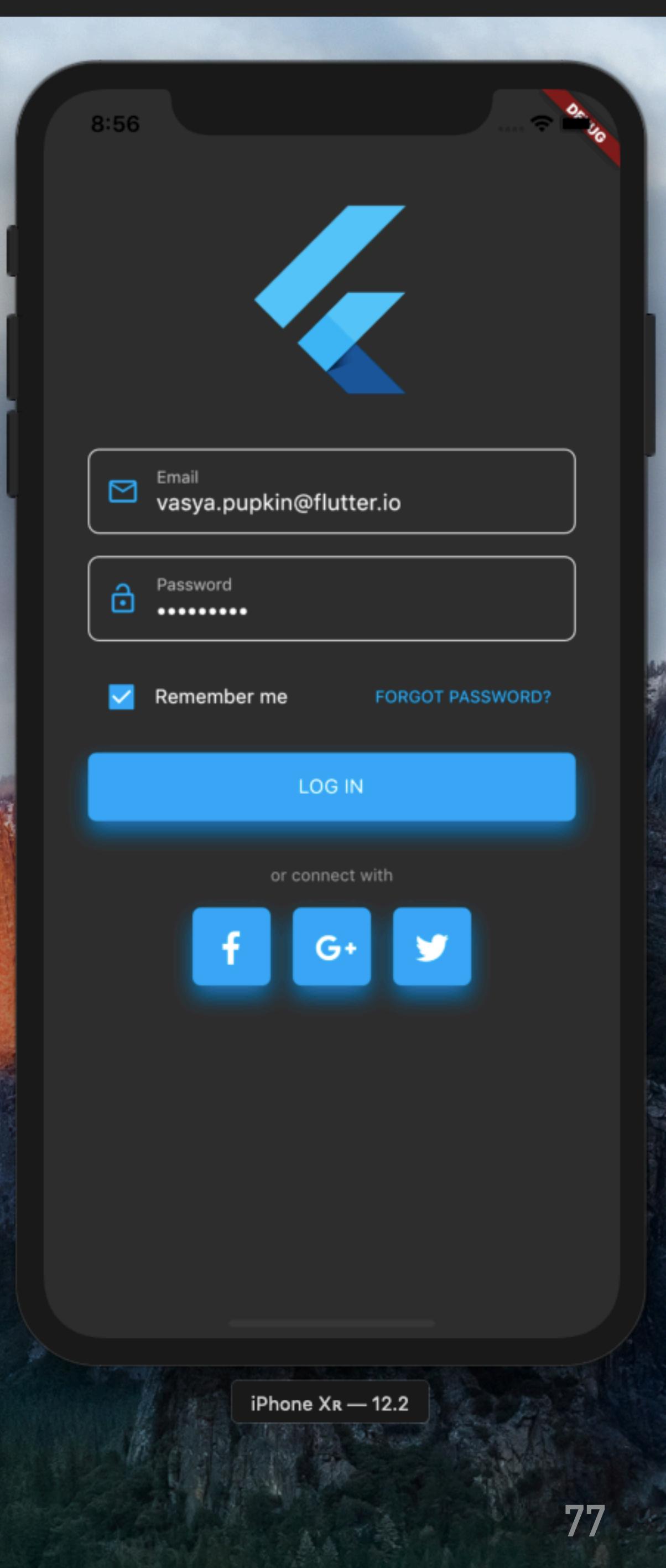
The screenshot shows an IDE interface with a Flutter project named "flutterthon". The main window displays the Dart file "main.dart" containing the following code:

```
96 Row(  
97   mainAxisAlignment: MainAxisAlignment.center,  
98   children: <Widget>[  
99     LoginButton(  
100       child: Icon(FontAwesomeIcons.facebookF),  
101       onPressed: () {},  
102     ), // LoginButton  
103     LoginButton(  
104       child: Icon(FontAwesomeIcons.googlePlusG),  
105       onPressed: () {},  
106     ), // LoginButton  
107     LoginButton(  
108       child: Icon(FontAwesomeIcons.twitter),  
109       onPressed: () {},  
110     ), // LoginButton  
111   ], // <Widget>[]  
112 ), // Row  
113 SizedBox(  
114   height: 16.0,  
115 ), // SizedBox  
116 ], // <Widget>[]  
117 ), // Column  
118 ), // Padding  
119 ), // SafeArea
```

The right side of the interface shows a preview of the Flutter application running on an iPhone XR. The app features a dark theme with a large blue Flutter logo at the top. Below it is a form with fields for "Email" and "Password", a "Remember me" checkbox, and a "FORGOT PASSWORD?" link. A large blue "LOG IN" button is at the bottom. Below the log-in section, there's a "or connect with" button followed by icons for Facebook, Google+, and Twitter.

```
flutterthon [~/FlutterProjects/flutterthon] - .../lib/main.dart [flutterthon]
main.dart
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119

  // DIZENBOX
Row(
  mainAxisAlignment: MainAxisAlignment.center,
  children: <Widget>[
    LoginButton(
      child: Icon(FontAwesomeIcons.facebookF),
      onPressed: () {},
    ), // LoginButton
    SizedBox(width: 16.0),
    LoginButton(
      child: Icon(FontAwesomeIcons.googlePlusG),
      onPressed: () {},
    ), // LoginButton
    SizedBox(width: 16.0),
    LoginButton(
      child: Icon(FontAwesomeIcons.twitter),
      onPressed: () {},
    ), // LoginButton
  ], // <Widget>[]
), // Row
SizedBox(
  height: 32.0,
), // SizedBox
], // <Widget>[]
) // Column
```



```
15  
16 class LoginScreen extends StatefulWidget {  
17     @override  
18     _LoginScreenState createState() => _LoginScreenState();  
19 }  
20  
21 class _LoginScreenState extends State<LoginScreen> {  
22     bool _isRememberMeChecked = false;  
23  
24     @override  
25     Widget build(BuildContext context) {  
26         return Scaffold(  
27             body: SafeArea(  
28                 child: Padding(  
29                     padding: const EdgeInsets.symmetric(horizontal: 32.0),  
30                     child: Column(  
31                         crossAxisAlignment: CrossAxisAlignment.stretch,  
32                         children: <Widget>[  
33                             Padding(  
34                                 padding: const EdgeInsets.symmetric(vertical: 32.0),  
35                                 child: Image.asset(  
36                                     'images/flutter.png',  
37                                     height: 150.0,  
38                                     width: 150.0,
```

flutterthon [~/FlutterProjects/flutterthon] - .../lib/main.dart [flutterthon]

main.dart

```
15
16 class LoginScreen extends StatefulWidget {
17   @override
18   _LoginScreenState createState() => _LoginScreenState();
19 }
20
21 class _LoginScreenState extends State<LoginScreen> {
22   bool _isRememberMeChecked = false;
23
24   @override
25   Widget build(BuildContext context) {
26     return Scaffold(
27       body: SafeArea(
28         child: SingleChildScrollView(
29           child: Padding(
30             padding: const EdgeInsets.symmetric(horizontal: 32.0),
31             child: Column(
32               crossAxisAlignment: CrossAxisAlignment.stretch,
33               children: <Widget>[
34                 Padding(
35                   padding: const EdgeInsets.symmetric(vertical: 32.0),
36                   child: Image.asset(
37                     'images/flutter.png',
38                     height: 150.0,
```

Key Promoter X

Flutter Outline

Flutter Inspector

Device File Explorer

Logcat

TODO

Terminal

Dart Analysis

Run

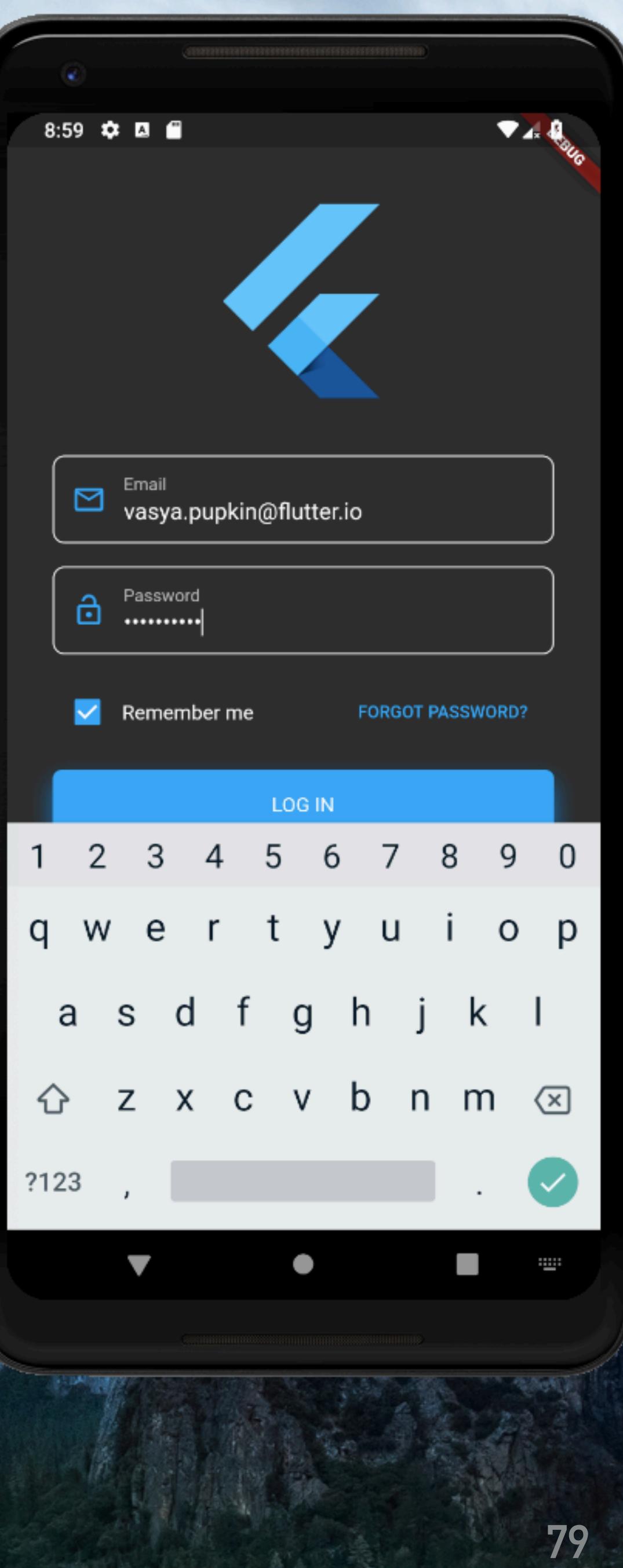
Messages

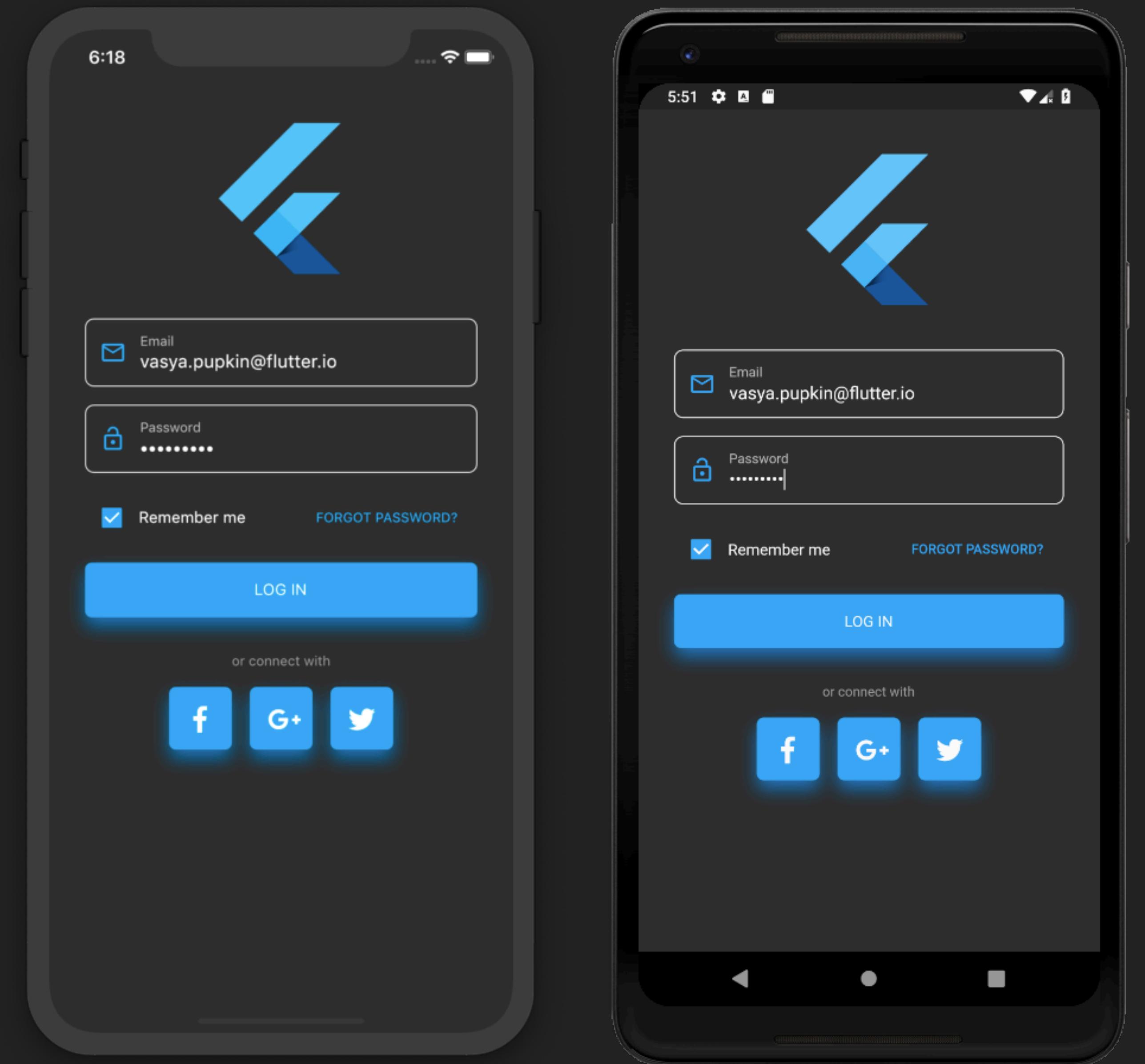
Event Log

Flutter Performance

21 chars 28:37 LF UTF-8 2 spaces

79





ЗАКЛЮЧЕНИЕ

- ▶ Узнали что такое Flutter
- ▶ Познакомились с основными компонентами
- ▶ Научились создавать кастомные виджеты
- ▶ Выяснили как ускорить разработку в IDE

СПАСИБО ЗА ВНИМАНИЕ!

ВОПРОСЫ?

КОНТАКТЫ

АЛЕКСЕЙ ПАНОВ

t.me/ajieksx