

• FLUTTER

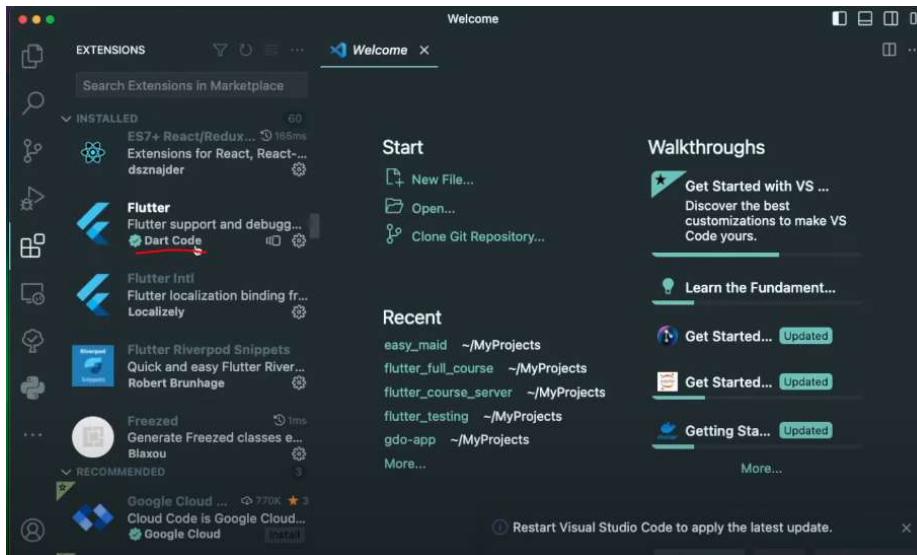
INSTALLATION

Install:

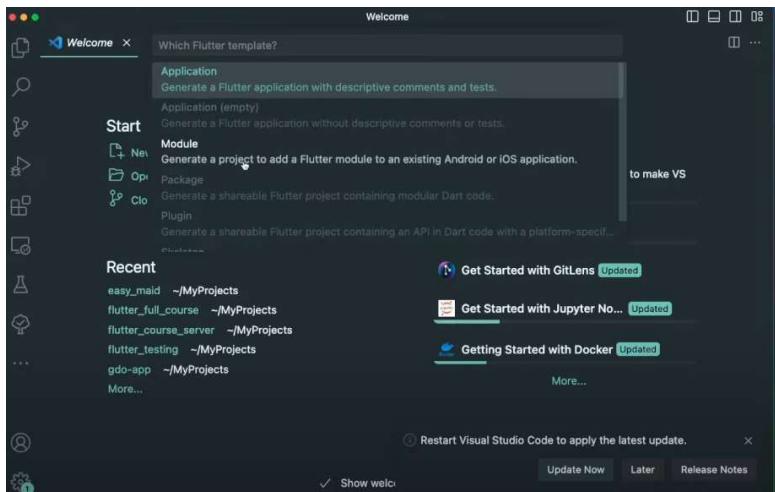
docs.flutter.dev/get-started

Setup Editor:

docs.flutter.dev/get-started/editor

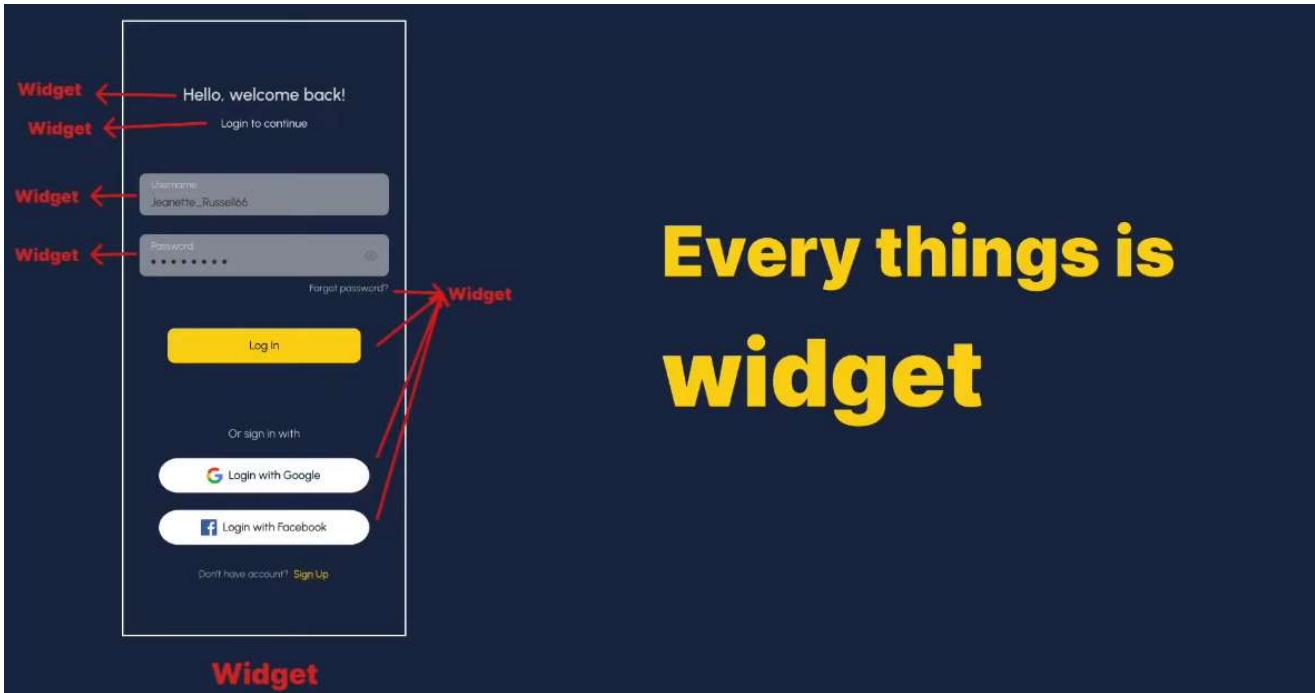


CREATING PROJECT

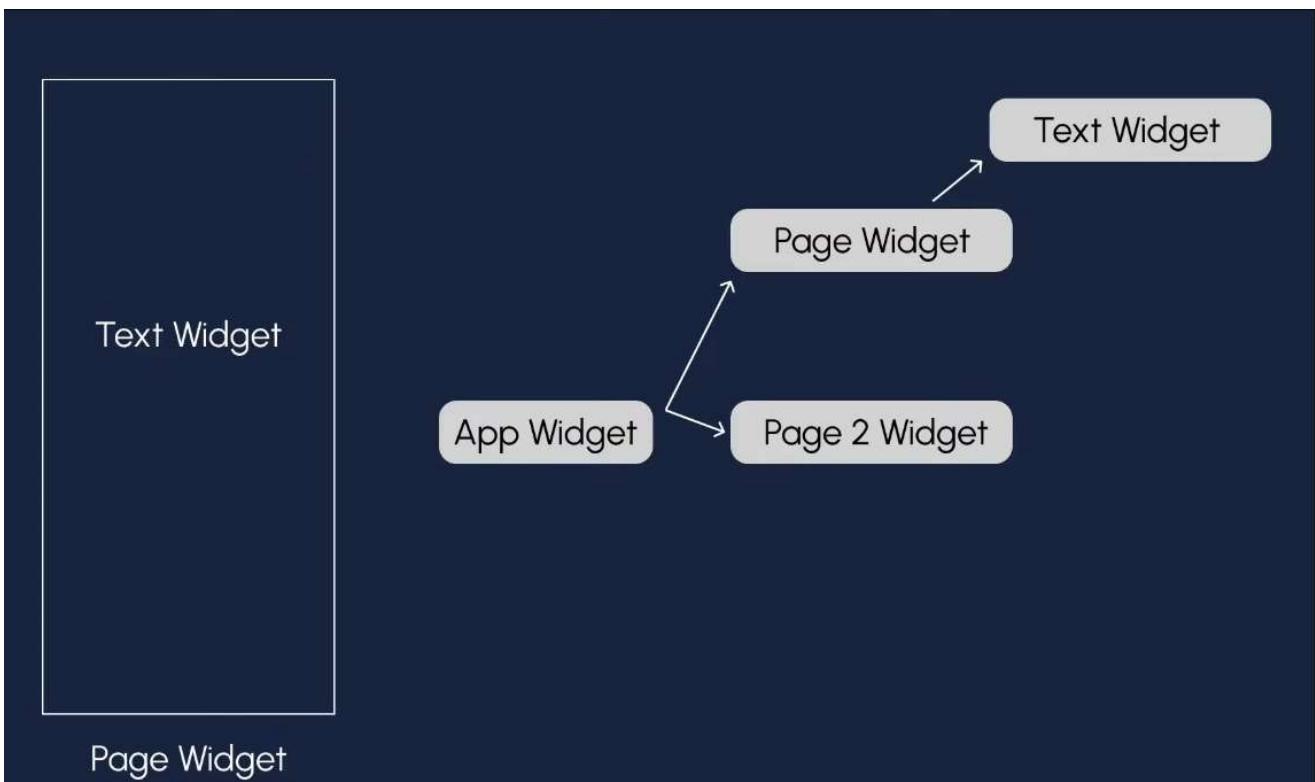


Figma design: <https://www.figma.com/design/l1UCcQApMTiTzs2jzAx7lw/Flutter-course>

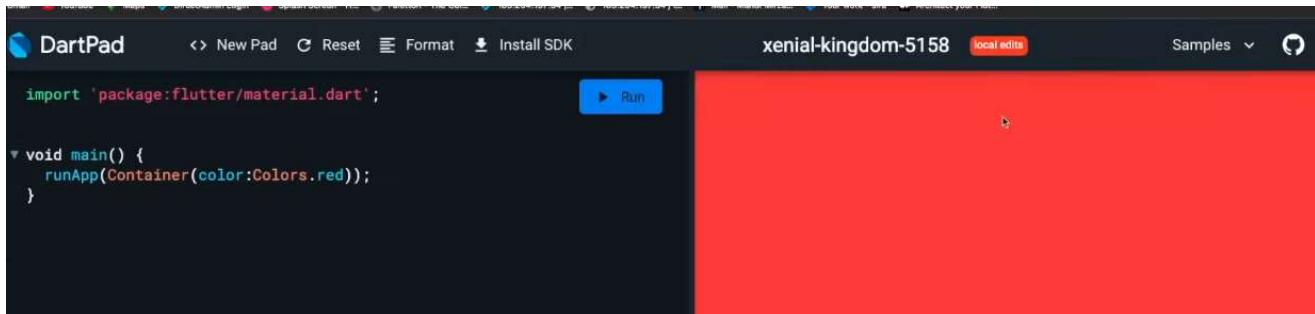
- **WIDGET**



**Every things is
widget**



- color



A screenshot of the DartPad interface. The code editor on the left contains the following Dart code:

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

void main() {
  runApp(Container(color:Colors.red));
}
```

The run button is blue and labeled "Run". To the right is a large red rectangular area representing the rendered UI.

- Text()



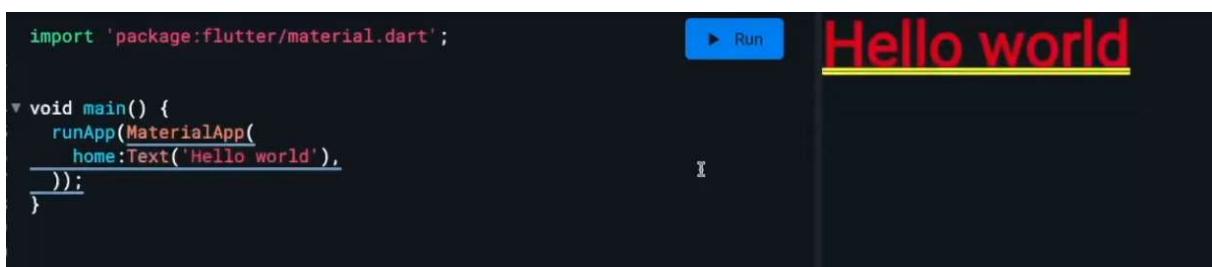
A screenshot of the DartPad interface. The code editor on the left contains the following Dart code:

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

void main() {
  runApp(Text('Hello world'));
}
```

The run button is blue and labeled "Run". Below the code editor is a dark bar with tabs for "Console" and "Documentation". The "Console" tab shows the message "Script error.".

- MaterialApp(home:widget) (style auto came up)



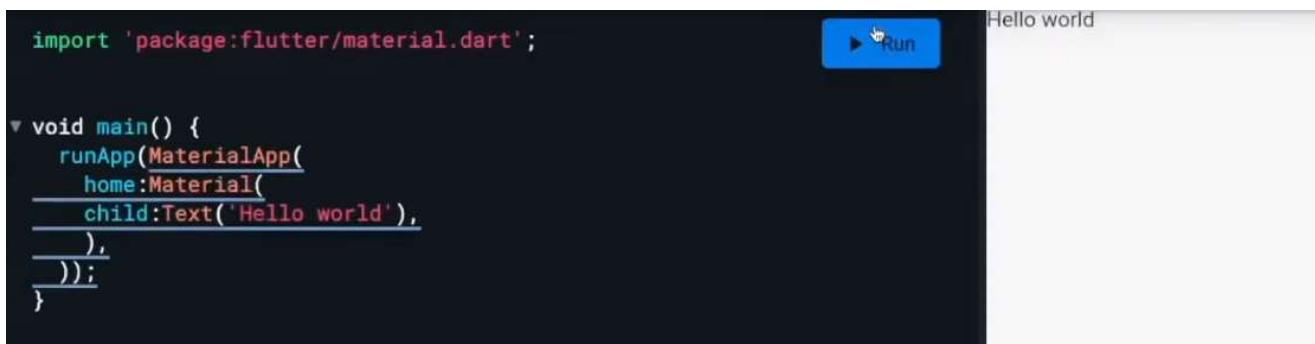
A screenshot of the DartPad interface. The code editor on the left contains the following Dart code:

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

void main() {
  runApp(MaterialApp(
    home:Text('Hello world'),
  ));
}
```

The run button is blue and labeled "Run". To the right is a white area containing the text "Hello world" in black font, which has a yellow underline.

Material Text() without style



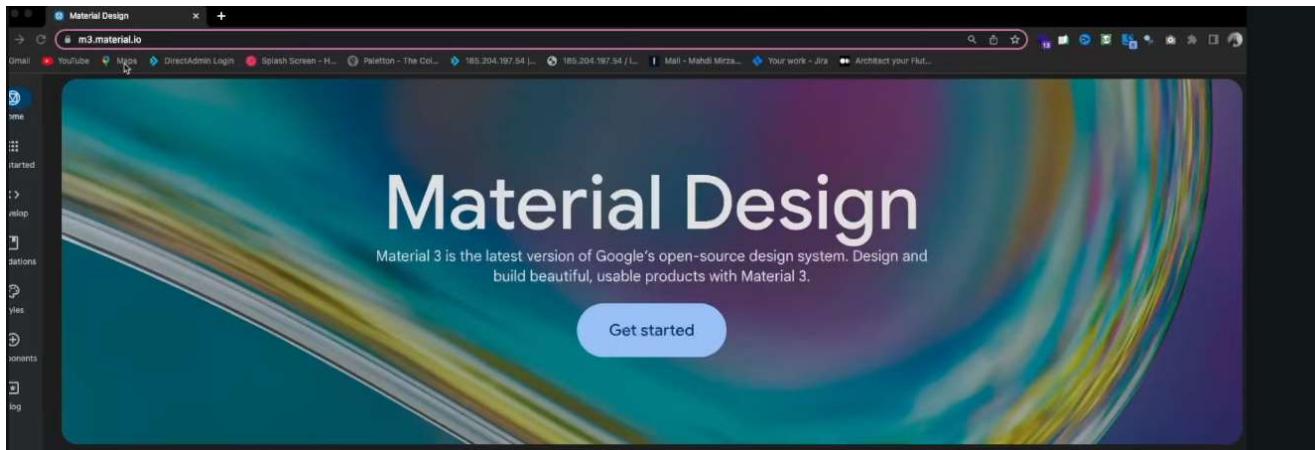
A screenshot of the DartPad interface. The code editor on the left contains the following Dart code:

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

void main() {
  runApp(MaterialApp(
    home:Material(
      child:Text('Hello world'),
    ),
  ));
}
```

The run button is blue and labeled "Run". To the right is a white area containing the text "Hello world" in black font, which has a black underline.

- Material Design:



What's Material?

Material Design is a design system built and supported by Google designers and developers. Material.io includes in-depth UX guidance and UI component implementations for Android, Flutter, and the Web.

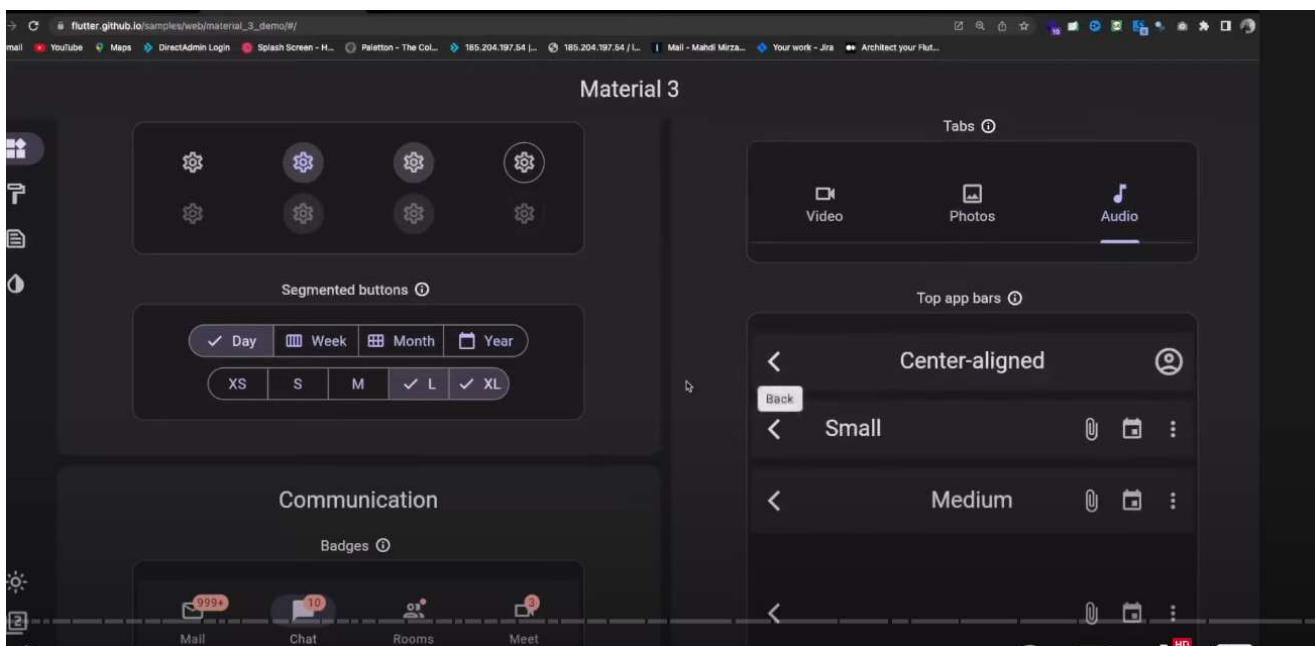
The latest version, Material 3, enables personal, adaptive, and expressive experiences – from dynamic color and enhanced accessibility, to foundations for large screen layouts and design tokens.

UX foundations
Foundations like color, type, and shape are customizable systems in Material

Open-source code
Multi-platform code to build beautiful products, faster

Tutorials, case studies & news
Follow Material's blog for updates, deep dives, and more

Some builtin components like BOOTSTRAP



Material 3

Segments

Segmented buttons

Communication

Badges

Tabs

Top app bars

- Text (in mobile)

The screenshot shows the VS Code interface with the file `main.dart` open. The code defines a `MaterialApp` with a `Material` home screen containing a single `Text` widget with the text "Hello, welcome back!". To the right, an iPhone 14 Pro simulator displays the app's UI, showing the text "Hello, welcome back!" on the screen.

```
main.dart - flutter_full_course
RUN AND ... > No Conf > ... Settings main.dart
lib > main.dart > ...
import 'package:flutter/material.dart';

void main() {
  runApp(MaterialApp(
    home: Material(
      child: Text('Hello, welcome back!'),
    ), // Material
  )); // MaterialApp
}

10 |
```

- Column (Widget below each other)

The screenshot shows the VS Code interface with the file `main.dart` open. The code defines a `MaterialApp` with a `Material` home screen containing a `Column` widget with two children: a `Text` widget with the text "Hello, welcome back!" and another `Text` widget with the text "Login to continue". To the right, an iPhone 8 simulator displays the app's UI, showing the text "Hello, welcome back!" and "Login to continue" stacked vertically.

```
Settings main.dart
lib > main.dart > main
import 'package:flutter/material.dart';

Run | Debug | Profile
void main() {
  runApp(MaterialApp(
    home: Material(
      child: Column(
        children: [
          Text('Hello, welcome back!'),
          Text('Login to continue'),
        ],
      ), // Column
    ), // Material
  )); // MaterialApp
}
```

- Color

The screenshot shows the VS Code interface with the file `main.dart` open. The code defines a `MaterialApp` with a `Material` home screen containing a `Column` with two `Text` widgets. The `Material` widget has its `color` property set to `Colors.blueAccent`, giving the entire app a blue background. To the right, an iPhone 8 simulator displays the app's UI, showing a solid blue background with the text "Hello, welcome back!" and "Login to continue" in white.

```
Settings main.dart
lib > main.dart > MyApp > build
Run | Debug | Profile
void main() {
  runApp(MyApp());
}

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      home: Material(
        color: Colors.blueAccent,
        child: Column(
          children: [
            Text('Hello, welcome back!'),
            Text('Login to continue'),
          ],
        ), // Column
      ), // Material
    ); // MaterialApp
}
```

• StatelessWidget

A screenshot of a Flutter application running on an iPhone 6 simulator. The screen displays a stack of widgets: 'Hello, welcome back!', 'Login to continue', 'New Text', and 'Text is widget'. The code in main.dart defines a MaterialApp with a Column child, which contains these Text widgets.

```
Settings main.dart
lib > main.dart > main
import 'package:flutter/material.dart';

Run | Debug | Profile
void main() {
  runApp(MaterialApp(
    home: Material(
      child: Column(
        children: [
          Text('Hello, welcome back!'),
          Text('Login to continue'),
          Text('New Text'),
          MyApp(),
        ],
      ),
    ),
  ), // Material
); // MaterialApp
}

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Text('Text is widget');
  }
}
```

A screenshot of a Flutter application running on an iPhone 6 simulator. The screen displays a stack of widgets: 'Hello, welcome back!', 'Login to continue', 'New Text', and 'Its a text after state less widget'. The code in main.dart has been modified to use MyApp directly instead of runApp(MyApp()). Red arrows point from the original runApp call to the new MyApp reference, indicating the change.

```
Settings main.dart
lib > main.dart > MyApp > build
import 'package:flutter/material.dart';

Run | Debug | Profile
void main() {
  runApp(MyApp());
}

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      home: Material(
        child: Column(
          children: [
            Text('Hello, welcome back!'),
            Text('Login to continue'),
            Text('New Text'),
            Text('Its a text after state less widget'),
          ],
        ),
      ),
    );
  }
}
```

By doing above structure we can hot reload and see updates on screen in the 1st screen shoot hot reload won't work due to the main function doesn't re-renders

• MainAxisAlignment

```
mainAxisAlignment: MainAxisAlignment.,  
  center  
  end  
  spaceAround  
  spaceBetween  
  spaceEvenly  
] start  
  values
```

The screenshot shows an IDE interface with a code editor and a simulator window.

Code Editor:

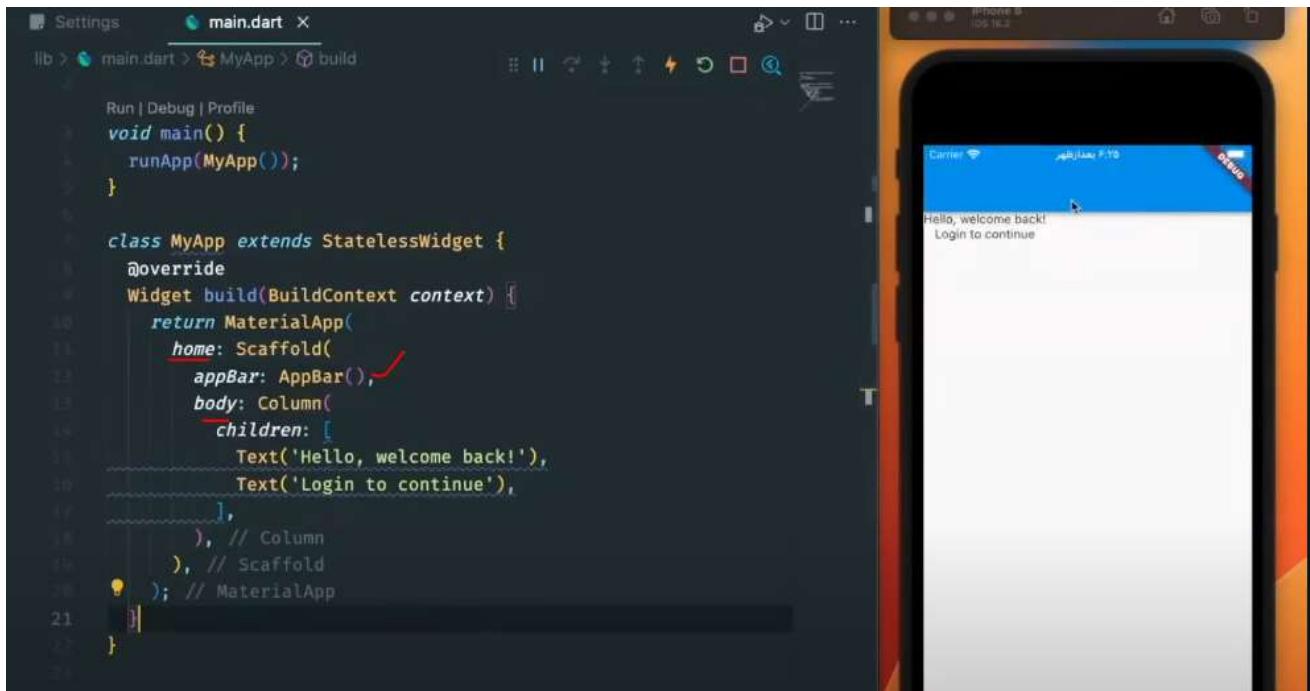
```
Settings main.dart ×  
lib > main.dart > MyApp > build  
Run | Debug | Profile  
void main() {  
  runApp(MyApp());  
}  
  
class MyApp extends StatelessWidget {  
  @override  
  Widget build(BuildContext context) {  
    return MaterialApp(  
      home: Material(  
        child: Column(  
          mainAxisAlignment: MainAxisAlignment.center,  
          children: [  
            Text('Hello, welcome back!'),  
            Text('Login to continue'),  
            Text('New Text'),  
            Text('Its a text after state less widget'),  
          ],  
        ), // Column  
      ), // Material  
    ); // MaterialApp  
}
```

Simulator:

The simulator displays the following text output:

Hello, welcome back!
Login to continue
New Text
Its a text after state less widget

- Scaffold instead of MaterialApp to have more functions like AppBar



The screenshot shows the code for a Flutter application named 'MyApp'. The code uses a Scaffold widget as the home page of a MaterialApp. The Scaffold has an AppBar and a Column body containing two Text widgets: 'Hello, welcome back!' and 'Login to continue'. The application is running on an iPhone 6 simulator.

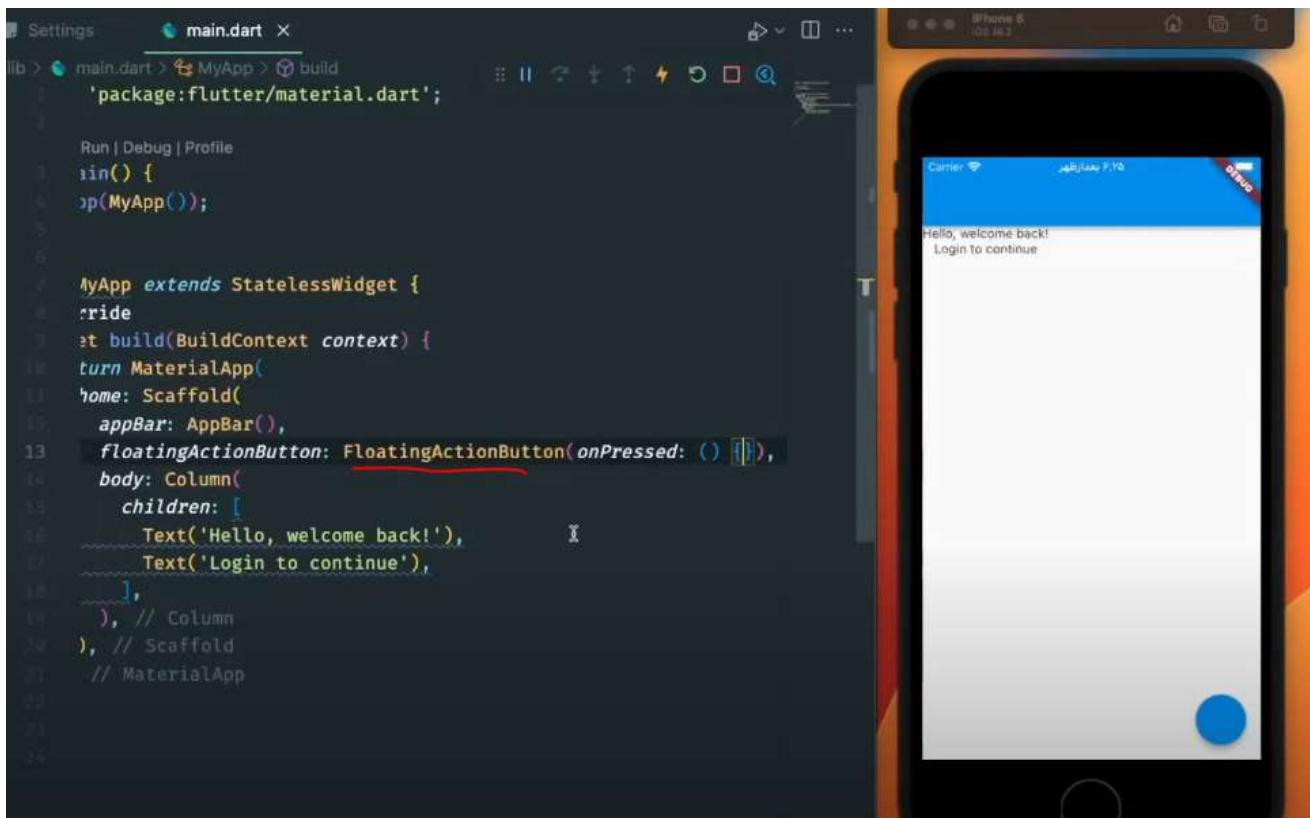
```

Settings main.dart X
lib > main.dart > MyApp > build
Run | Debug | Profile
void main() {
  runApp(MyApp());
}

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      home: Scaffold(
        appBar: AppBar(),
        body: Column(
          children: [
            Text('Hello, welcome back!'),
            Text('Login to continue'),
          ],
        ),
      ),
    );
  }
}

```

- FloatingActionButton in Scaffold



The screenshot shows the same 'MyApp' code as before, but with a FloatingActionButton added to the Scaffold's body. The FloatingActionButton has an onPressed callback. The application is running on an iPhone 6 simulator.

```

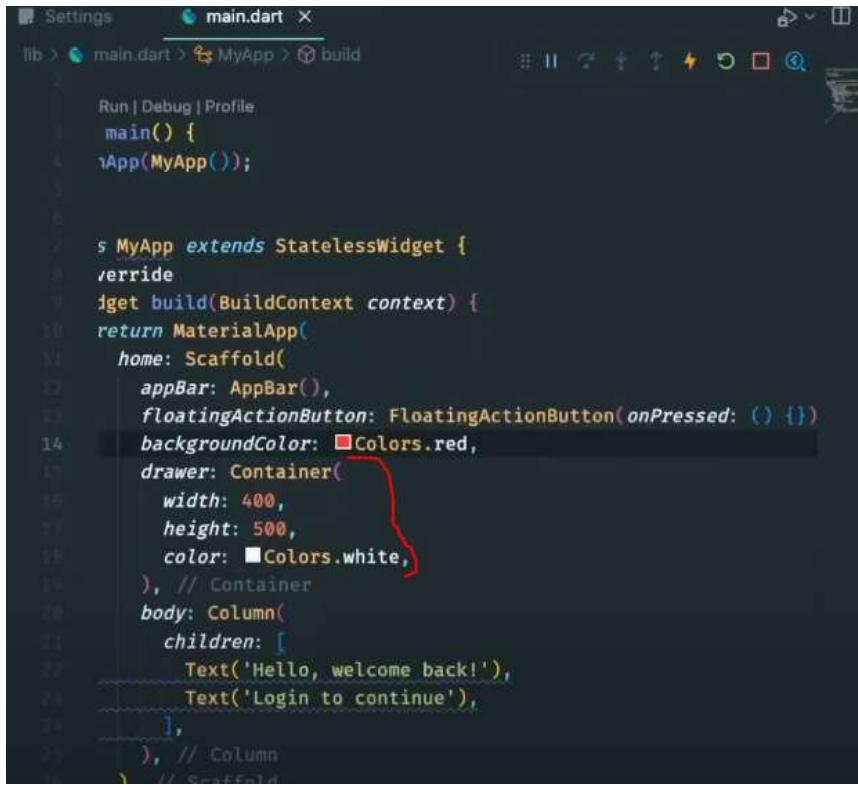
Settings main.dart X
lib > main.dart > MyApp > build
'package:flutter/material.dart';

Run | Debug | Profile
main() {
  runApp(MyApp());
}

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      home: Scaffold(
        appBar: AppBar(),
        floatingActionButton: FloatingActionButton(onPressed: () {}),
        body: Column(
          children: [
            Text('Hello, welcome back!'),
            Text('Login to continue'),
          ],
        ),
      ),
    );
  }
}

```

- drawer

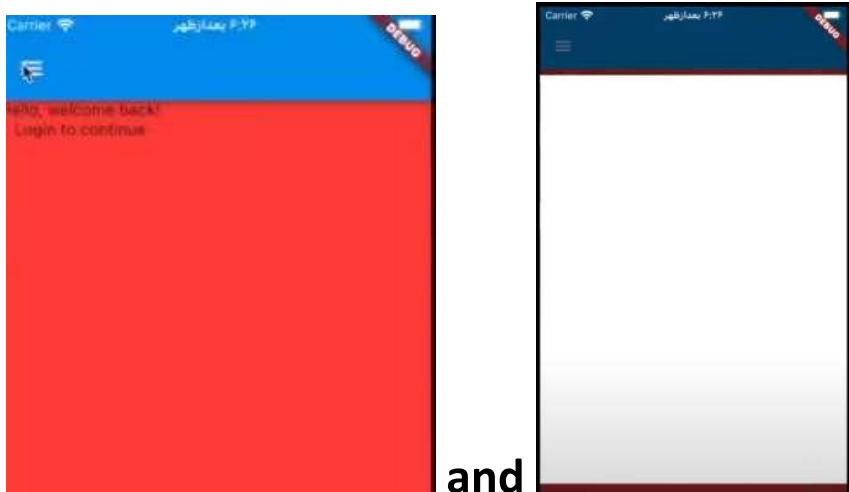


The screenshot shows a code editor with a dark theme. The file is named 'main.dart'. The code defines a 'MyApp' class that extends 'StatelessWidget'. It overrides the 'build' method to return a 'MaterialApp' widget. The 'home' property of 'MaterialApp' is set to a 'Scaffold' widget. The 'Scaffold' has an 'appBar' (not shown), a 'floatingActionButton' (not shown), a 'backgroundColor' set to red, and a 'drawer' property. The 'drawer' is a 'Container' with a width of 400 and a height of 500, colored white. The 'body' of the 'Scaffold' contains a 'Column' with two 'Text' widgets: 'Hello, welcome back!' and 'Login to continue'. A red arrow points from the text 'and' in the question to the 'drawer' code in the editor.

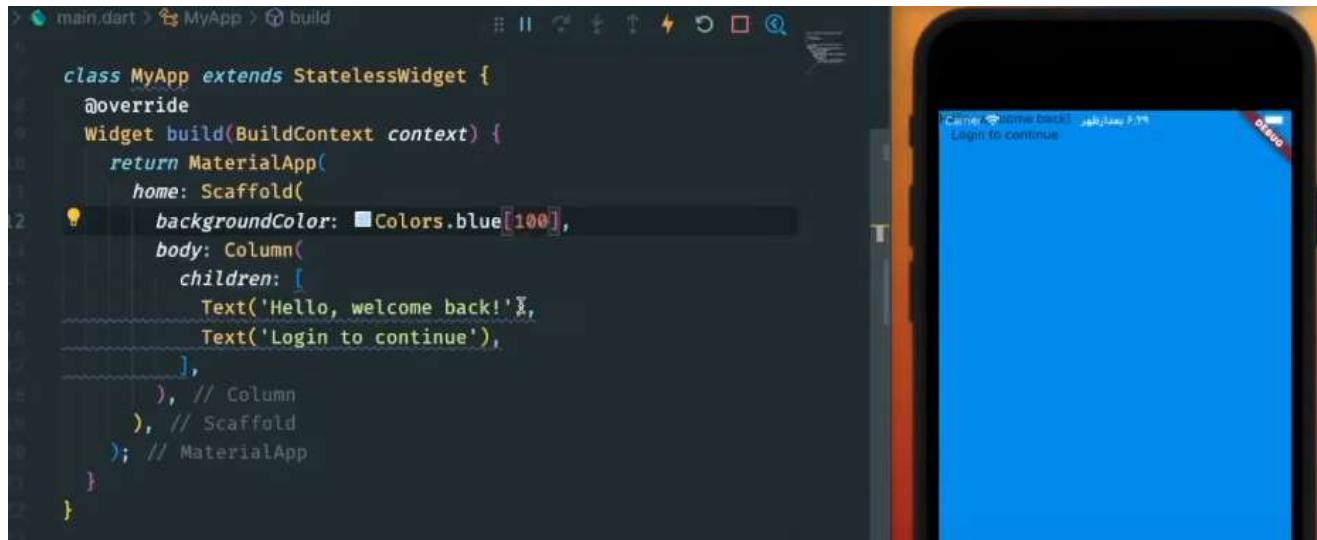
```
Settings main.dart X
lib > main.dart > MyApp > build
Run | Debug | Profile
main() {
  runApp(MyApp());
}

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      home: Scaffold(
        appBar: AppBar(),
        floatingActionButton: FloatingActionButton(onPressed: () {}),
        backgroundColor: Colors.red,
        drawer: Container(
          width: 400,
          height: 500,
          color: Colors.white,
        ), // Container
        body: Column(
          children: [
            Text('Hello, welcome back!'),
            Text('Login to continue'),
          ],
        ), // Column
      ), // Scaffold
    );
  }
}
```

Output

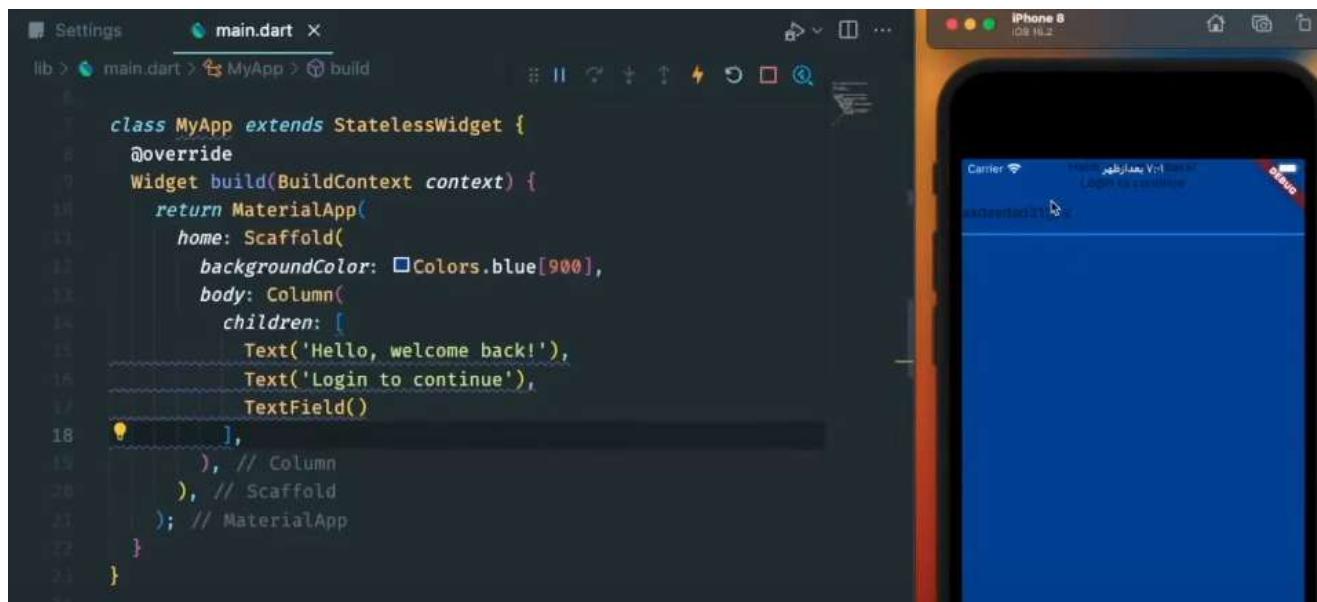


- BackgroundColor: Color.colorName[darkness]



- TextField (for taking input)

- a) Due to the addition of TextField() out previous text became center



b) To make the previous text to left wrap previous



```
class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      home: Scaffold(
        backgroundColor: Colors.blue[900],
        body: ColoredBox(
          color: Colors.red,
          child: Column(
            children: [
              Text('Hello, welcome back!'),
              Text('Login to continue'),
              TextField()
            ],
          ), // Column
        ), // ColoredBox
      ), // Scaffold
    ); // MaterialApp
  }
}
```

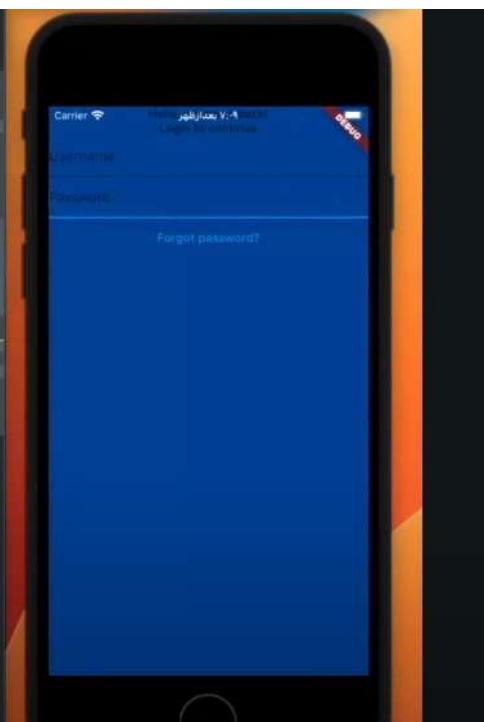
but when we will apply `TextField` red color will take over full width

- TextField(decoration)



```
class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      home: Scaffold(
        backgroundColor: Colors.blue[900],
        body: Column(
          children: [
            Text('Hello, welcome back!'),
            Text('Login to continue'),
            TextField(
              decoration: InputDecoration(hintText: 'Username'),
            ), // TextField
            TextField(),
          ],
        ), // Column
      ), // Scaffold
    ); // MaterialApp
  }
}
```

- **Buttons (2 types)**
- **TextButton (clickable text)**



The screenshot shows an Android emulator displaying a login interface. The background is blue. It features two text input fields labeled "Username" and "Password". Below these is a "Forgot password?" link. At the bottom right is a blue "TextButton" with the text "Forgot password?". The code for this screen is visible in the IDE.

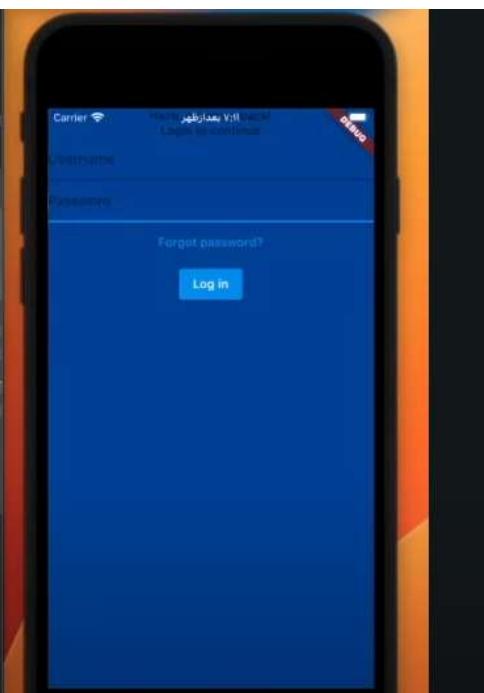
```

lib > main.dart > MyApp > build
void main() {
  runApp(MyApp());
}

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      home: Scaffold(
        backgroundColor: Colors.blue[900],
        body: Column(
          children: [
            Text('Hello, welcome back!'),
            Text('Login to continue'),
            TextField(
              decoration: InputDecoration(hintText: 'Username'),
            ), // TextField
            TextField(
              decoration: InputDecoration(hintText: 'Password'),
            ), // TextField
            TextButton(
              onPressed: () {
                print('Clicked');
              },
              child: Text('Forgot password?')) // TextButton
          ],
        ),
      ),
    );
  }
}

```

- **ElevatedButton**



The screenshot shows an Android emulator displaying a login interface, identical to the one above but with a different button type. The "Forgot password?" link is now part of an "ElevatedButton" at the bottom right. The code for this screen is visible in the IDE.

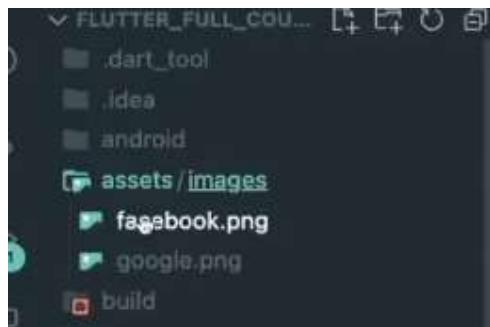
```

lib > main.dart > MyApp > build
void main() {
  runApp(MyApp());
}

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      home: Scaffold(
        backgroundColor: Colors.blue[900],
        body: Column(
          children: [
            Text('Hello, welcome back!'),
            Text('Login to continue'),
            TextField(
              decoration: InputDecoration(hintText: 'Username'),
            ), // TextField
            TextField(
              decoration: InputDecoration(hintText: 'Password'),
            ), // TextField
            TextButton(
              onPressed: () {
                print('Forgot is clicked');
              },
              child: Text('Forgot password?')) // TextButton
            ElevatedButton(
              onPressed: () {
                print('Login is clicked');
              },
              child: Text('Log in')) // ElevatedButton
          ],
        ),
      ),
    );
  }
}

```

- Adding assets folder for images



Update pubspec.yaml file to enable using assets folder

```
Settings main.dart pubspec.yaml
pubspec.yaml

# encourage good coding practices. The lints set provided by the
# activated in the 'analysis_options.yaml' file located at the
# package. See that file for information about deactivating specific
# rules and activating additional ones.
flutter_lints: ^2.0.0

# For information on the generic Dart part of this file, see the
# following page: https://dart.dev/tools/pub/pubspec

# The following section is specific to Flutter packages.
flutter:

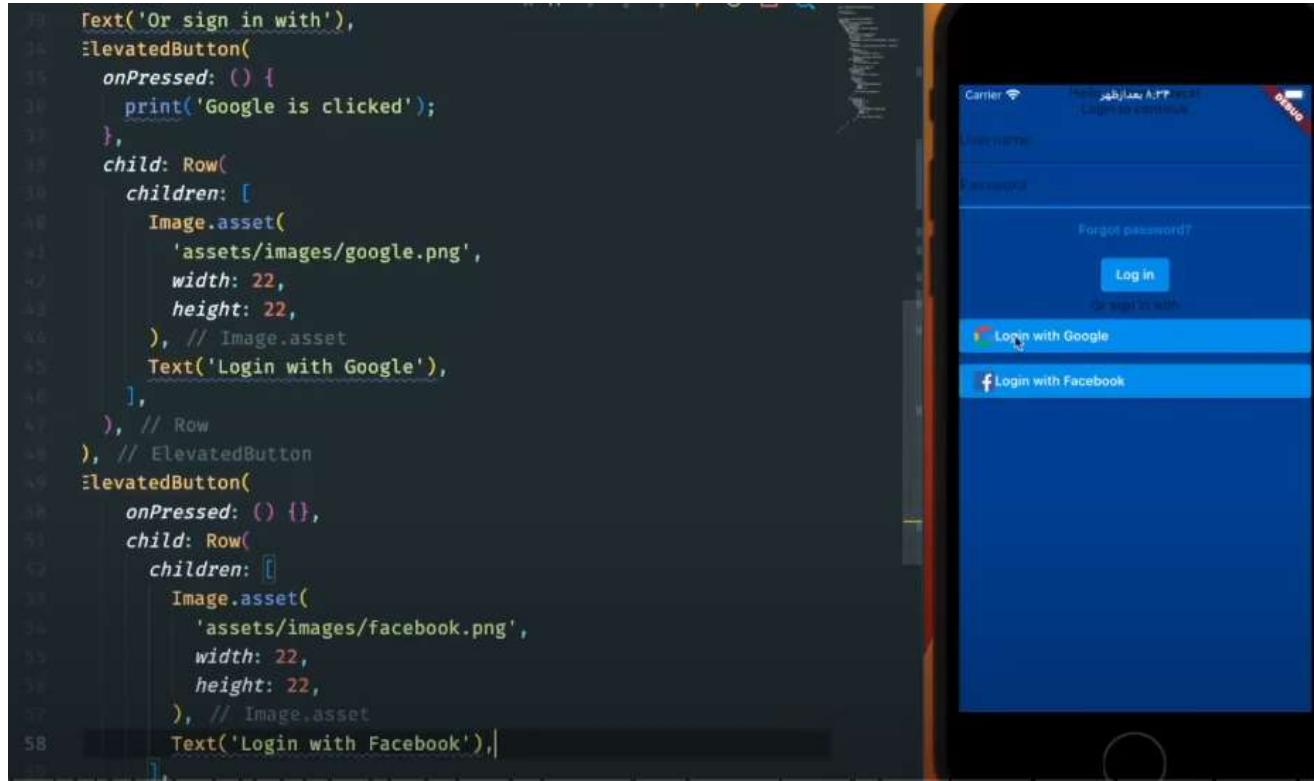
  # The following line ensures that the Material Icons font is
  # included with your application, so that you can use the icons
  # from the Material Icons class.
  uses-material-design: true

  # To add assets to your application, add an assets section, like
  assets:
    - images/
      # - images/a_dot_ham.jpeg

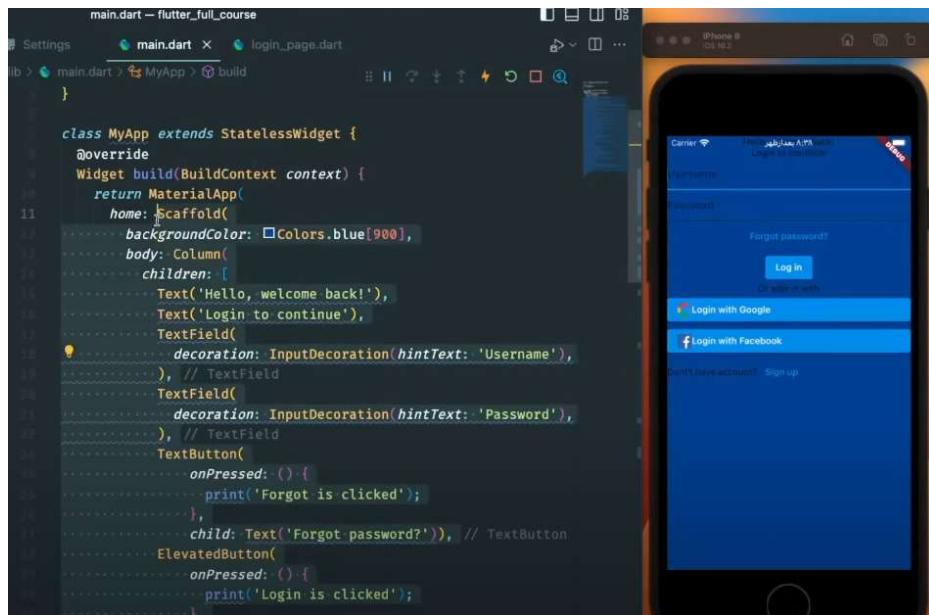
  # An image asset can refer to one or more resolution-specific
```

The screenshot shows a code editor with the 'pubspec.yaml' tab selected. The file contains configuration for Flutter, including linting rules, the flutter plugin, and asset handling. A note at the bottom explains that an image asset can refer to multiple resolution-specific files.

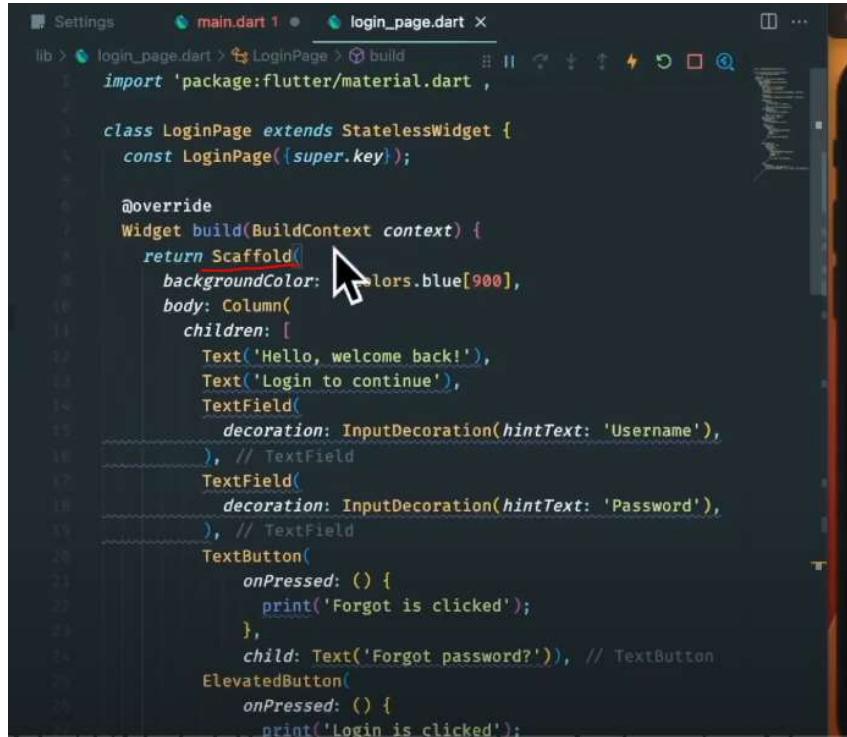
Using Row widget



* Making login file and use class with StatelessWidget



login_page.dart

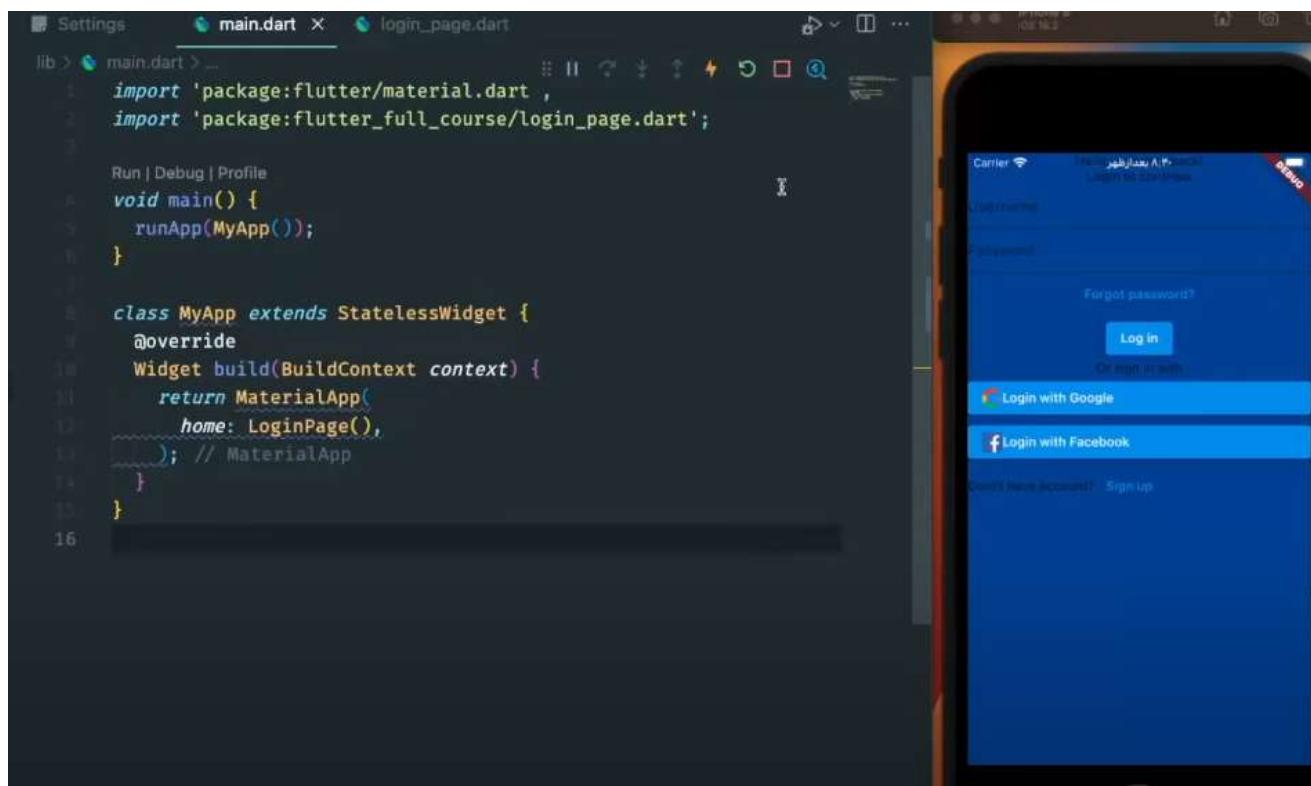


```
Settings main.dart 1 login_page.dart
lib > login_page.dart > LoginPage > build
import 'package:flutter/material.dart';

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

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: Colors.blue[900],
      body: Column(
        children: [
          Text('Hello, welcome back!'),
          Text('Login to continue'),
          TextField(
            decoration: InputDecoration(hintText: 'Username'),
          ), // TextField
          TextField(
            decoration: InputDecoration(hintText: 'Password'),
          ), // TextField
          TextButton(
            onPressed: () {
              print('Forgot is clicked');
            },
            child: Text('Forgot password?'),
          ), // TextButton
          ElevatedButton(
            onPressed: () {
              print('Login is clicked');
            },
            child: Text('Log in'),
          ),
        ],
      ),
    );
  }
}
```

Updated code



The screenshot shows the main.dart file on the left and a running Flutter application on an iPhone X on the right.

main.dart

```
Settings main.dart
lib > main.dart > ...
import 'package:flutter/material.dart';
import 'package:flutter_full_course/login_page.dart';

void main() {
  runApp(MyApp());
}

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      home: LoginPage(),
    );
  }
}
```

Running Application (iPhone X)

The application displays a login screen with the following features:

- Text fields for "User name" and "Password".
- A "Forgot password?" link.
- An "Elevated Button" labeled "Log in".
- Social login options: "Login with Google" and "Login with Facebook".
- Links for "Don't have account? Sign up".

• TextStyle(color:Colors.white)



The screenshot shows an iPhone X simulator displaying a login page. The background is a solid blue color. In the center, there is a white text area containing the message "Hello, welcome back!". The text is white because it is styled with `color: Colors.white;`. Below this text, there are several other UI elements like input fields and buttons, which are standard iOS-style components.

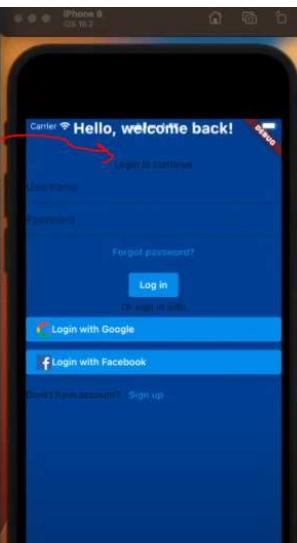
```

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

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: Colors.blue[900],
      body: Column(
        children: [
          Text(
            'Hello, welcome back!',
            style: TextStyle(
              color: Colors.white,
              fontSize: 22,
              fontWeight: FontWeight.bold,
            ),
          ), // Text
        ],
      ),
    );
  }
}

```

• SizeBox(width:int,height:int)



The screenshot shows an iPhone X simulator displaying a login page. A red box highlights a `SizedBox` widget located between two text inputs. This `SizedBox` has a height of 32 and a width of 50. It is used here to create a visual separator or gap between the email and password fields. The rest of the page includes standard login UI elements like a "Log in" button and social media login options.

```

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

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: Colors.blue[900],
      body: Column(
        children: [
          Text(
            'Hello, welcome back!',
            style: TextStyle(
              color: Colors.white,
              fontSize: 22,
              fontWeight: FontWeight.bold,
            ),
          ), // Text
          SizedBox(
            height: 32,
            width: 50,
          ), // SizedBox
          Text('Login to continue'),
        ],
      ),
    );
  }
}

```

Adding child in SizedBox()



The screenshot shows an iPhone X simulator displaying a login page. A red box highlights a `SizedBox` widget that contains a child `Text` element. The `SizedBox` has a height of 32 and a width of 30. The child `Text` contains the message "This is text for test sizebox". This demonstrates how you can add content directly into a `SizedBox` to create a specific layout element.

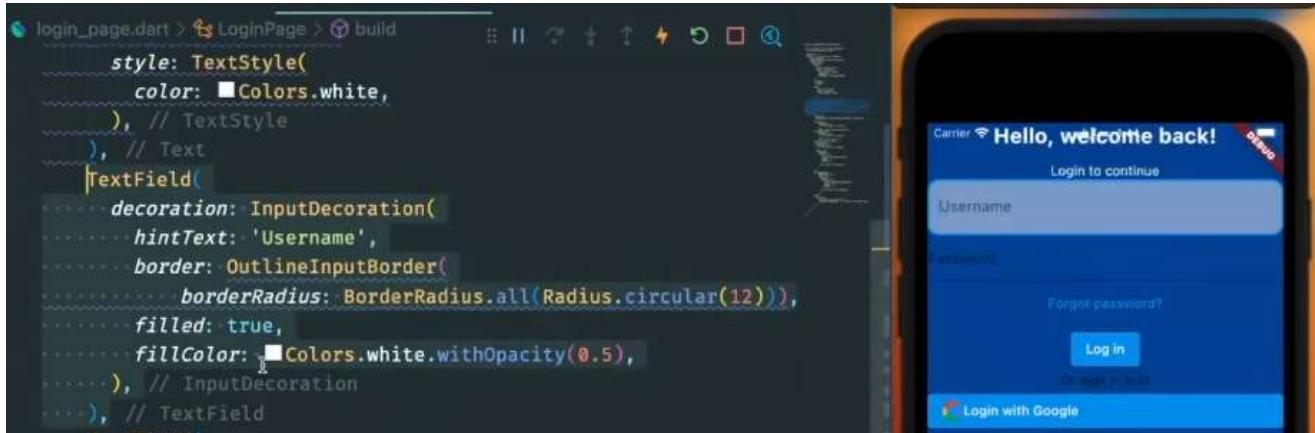
```

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

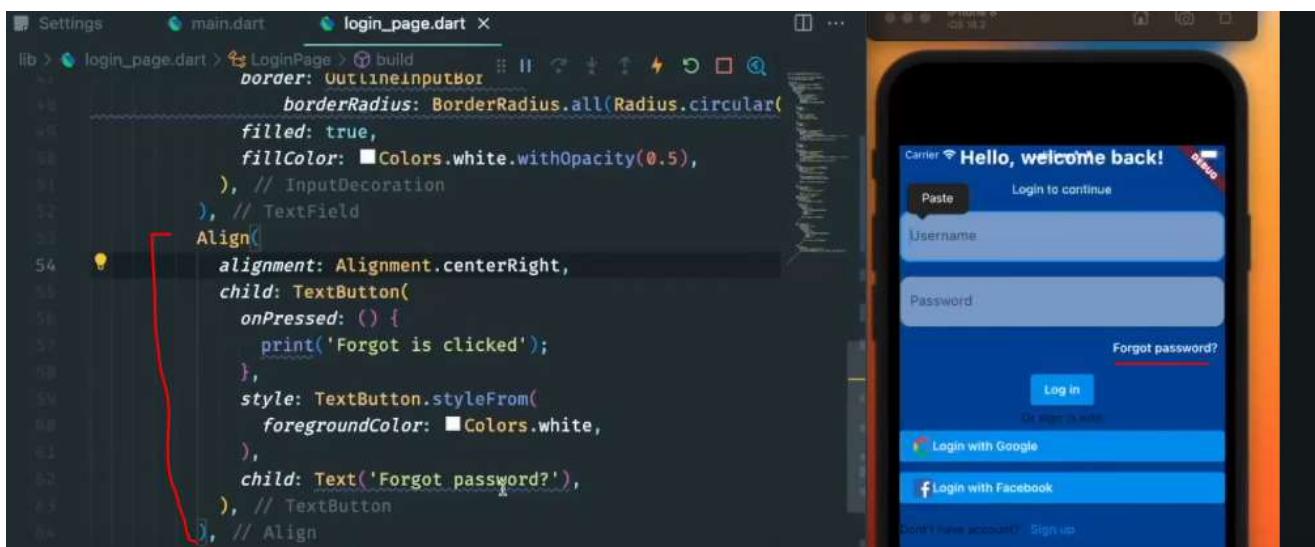
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: Colors.blue[900],
      body: Column(
        children: [
          Text(
            'Hello, welcome back!',
            style: TextStyle(
              color: Colors.white,
              fontSize: 22,
              fontWeight: FontWeight.bold,
            ),
          ), // Text
          SizedBox(
            height: 32,
            width: 30,
            child: Text('This is text for test sizebox'),
          ), // SizedBox
          Text('Login to continue'),
        ],
      ),
    );
  }
}

```

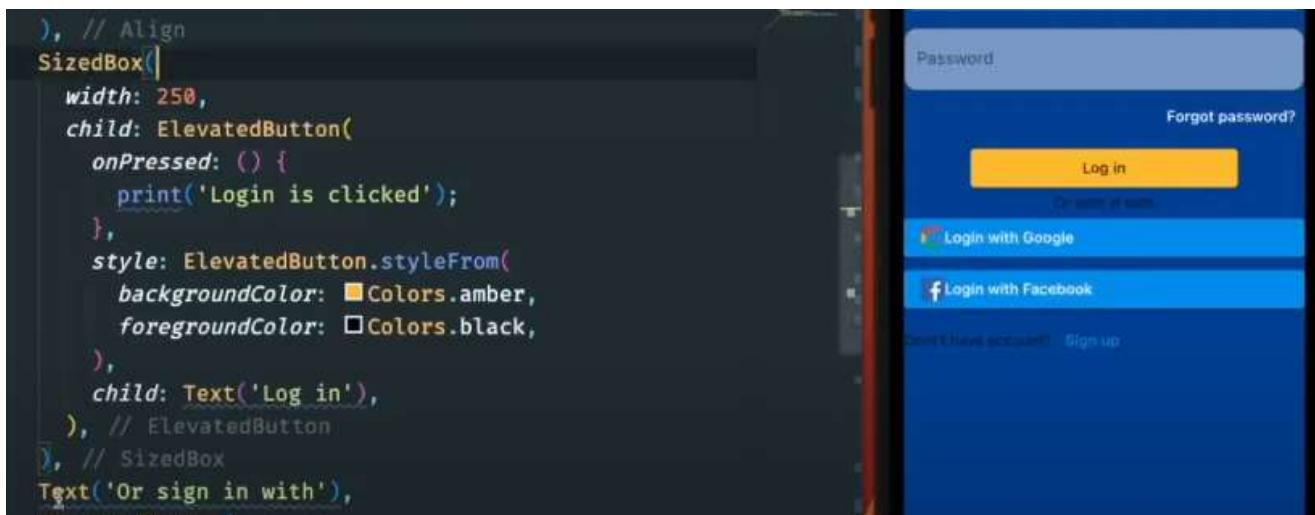
TextField Decoration



- Align() by wrapping widget in it



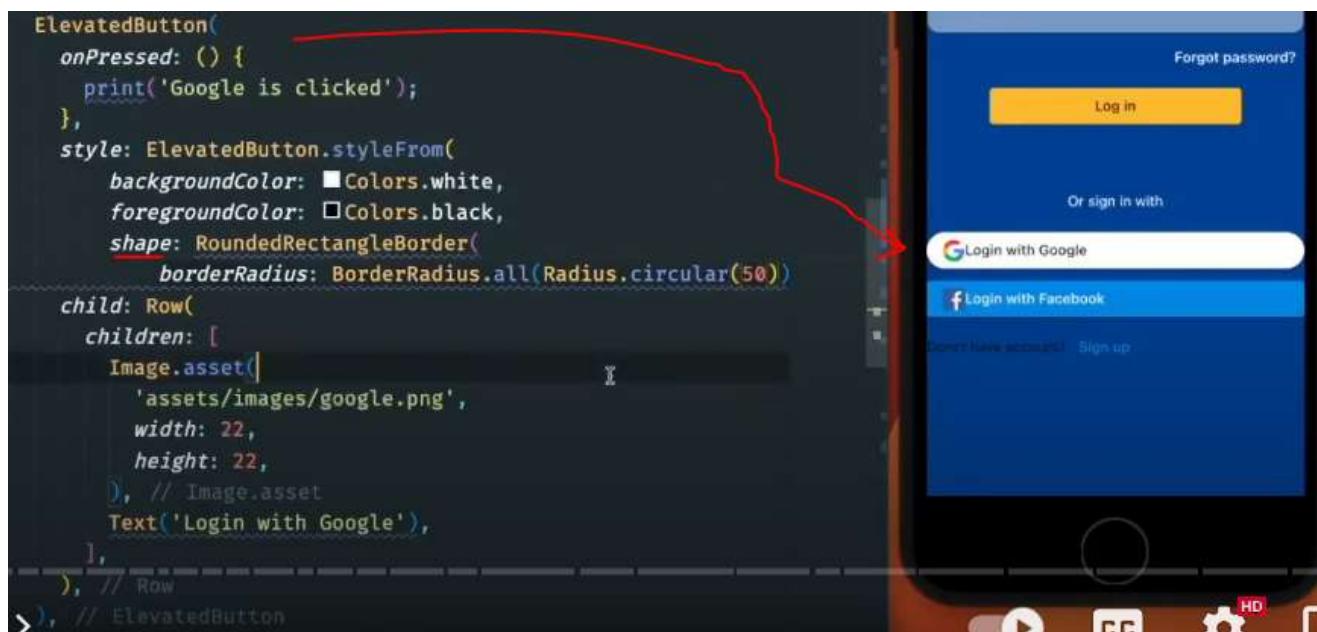
- **SizeBox** wrapping to give size to button



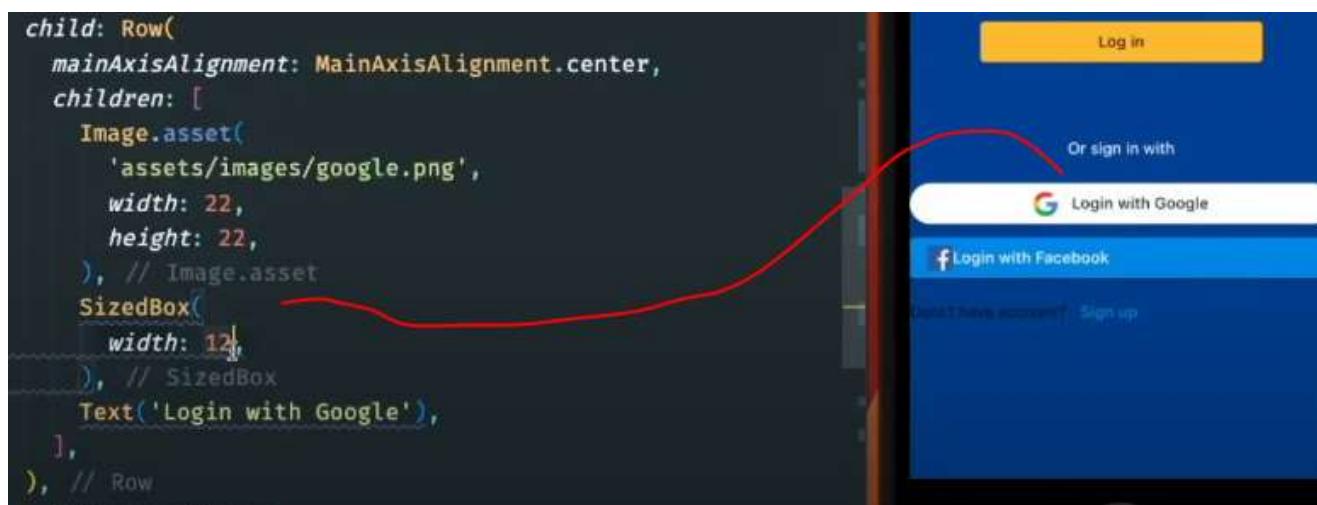
Spacing between text and styling Text



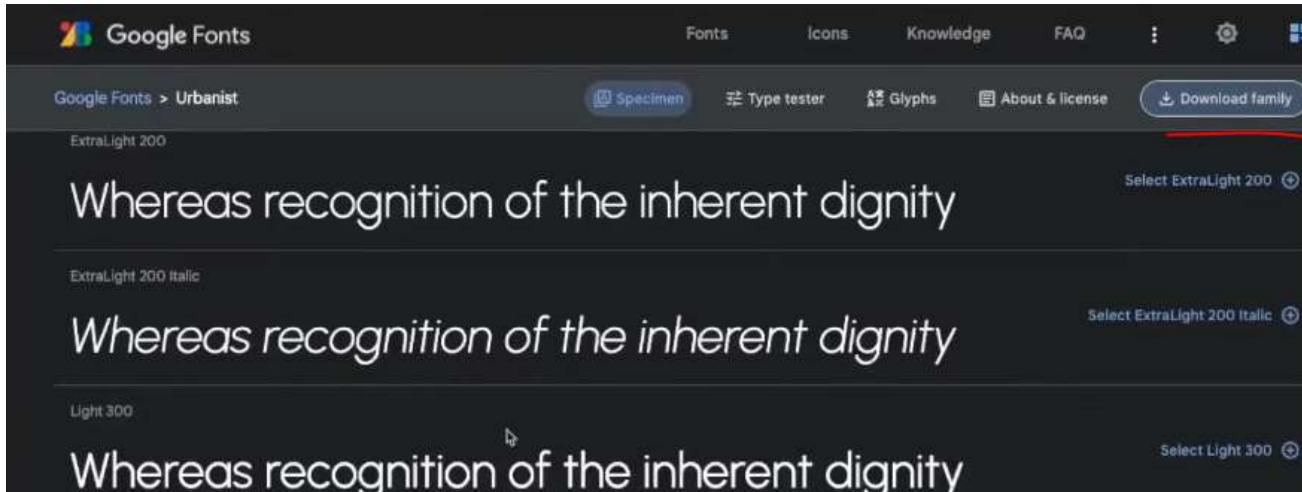
- shape to give BorderRadius in ElevatedButton



- MainAxisAlignment in Row



• ADDING GOOGLE FONTS



Download fonts unzip and add in

assets/fonts

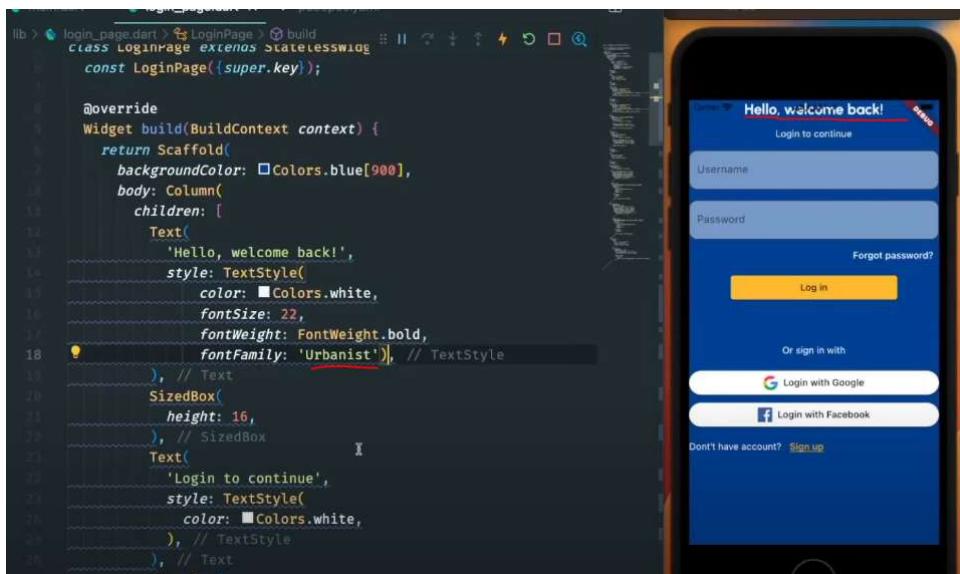
Update pubspec.yaml file along with the weight

The screenshot shows a code editor with two tabs: 'FLUTTER_FULL_COURSE' and 'pubspec.yaml'. The 'pubspec.yaml' tab contains the following code:

```
# list giving the asset and other descriptors for the font.
# example:
fonts:
  - family: Urbanist
    fonts:
      - asset: assets/fonts/Urbanist-Light.ttf
        weight: 300
      - asset: assets/fonts/Urbanist-Regular.ttf
      - asset: fonts/Urbanist-Medium.ttf
        weight: 500
```

The 'FLUTTER_FULL_COURSE' tab shows the project structure with an 'assets' folder containing various font files: Urbanist-Black.ttf, Urbanist-Bold.ttf, Urbanist-ExtraBold.ttf, Urbanist-Light.ttf, Urbanist-Medium.ttf, Urbanist-Regular.ttf, and Urbanist-SemiBold.ttf.

a) Using Fonts locally

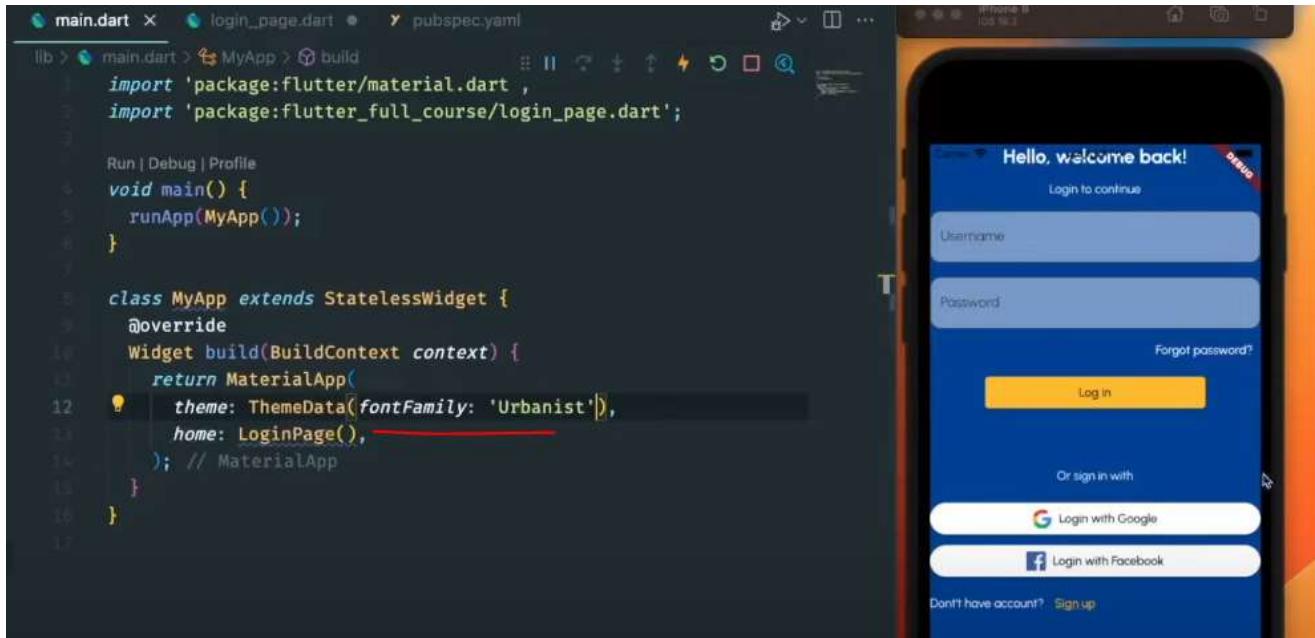


b) Using Fonts Globally

- Theme

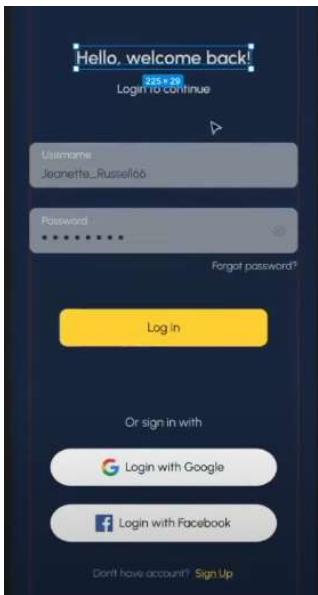
In main.dart under

`MaterialApp(theme: ThemeData(fontFamily:"NameOfFont"))`

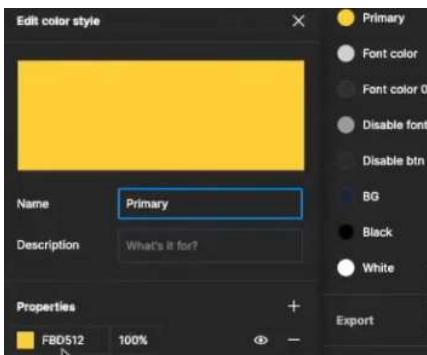


Now we don't need to define where we use it

- Optimizing out Login UI as per below figma



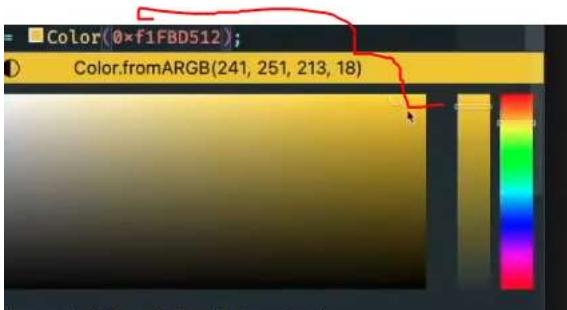
Defining our figma file colors inside a file with class named “AppColors”



Create file: Lib/styles/app_colors.dart

>Before writing Hex code prepend “0xff” to tell FullOpacity

Eg: f1 means 18% opacity



```
main.dart app_colors.dart login_page.dart pubspec.yaml
lib > styles > app_colors.dart > AppColors > disableFont
import 'package:flutter/material.dart',
```

```
class AppColors {
    static const primary = Color(0xFFFBBD51);
    static const font = Color(0xFFD8D8D8);
    static const font2 = Color(0xFF373737);
    static const disableFont = Color(0xFFA7A7A7);
    static const disableButton = Color(0xFF303030);
    static const background = Color(0xFF1A2947);
    static const black = Color(0xFF000000);
    static const white = Color(0xFFFFFFFF);}
```

Update the background color globally inside theme in main.dart

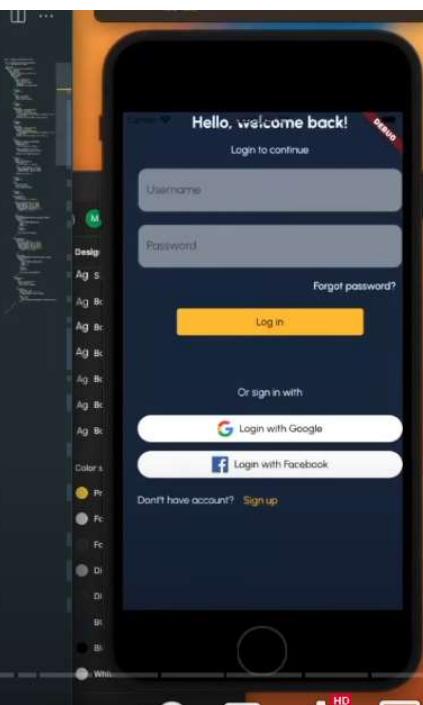
```
main.dart login_page.dart pubspec.yaml
lib > main.dart > MyApp > build
Run | Debug | Profile
void main() {
    runApp(MyApp());
}

class MyApp extends StatelessWidget {
    @override
    Widget build(BuildContext context) {
        return MaterialApp(
            theme: ThemeData(
                fontFamily: 'Urbanist',
                scaffoldBackgroundColor: AppColors.background,
            ),
            home: LoginPage(),
        );
    }
}
```

The screenshot shows a mobile application's login screen. The background of the entire screen is a dark blue color, which corresponds to the `scaffoldBackgroundColor` set in the `MaterialApp` theme. The text and other UI elements are displayed in white or light colors to provide contrast.

- Padding: EdgeInsets.type(position:0)

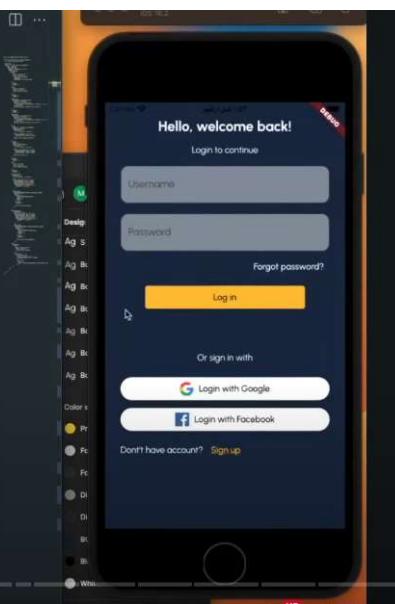
Adding padding to left



The screenshot shows the code for a login page in a Flutter application. The code uses a `Padding` widget with `EdgeInsets.only(left: 20)` to add padding to the left side of the body. The resulting UI on the right shows a dark blue background with white text and input fields. A yellow "Log in" button is centered horizontally, and there is additional padding on the left side of the screen.

```
class LoginPage extends StatelessWidget { const LoginPage({super.key}); @override Widget build(BuildContext context) { return Scaffold( body: Padding( padding: EdgeInsets.only(left: 20), child: Column( children: [ Text( 'Hello, welcome back!', style: TextStyle( color: Colors.white, fontSize: 22, fontWeight: FontWeight.bold, ), ), // Text SizedBox( height: 16, ), // SizedBox Text( 'Login to continue', style: TextStyle( color: Colors.white, ), ), // Text ], ), ), ); } }
```

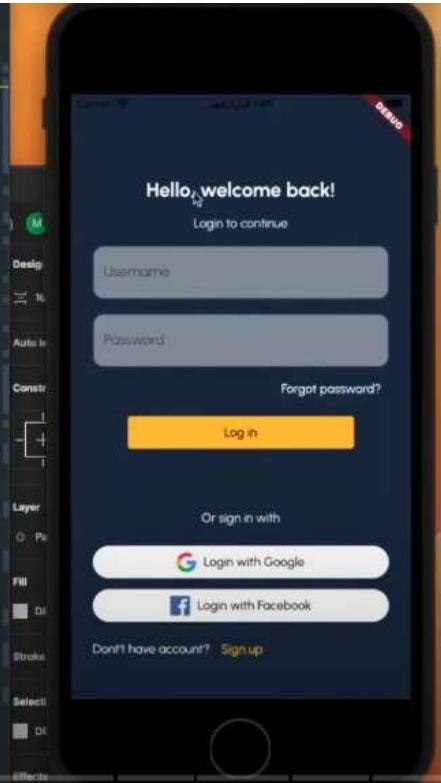
Adding padding on both side



The screenshot shows the same login page code as above, but with `padding: EdgeInsets.all(24)` applied to the `Padding` widget. This adds equal padding to all four edges of the body. The resulting UI on the right shows a dark blue background with white text and input fields, and a yellow "Log in" button centered horizontally, with padding on both the left and right sides of the screen.

```
class LoginPage extends StatelessWidget { const LoginPage({super.key}); @override Widget build(BuildContext context) { return Scaffold( body: Padding( padding: EdgeInsets.all(24), child: Column( children: [ Text( 'Hello, welcome back!', style: TextStyle( color: Colors.white, fontSize: 22, fontWeight: FontWeight.bold, ), ), // Text SizedBox( height: 16, ), // SizedBox Text( 'Login to continue', style: TextStyle( color: Colors.white, ), ), // Text ], ), ), ); } }
```

Adding padding to top

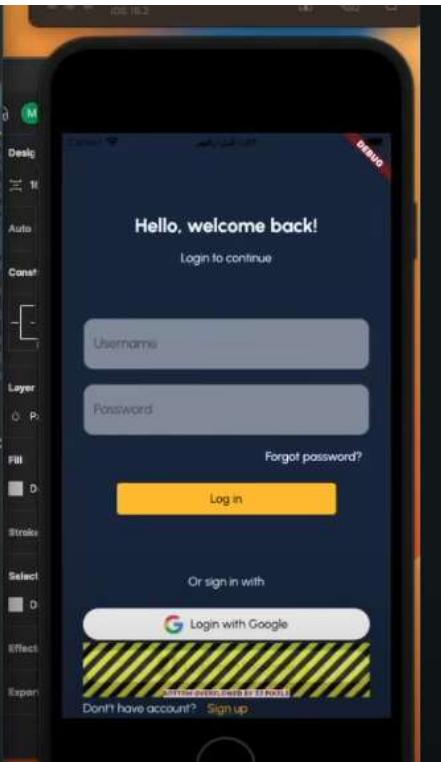


Code snippet from `login_page.dart`:

```
const LoginPage({super.key});  
  
@override  
Widget build(BuildContext context) {  
  return Scaffold(  
    body: Padding(  
      padding: EdgeInsets.all(24),  
      child: Column(  
        children: [  
          SizedBox(  
            height: 68,  
          ), // SizedBox  
          Text(  
            'Hello, welcome back!',  
            style: TextStyle(  
              color: Colors.white,  
              fontSize: 22,  
              fontWeight: FontWeight.bold,  
            ), // TextStyle  
          ), // Text  
          SizedBox(  
            height: 16,  
          ), // SizedBox  
          Text(  
            'Login to continue',  
            style: TextStyle(  
              color: Colors.white,  
            ),  
          ), // Text  
        ],  
      ),  
    ),  
  );  
}
```

The screenshot shows the Flutter IDE with the code above. To the right is a preview of the mobile application. The screen has a dark blue background. At the top, the text "Hello, welcome back!" is displayed in white. Below it is a "Login to continue" button. The main content area contains two input fields: "Username" and "Password", followed by a "Forgot password?" link and a large yellow "Log in" button. Below these are social login options for Google and Facebook, and links for "Don't have account? Sign up". The UI is centered and looks good.

Pixel out of the screen



Code snippet from `login_page.dart`:

```
const LoginPage({super.key});  
  
@override  
Widget build(BuildContext context) {  
  return Scaffold(  
    body: Padding(  
      padding: EdgeInsets.all(24),  
      child: Column(  
        children: [  
          SizedBox(  
            height: 68,  
          ), // SizedBox  
          Text(  
            'Hello, welcome back!',  
            style: TextStyle(  
              color: Colors.white,  
              fontSize: 22,  
              fontWeight: FontWeight.bold,  
            ), // TextStyle  
          ), // Text  
          SizedBox(  
            height: 16,  
          ), // SizedBox  
          Text(  
            'Login to continue',  
            style: TextStyle(  
              color: Colors.white,  
            ),  
          ), // Text  
        ],  
      ),  
    ),  
  );  
}
```

PROBLEMS 41 OUTPUT TERMINAL GITLENS DEBUG CONSOLE Filter (e.g. text, tex... ▾ ^
Reloaded 1 of 642 libraries in 219ms (compile: 44 ms, reload: 83 ms, reassemble: 71 ms).
Reloaded 1 of 642 libraries in 246ms (compile: 52 ms, reload: 82 ms, reassemble: 78 ms).

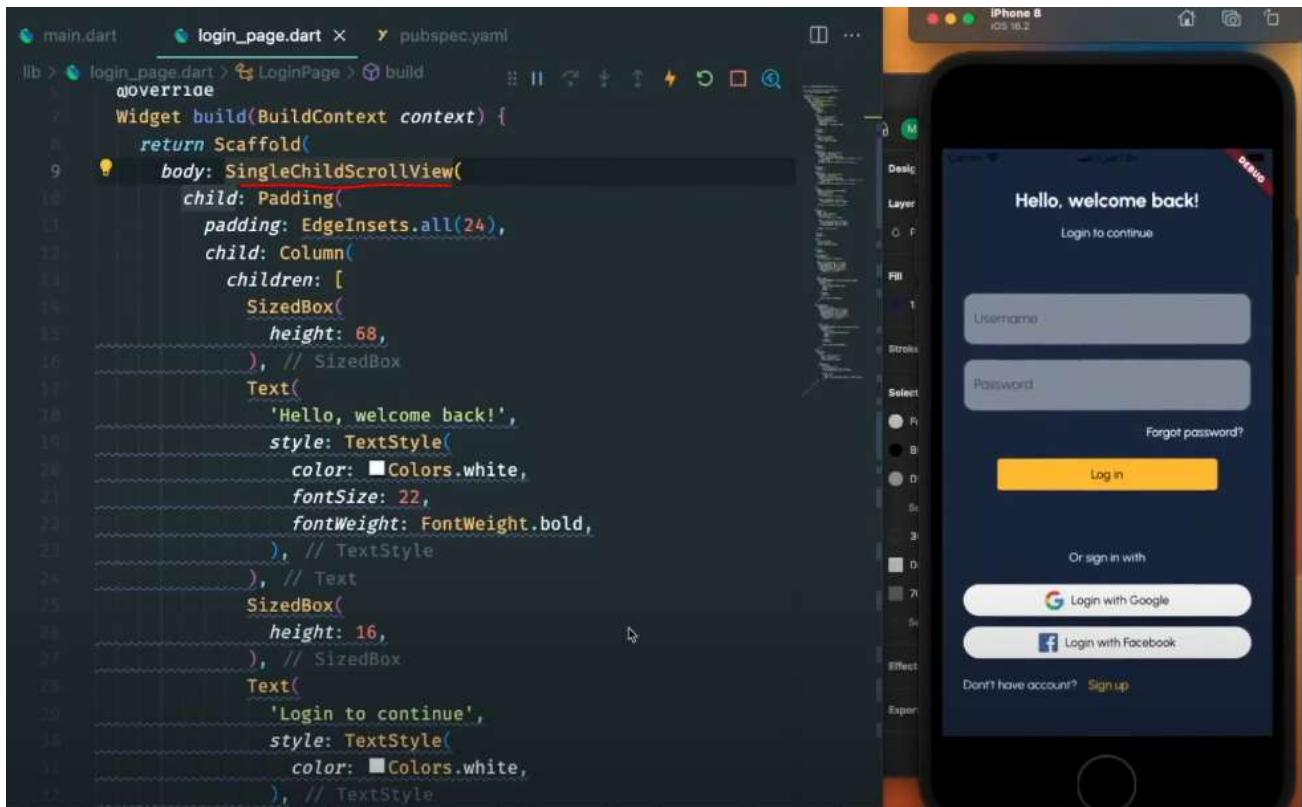
Exception caught by rendering library
The following assertion was thrown during layout:
A RenderFlex overflowed by 37 pixels on the bottom.

The relevant error-causing widget was
Column
To inspect this widget in Flutter DevTools, visit: http://127.0.0.1:9101/#/inspector?url=
=http%3A%2F%2F127.0.0.1%3A59608%2FlwzyZCb7q_E%3D%2F&inspectorRef=inspector-0
The overflowing RenderFlex has an orientation of Axis.vertical.
The edge of the RenderFlex that is overflowing has been marked in the rendering with a yellow
and black striped pattern. This is usually caused by the contents being too big for
the space available.

The screenshot shows the same code as the previous one, but now with a red border around the bottom of the screen. A yellow and black striped pattern is visible at the bottom, indicating a rendering error. The IDE's problem pane shows an assertion error: "A RenderFlex overflowed by 37 pixels on the bottom." The rest of the UI looks identical to the first screenshot.

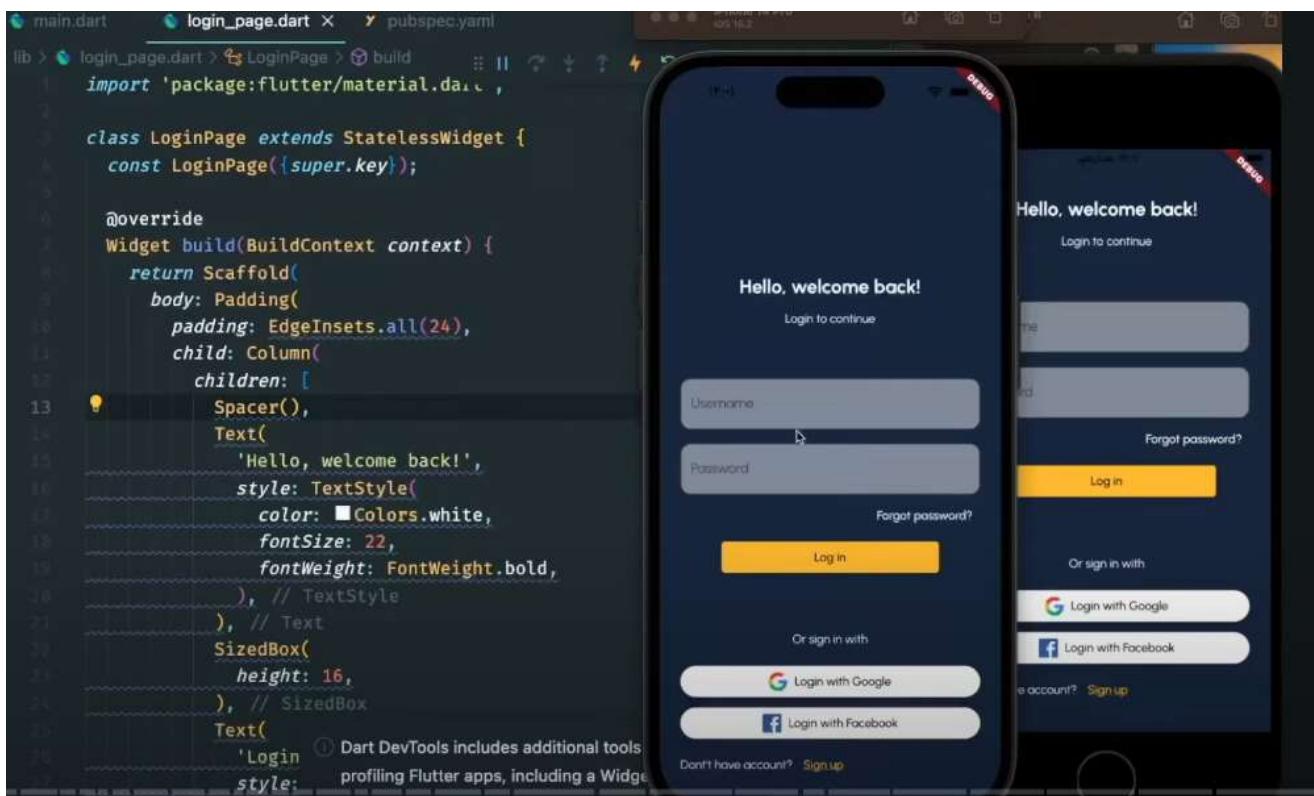
- **SingleChildScrollView (to add scroll)**

- a) To fix it wrap the Padding insider

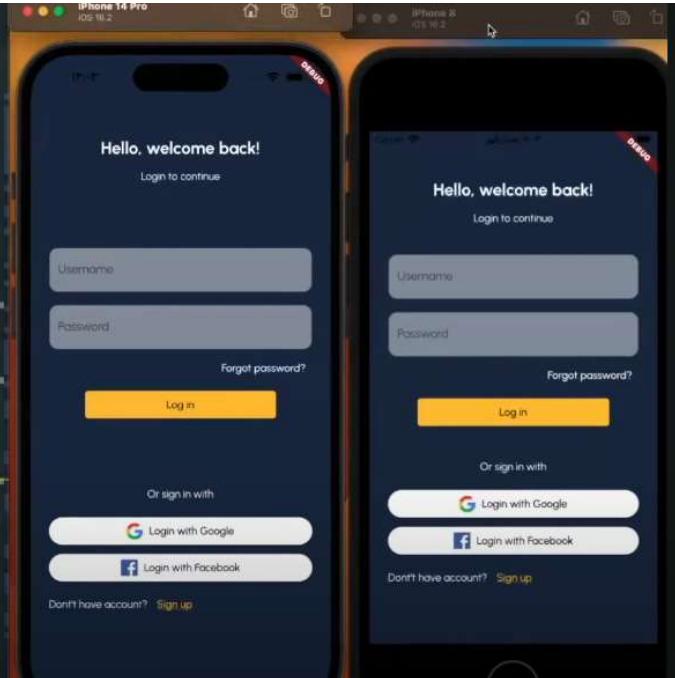


- b) To make responsive design

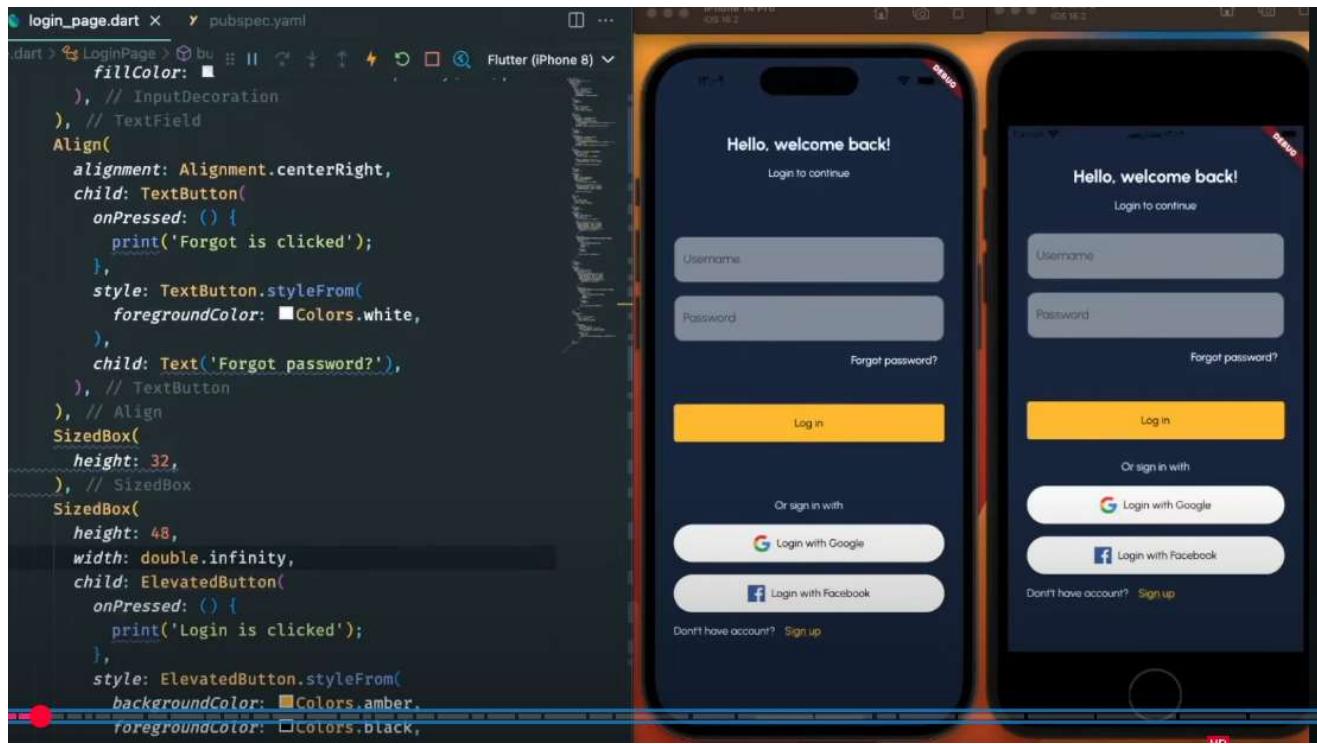
- **Spacer() [takes all available space]**



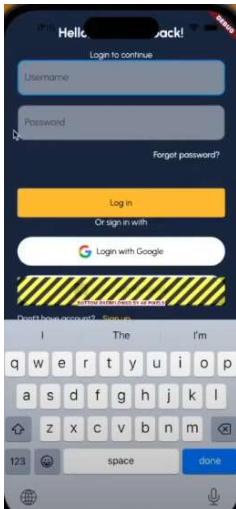
Applying multiple spacer() instead of fixed value SizeBox()



- Width=double.infinity [to cover the entire width except padding]



- Keyboard enable layout break



To fix this use: `resizeToAvoidBottomInset: false`

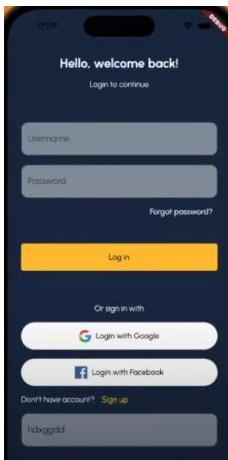
```

main.dart          login_page.dart x  pubspec.yaml
lib > login_page.dart > LoginPage > build ...  Flutter (iPhone 8) ...
class LoginPage extends StatelessWidget {
  const LoginPage({super.key});

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      resizeToAvoidBottomInset: false, ✓
      body: Padding(
        padding: EdgeInsets.all(24),
        child: Column(
          children: [
            Spacer(),
            Text(
              'Hello, welcome back!',
              style: TextStyle(
                color: Colors.white,
                fontSize: 22,
                fontWeight: FontWeight.bold,
              ),
            ), // Text
            SizedBox(
              height: 16,
            ), // SizedBox
            Text(
              'Login to continue',
              style: TextStyle(

```

- Input at the bottom of screen



With the “resizeToAvoidBottomInset: false” we can see the input which is in the bottom of screen

- SingleChildScrollView
 - MediaQuery.of(context).size.height [gives full screen height]

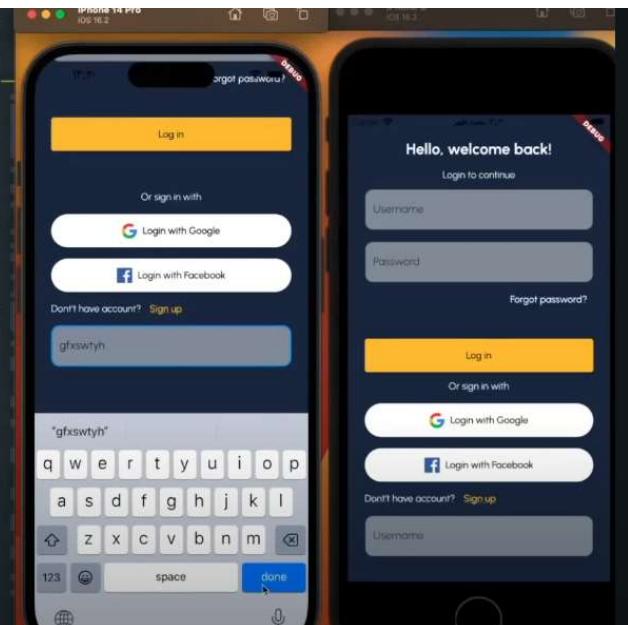


youtu.be/A3WrA4zAaPw

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

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

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: SingleChildScrollView(
        child: SizedBox(
          height: MediaQuery.of(context).size.height,
          child: Padding(
            padding: EdgeInsets.all(24),
            child: Column(
              children: [
                Spacer(),
                Text(
                  'Hello, welcome back!',
                  style: TextStyle(
                    color: Colors.white,
                    fontSize: 22,
                    fontWeight: FontWeight.bold,
                  ),
                ), // TextStyle
                Text(
                  'I am a Flutter developer',
                  style: TextStyle(
                    color: Colors.white,
                    fontSize: 16,
                  ),
                ),
              ],
            ),
          ),
        ),
      ),
    );
  }
}
```



- Center(child:widget)

>> MAKING HOME PAGE

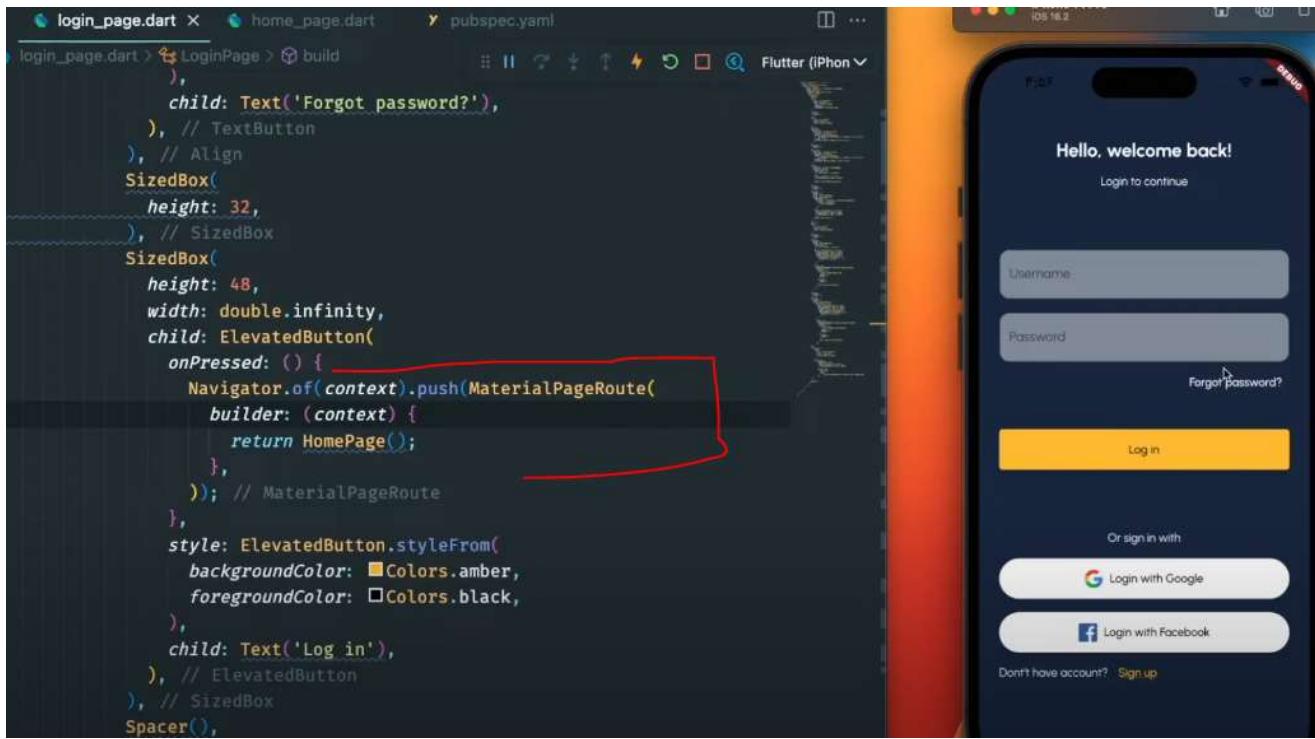


```
lib > pages > home_page.dart
class HomePage extends StatelessWidget {
    const HomePage({super.key});

    @override
    Widget build(BuildContext context) {
        return Scaffold(
            body: Center(child: Text("Hello this is home p
        ); // Scaffold
    }
}
```

- Navigation to different screens

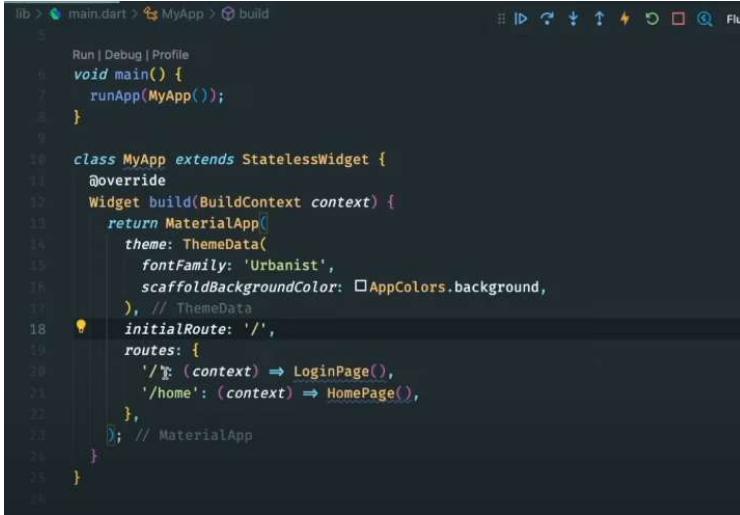
a) using Navigator and MaterialPageRoute



```
login_page.dart
    ),
    child: Text('Forgot password?'),
),
),
SizedBox(
height: 32,
),
SizedBox(
height: 48,
width: double.infinity,
child: ElevatedButton(
onPressed: () {
Navigator.of(context).push(MaterialPageRoute(
builder: (context) {
return HomePage();
}),
);
},
style: ElevatedButton.styleFrom(
backgroundColor: Colors.amber,
foregroundColor: Colors.black,
),
child: Text('Log in'),
),
),
Spacer(),

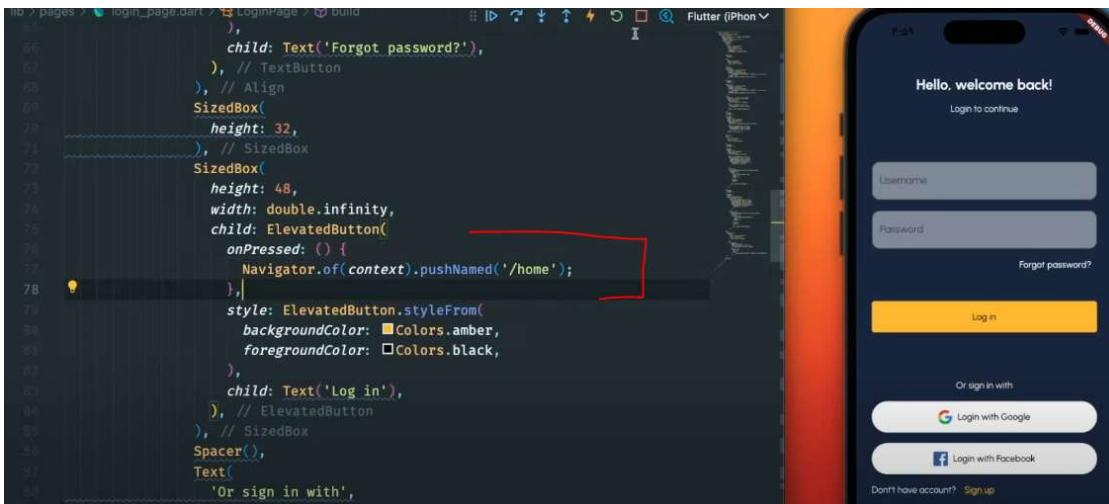
```

b) Using Route name



```
lib > main.dart > MyApp > build
  5
  Run | Debug | Profile
  6 void main() {
  7   runApp(MyApp());
  8 }
  9
 10 class MyApp extends StatelessWidget {
 11   @override
 12   Widget build(BuildContext context) {
 13     return MaterialApp(
 14       theme: ThemeData(
 15         fontFamily: 'Urbanist',
 16         scaffoldBackgroundColor: AppColors.background,
 17       ), // ThemeData
 18       initialRoute: '/',
 19       routes: {
 20         '/': (context) => LoginPage(),
 21         '/home': (context) => HomePage(),
 22       },
 23     ); // MaterialApp
 24   }
 25 }
```

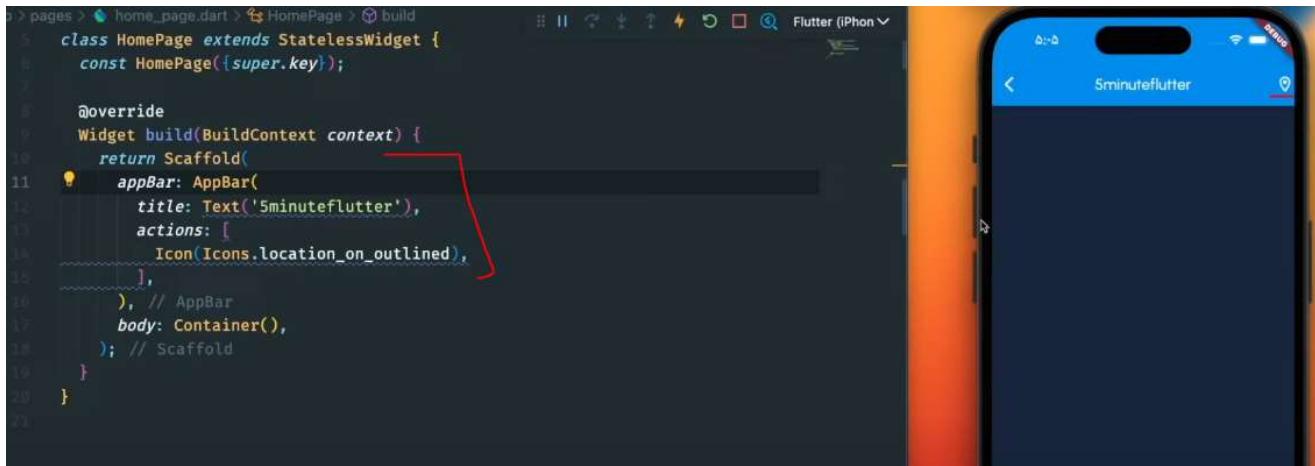
Using routes to navigate



Output

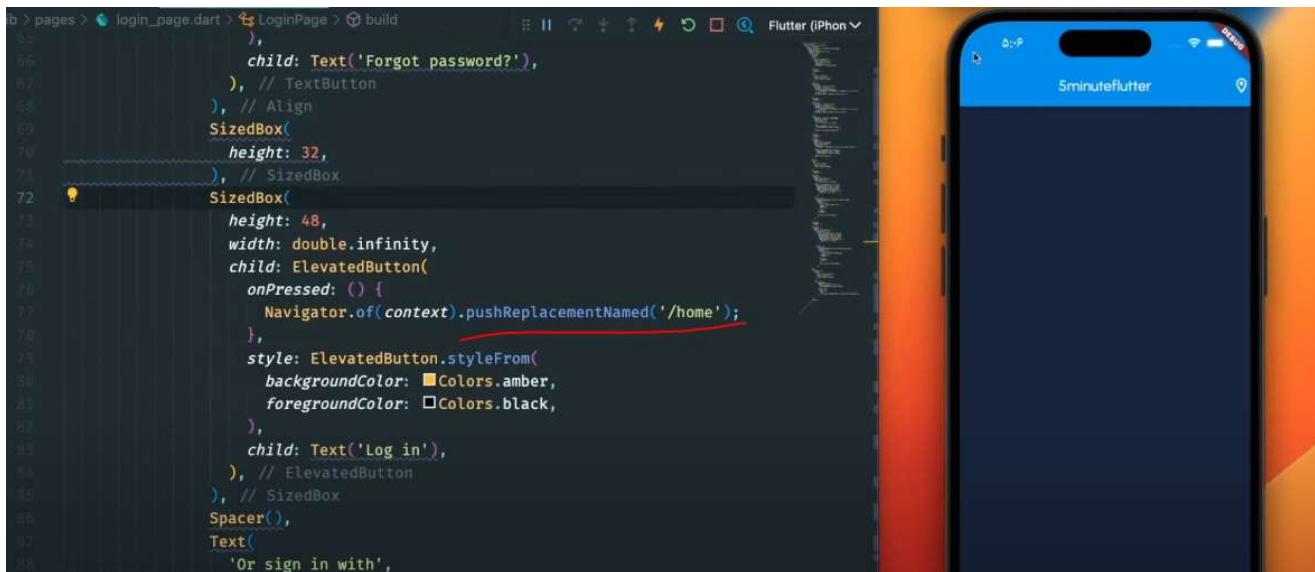


- AppBar()

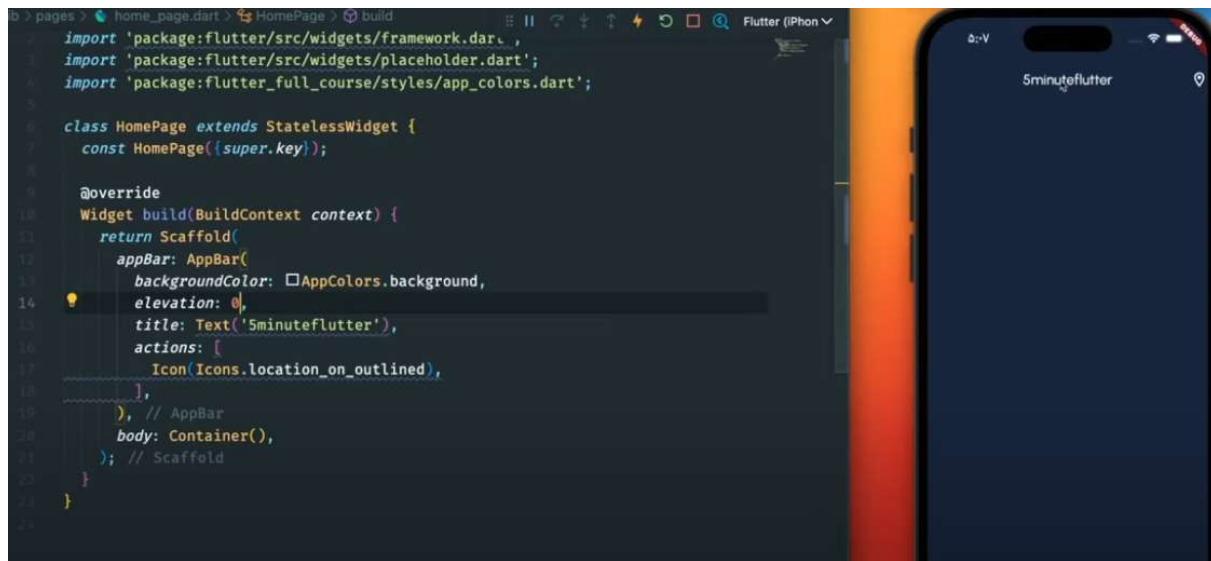


But the appBar is showing us the back button because it has value in the stack

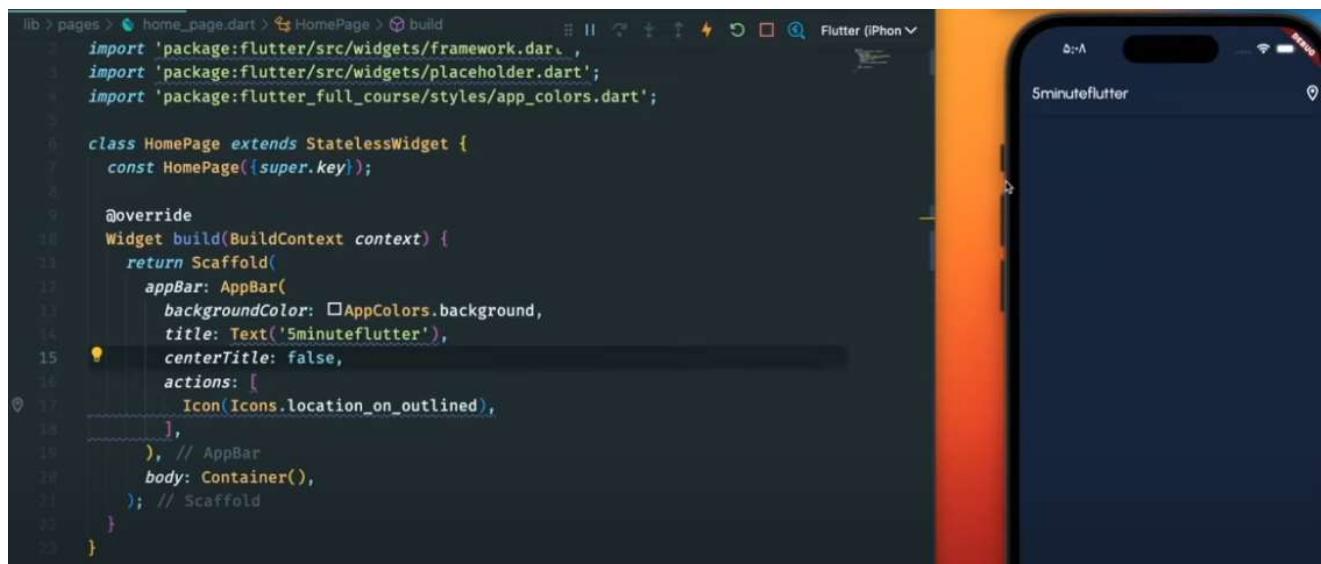
To fix we can: pushReplacementNamed('/home')



- elevation:0 [to remove elevation]



- centerTitle:true [to keep text in center]



Making temp folder for images that will later come from server

The screenshot shows the VS Code interface with the Explorer sidebar on the left displaying a project structure for a Flutter application named 'FLUTTER_FULL_COURSE'. The current file is 'pubspec.yaml' in the 'lib/pages' folder. The code in the editor is the content of the pubspec.yaml file, which includes configuration for lints, flutter-specific settings, and assets.

```
# The "flutter_lints" package below contains a set of configuration options
# encourage good coding practices. The lint set provided by this package is
# activated in the 'analysis_options.yaml' file located in the root of your
# package. See that file for information about deactivating or
# rules and activating additional ones.
flutter_lints: ^2.0.0

# For information on the generic Dart part of this file,
# follow the page: https://dart.dev/tools/pub/pubspec

# The following section is specific to Flutter packages.
flutter:
  uses-material-design: true

  # To add assets to your application, add an assets
  assets:
    - assets/images/
    - assets/temp/
```

• Image

The screenshot shows the VS Code interface with the Editor tab open to 'home_page.dart' in the 'lib/pages' folder. The code defines a Scaffold with an AppBar containing a user icon and title, and a Column body with a Row containing an image and a text label. To the right of the editor is a mobile application preview running on an iPhone X simulator, showing the resulting UI with the title '5minuteflutter' and a user profile card for 'Sarah Fernandez'.

```
appBar: AppBar(
  backgroundColor: AppColors.background,
  title: Text('5minuteflutter'),
  centerTitle: false,
  actions: [
    Icon(Icons.location_on_outlined),
  ],
), // AppBar
body: Column(
  children: [
    Row(
      children: [
        Image.asset(
          'assets/temp/user1.png',
          width: 40,
          height: 40,
        ), // Image.asset
        SizedBox(
          width: 16,
        ), // SizedBox
        Text("Sarah Fernandez"),
      ],
    ), // Row
  ],
); // Column
); // Scaffold
```

To print multiple user list

```
lib > pages > home_page.dart Flutter (iPhone)
      centerTitle: false,
      actions: [
        Icon(Icons.location_on_outlined),
      ],
    ), // AppBar
    body: Column(
      children: mockUsersFromServer(),
    ), // Column
  ); // Scaffold
}

Widget _userItem() {
  return Row(
    children: [
      Image.asset(
        'assets/temp/user1.png',
        width: 40,
        height: 40,
      ), // Image.asset
      SizedBox(
        width: 16,
      ), // SizedBox
      Text("Sarah Fernandez"),
    ],
  ); // Row
}

List<Widget> mockUsersFromServer() {
  List<Widget> users = [];
  for (var i = 0; i < 20; i++) {
    users.add(_userItem());
  }
  return users;
}
```

Output



Updating to “SingleChildScrollView” to view full list

```
body: SingleChildScrollView(  
    child: Column(  
        children: mockUsersFromServer(),  
    ), // Column  
, // SingleChildScrollView  
); // Scaffold  
}
```



Since we are using “Column” widget to show mockusers that's why the list of user is so laggy and “Column” is not recommended to use here instead use “

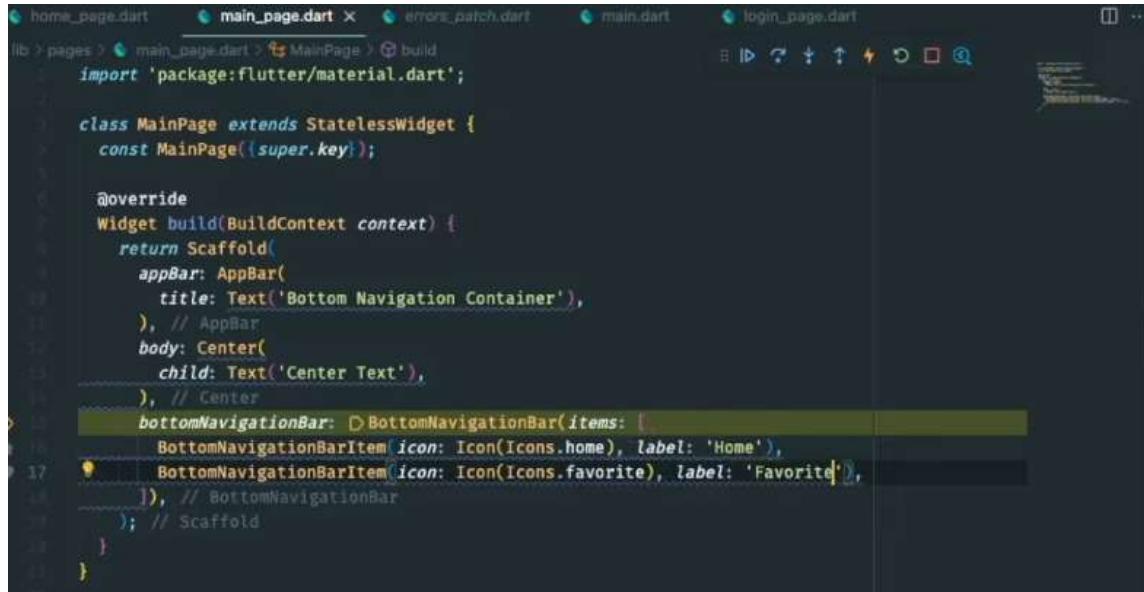
- ListView

```
), // AppBar  
body: ListView(children: mockUsersFromServer()),  
); // Scaffold  
}
```

Now the list of users will not be laggy because ListView uses lazy loading technique to avoid lagging

Making Bottom navigation

- bottomNavigationBar



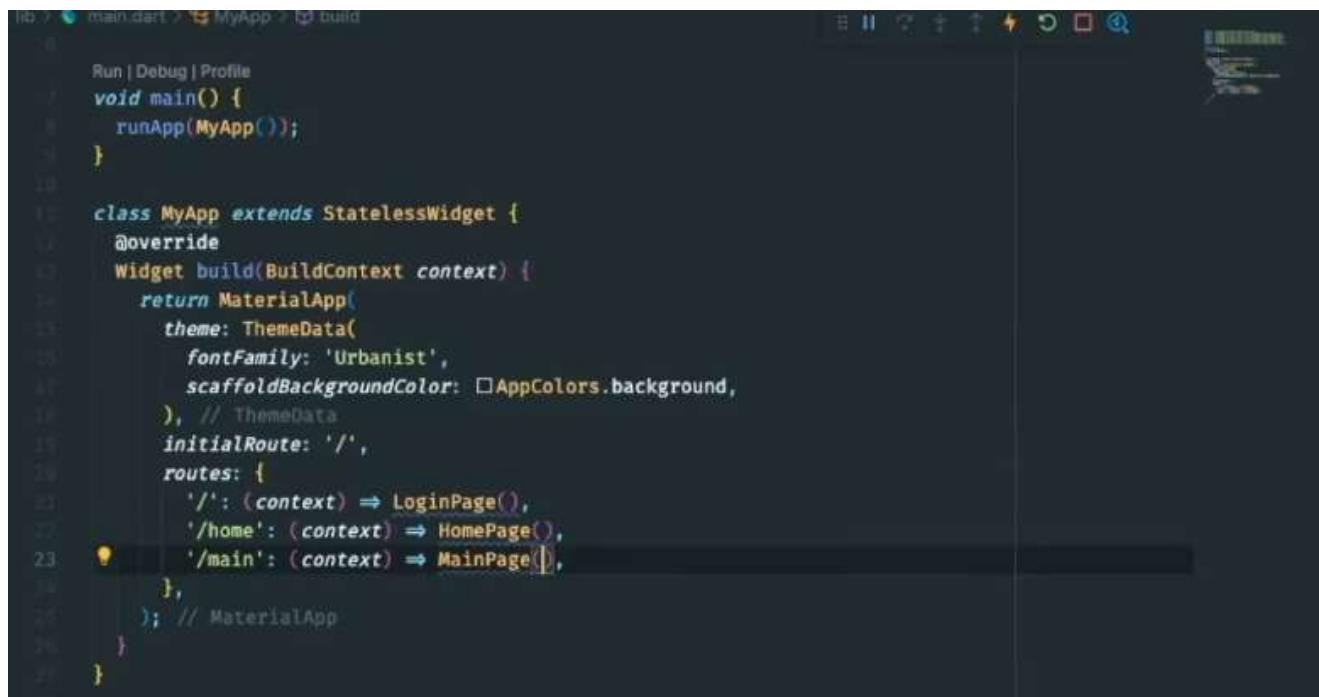
A screenshot of a code editor showing the `MainPage` class. The code includes an `AppBar` with a title and a `body` containing a center text. Below this, a `bottomNavigationBar` is defined with two items: `BottomNavigationBarItem(icon: Icon(Icons.home), label: 'Home')` and `BottomNavigationBarItem(icon: Icon(Icons.favorite), label: 'Favorite')`. The code editor has tabs for `home_page.dart`, `main_page.dart`, `errors_patch.dart`, `main.dart`, and `login_page.dart`.

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

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

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('Bottom Navigation Container'),
      ),
      body: Center(
        child: Text('Center Text'),
      ),
      bottomNavigationBar: BottomNavigationBar(items: [
        BottomNavigationBarItem(icon: Icon(Icons.home), label: 'Home'),
        BottomNavigationBarItem(icon: Icon(Icons.favorite), label: 'Favorite'),
      ]),
    );
  }
}
```

Update main.dart

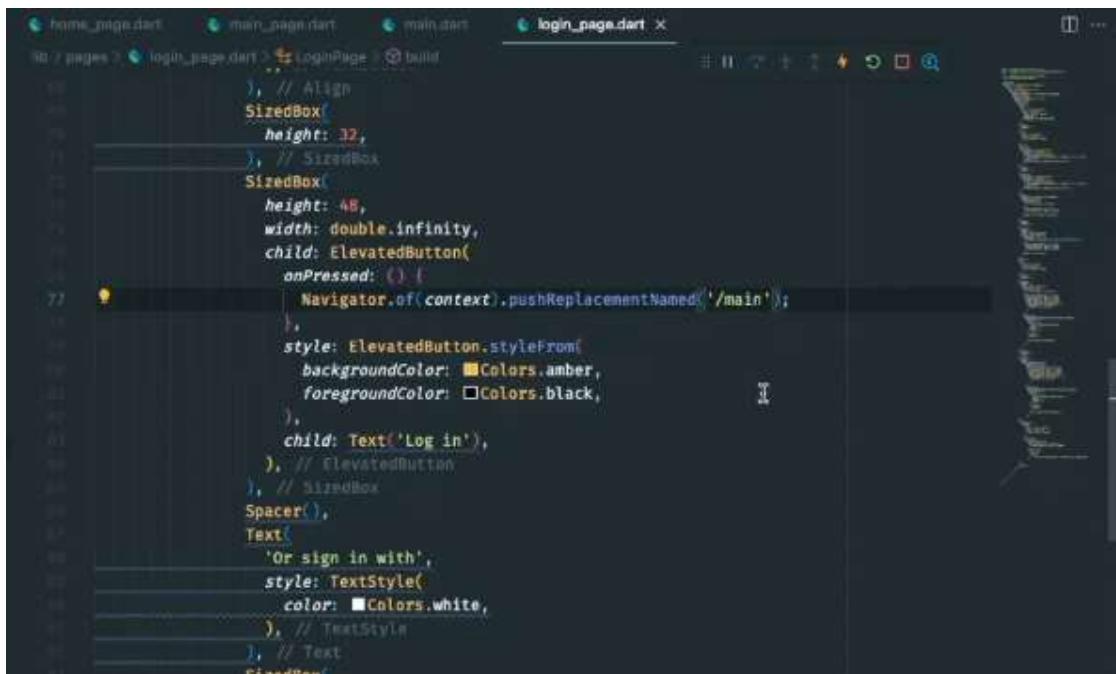


A screenshot of a code editor showing the `main.dart` file. It defines a `void main()` function that runs the `MyApp` widget. The `MyApp` class is a `StatelessWidget` that returns a `MaterialApp` with a theme, scaffold background color, initial route, and routes mapping. The routes map `'/'` to `LoginPage()`, `'/home'` to `HomePage()`, and `'/main'` to `MainPage()`.

```
void main() {
  runApp(MyApp());
}

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      theme: ThemeData(
        fontFamily: 'Urbanist',
        scaffoldBackgroundColor: AppColors.background,
      ),
      initialRoute: '/',
      routes: {
        '/': (context) => LoginPage(),
        '/home': (context) => HomePage(),
        '/main': (context) => MainPage(),
      },
    );
  }
}
```

Update login page to navigate to '/main'

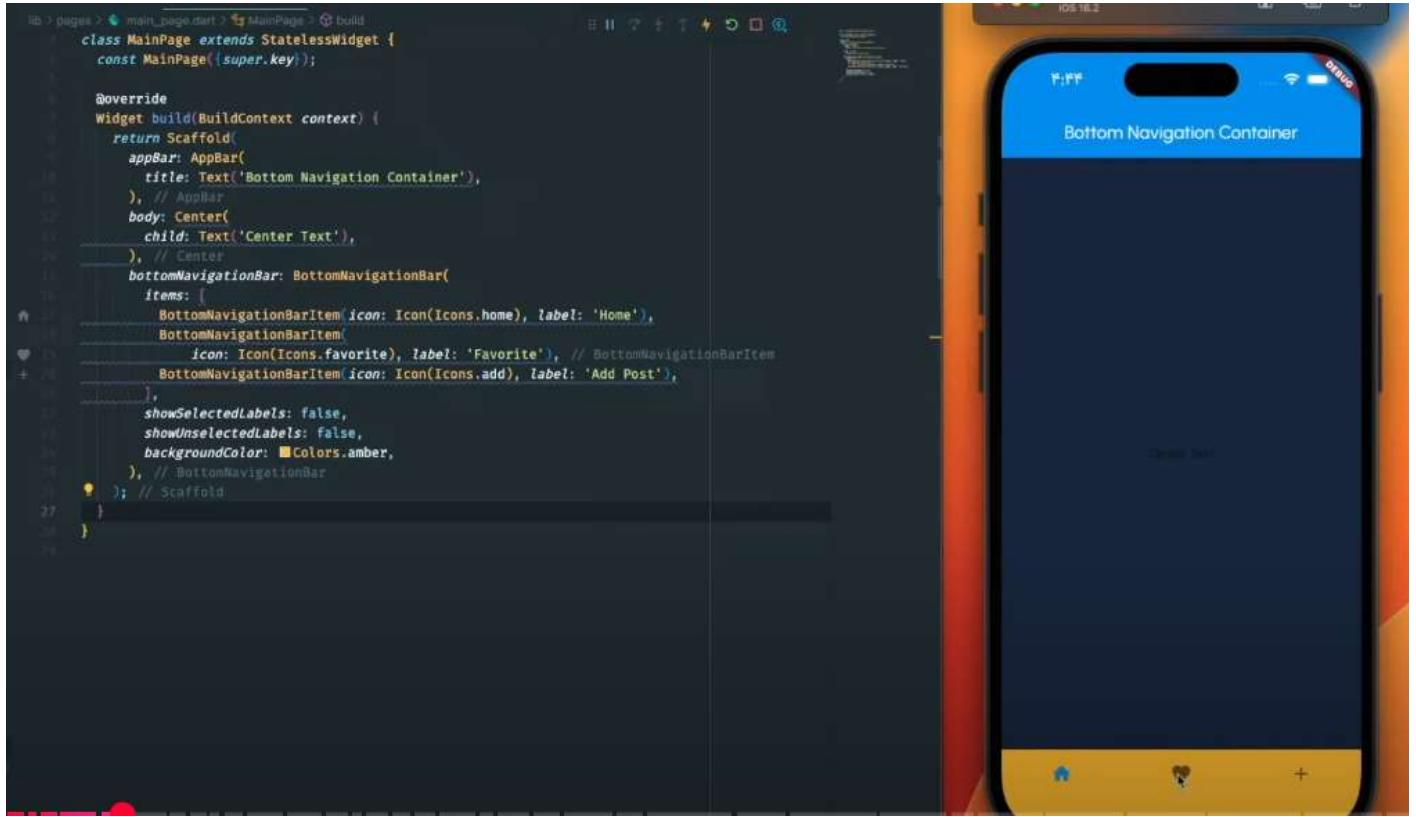


```
home_page.dart main_page.dart main.dart login_page.dart
lib/pages/login_page.dart
    ...
        ), // Align
        SizedBox(
            height: 32,
        ), // SizedBox
        SizedBox(
            height: 48,
            width: double.infinity,
            child: ElevatedButton(
                onPressed: () {
                    Navigator.of(context).pushReplacementNamed('/main');
                },
                style: ElevatedButton.styleFrom(
                    backgroundColor: Colors.amber,
                    foregroundColor: Colors.black,
                ),
                child: Text('Log in'),
            ), // ElevatedButton
        ), // SizedBox
        Spacer(),
        Text(
            'Or sign in with',
            style: TextStyle(
                color: Colors.white,
            ),
        ), // TextStyle
    ), // Text
), // sizedBox
```

Output



Hiding the label text from the bottom of icons



But the other icons are not selectable due to the " StatelessWidget" now we will use " StatefulWidget"

- StatefulWidget

Creating a TestPage file

Update main.dart

```
lib > main.dart > MyApp > build
Run | Debug | Profile
void main() {
  runApp(MyApp());
}

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      theme: ThemeData(
        fontFamily: 'Urbanist',
        scaffoldBackgroundColor: AppColors.background
      ), // ThemeData
      home: TestPage(),
      // initialRoute: '/',
      routes: [
        '/login': (context) => LoginPage(),
        '/home': (context) => HomePage(),
        '/main': (context) => MainPage(),
      ],
    ); // MaterialApp
  }
}
```

- FloatingActionButton()

```
lib > pages > test_page.dart > _TestPageState
import 'package:flutter/src/widgets/placenoder.dart';

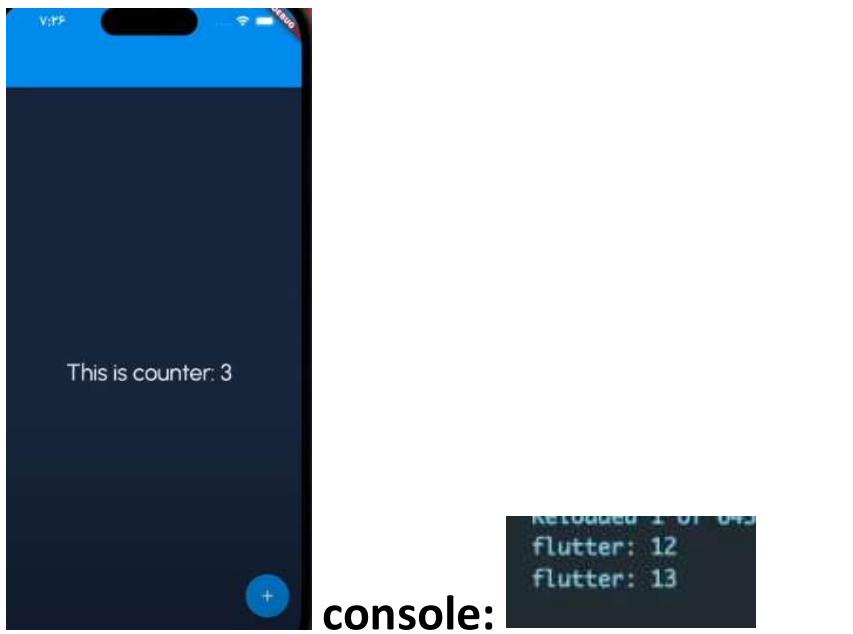
class TestPage extends StatefulWidget {
  const TestPage({super.key});

  @override
  State<TestPage> createState() => _TestPageState();
}

class _TestPageState extends State<TestPage> {
  int count = 3;

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(),
      body: Center(
        child: Text(
          'This is counter: $count',
          style: TextStyle(fontSize: 30, color: Colors.white),
        ), // Text // Center
        floatingActionButton: FloatingActionButton(
          onPressed: () {
            count++;
            print(count);
          },
          child: Icon(Icons.add),
        ), // FloatingActionButton
      ); // Scaffold
  }
}
```

Output: on clicking the button the Ui is not changing but variable is updating



To fix this issue update the value of count using setState()

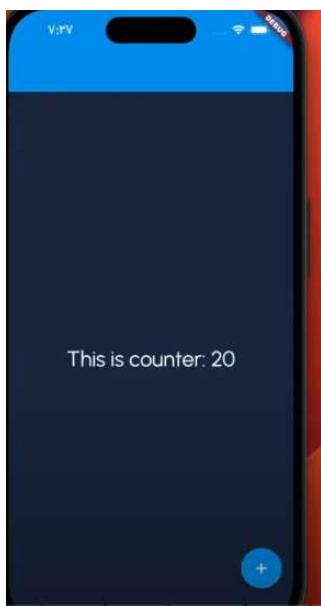
- `setState()` : [to update UI based on variable use it]

```

floatingActionButton: FloatingActionButton(
  onPressed: () {
    setState(() {
      count++;
    });
  },
  child: Icon(Icons.add),
), // FloatingActionButton
); // Scaffold

```

Output: Interaction counter



Now coming back to our bottom navigation

```

lib > pages > main_page.dart > _MainPageState
1 class MainPage extends StatefulWidget {
2   const MainPage({super.key});
3
4   @override
5   State<MainPage> createState() => _MainPageState();
6 }
7
8 class _MainPageState extends State<MainPage> {
9   int currentIndex = 0;
10  @override
11  Widget build(BuildContext context) {
12    return Scaffold(
13      appBar: AppBar(
14        title: Text('Bottom Navigation Container'),
15      ), // AppBar
16      body: Center(
17        child: Text('Center Text'),
18      ), // Center
19      bottomNavigationBar: BottomNavigationBar(
20        items: [
21          BottomNavigationBarItem(icon: Icon(Icons.home), label: 'Home'),
22          BottomNavigationBarItem(
23            icon: Icon(Icons.favorite), label: 'Favorite'), // BottomNavigationBa
24          BottomNavigationBarItem(icon: Icon(Icons.add), label: 'Add Post'),
25        ],
26      ),
27    );
28  }
29}

```

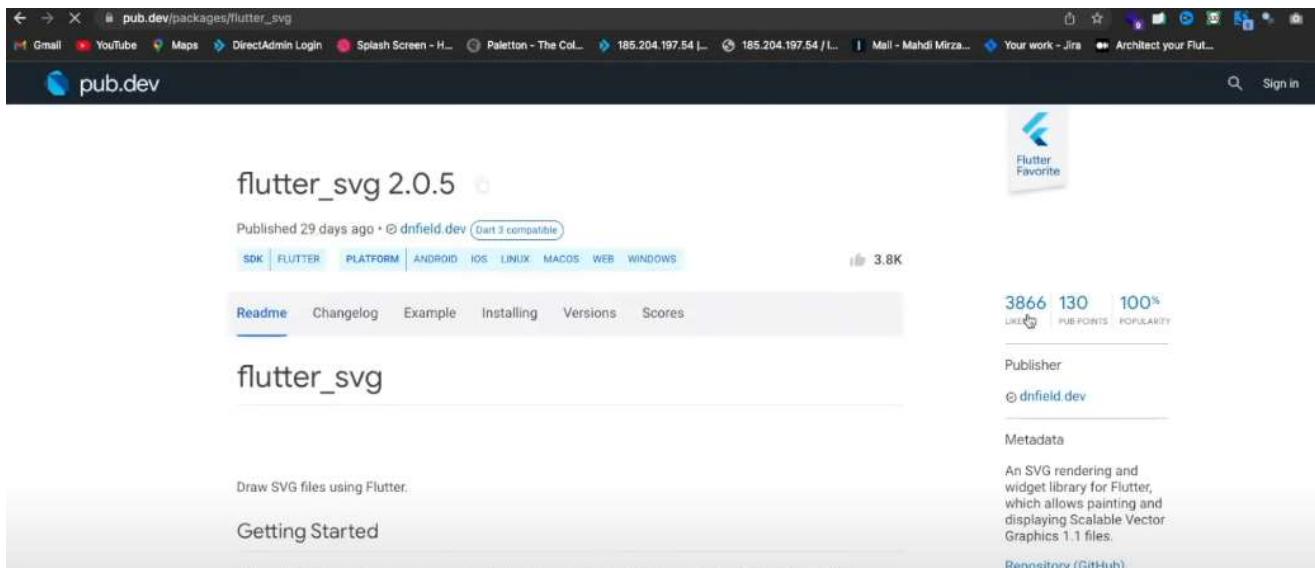
```
currentIndex: currentIndex,
onTap: (index) {
  setState(() {
    currentIndex = index;
  });
},
showSelectedLabels: false,
showUnselectedLabels: false,
backgroundColor: Colors.amber,
), // BottomNavigationBar
```

Output



• Using SVGs in flutter

Installing package

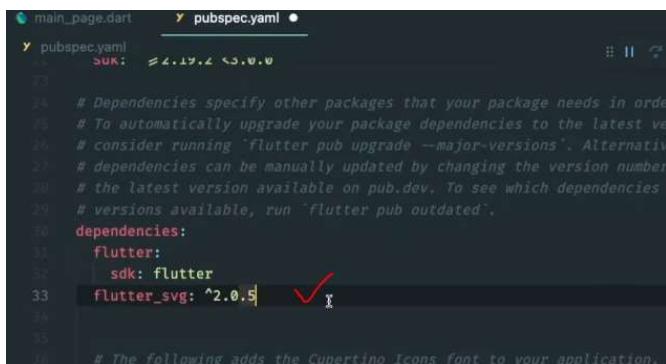


The screenshot shows the pub.dev website with the URL pub.dev/packages/flutter_svg. The page displays the **flutter_svg 2.0.5** package. Key details include:

- Published 29 days ago by [dnfield dev](#) (Dart 3 compatible)
- Flutter library
- 3.8K stars
- 3866 likes, 130 pub points, 100% popularity
- Publisher: [dnfield dev](#)
- Metadata: An SVG rendering and widget library for Flutter, which allows painting and displaying Scalable Vector Graphics 1.1 files.
- Repository: [GitHub](#)

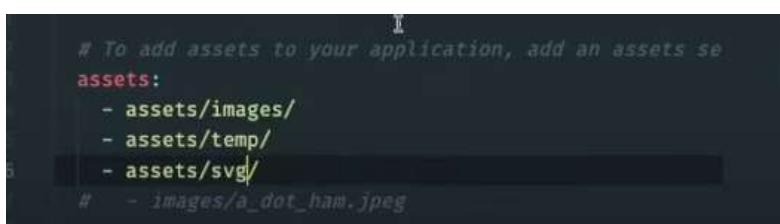
The package description states: "Draw SVG files using Flutter." and includes a "Getting Started" section.

**To save in your project just add in pubspec.yaml and run
“flutter pub get”**

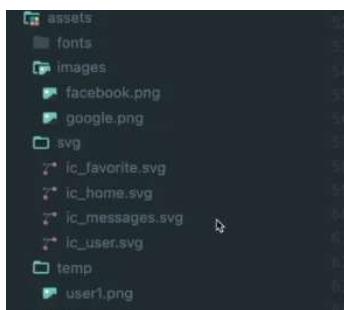


```
# Dependencies specify other packages that your package needs in order to work.
# To automatically upgrade your package dependencies to the latest version
# consider running 'flutter pub upgrade --major-versions'. Alternatively,
# dependencies can be manually updated by changing the version number
# to the latest version available on pub.dev. To see which dependencies
# versions are available, run 'flutter pub outdated'.
dependencies:
  flutter:
    sdk: flutter
  flutter_svg: ^2.0.5
```

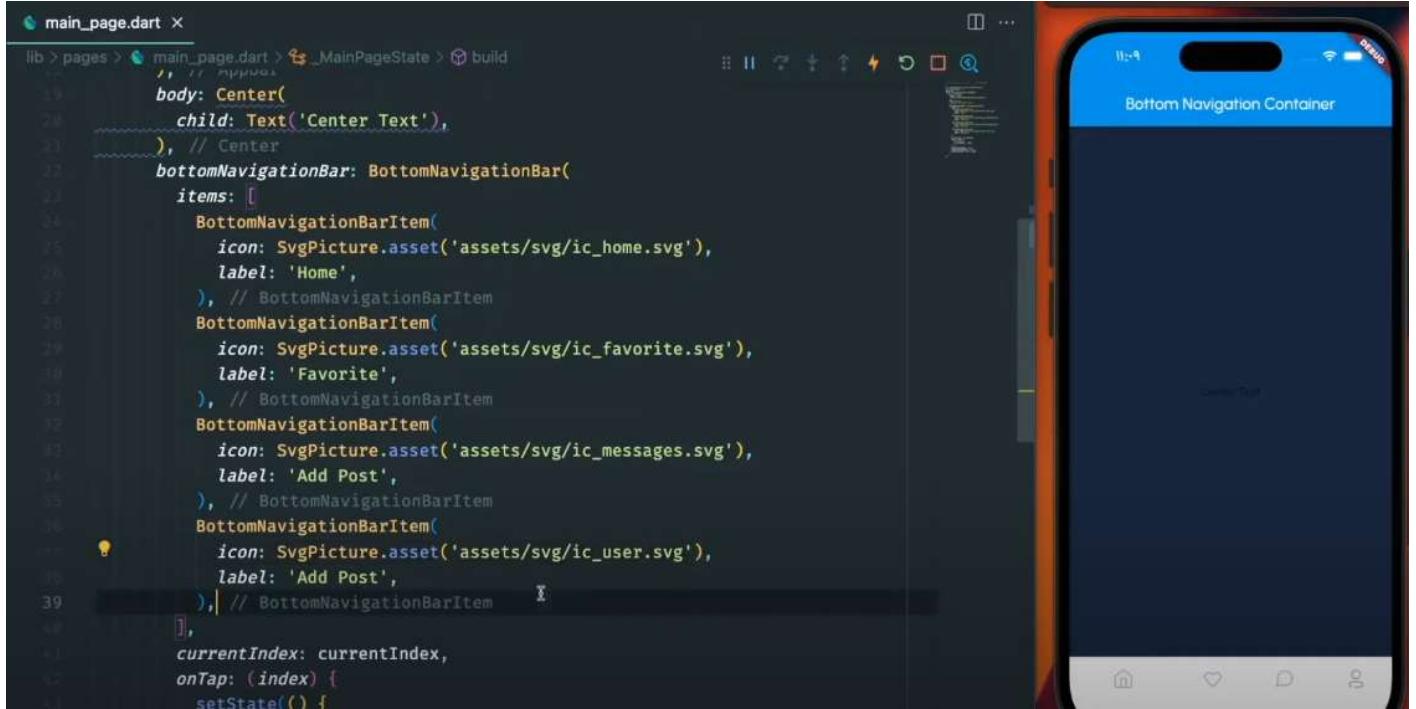
Creating diff folder for svg



```
# To add assets to your application, add an assets section here
assets:
  - assets/images/
  - assets/temp/
  - assets/svg/
  # - images/a_dot_ham.jpeg
```



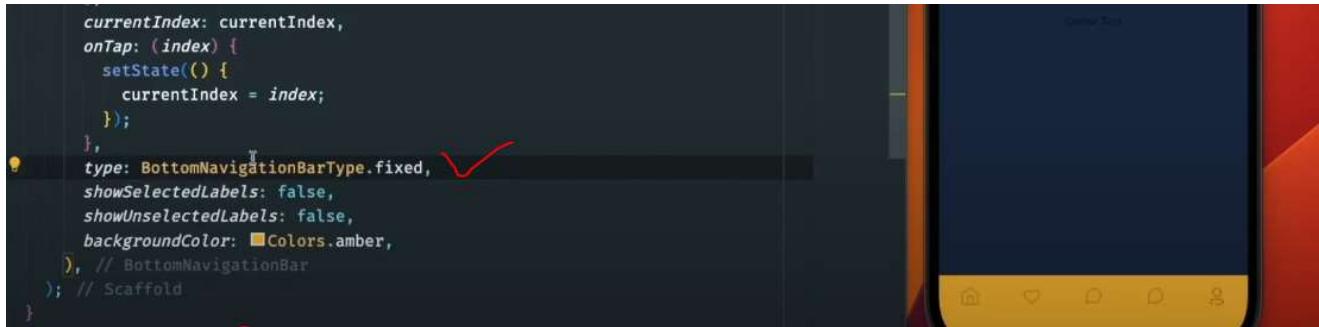
Using in the project



The screenshot shows the code for `main_page.dart` in a Flutter IDE. The code defines a `BottomNavigationBar` with four items: Home, Favorite, Add Post, and another Add Post. The `type` is set to `BottomNavigationBarType.fixed`. The preview on the right shows a black iPhone X with a blue header and a white bottom navigation bar containing four yellow icons labeled Home, Favorite, Add Post, and another Add Post.

```
lib > pages > main_page.dart > _MainPageState > build
  ...
  body: Center(
    child: Text('Center Text'),
  ), // Center
  bottomNavigationBar: BottomNavigationBar(
    items: [
      BottomNavigationBarItem(
        icon: SvgPicture.asset('assets/svg/ic_home.svg'),
        label: 'Home',
      ), // BottomNavigationBarItem
      BottomNavigationBarItem(
        icon: SvgPicture.asset('assets/svg/ic_favorite.svg'),
        label: 'Favorite',
      ), // BottomNavigationBarItem
      BottomNavigationBarItem(
        icon: SvgPicture.asset('assets/svg/ic_messages.svg'),
        label: 'Add Post',
      ), // BottomNavigationBarItem
      BottomNavigationBarItem(
        icon: SvgPicture.asset('assets/svg/ic_user.svg'),
        label: 'Add Post',
      ), // BottomNavigationBarItem
    ],
    currentIndex: currentIndex,
    onTap: (index) {
      setState(() {
        currentIndex = index;
      });
    },
  );
}
```

To see the color again: `BottomNavigationBarType:fixed`



The screenshot shows the same `main_page.dart` code as above, but with a red checkmark placed over the line where `type: BottomNavigationBarType.fixed` is defined. The preview on the right shows the same black iPhone X with a blue header and a white bottom navigation bar containing four yellow icons.

```
currentIndex: currentIndex,
onTap: (index) {
  setState(() {
    currentIndex = index;
  });
},
type: BottomNavigationBarType.fixed, ✓
showSelectedLabels: false,
showUnselectedLabels: false,
backgroundColor: Colors.amber,
), // BottomNavigationBar
); // Scaffold
}
```

Applying HomePage() class In body and removing appBar



```
lib > pages > main_page.dart > _MainPageState > build
  const MainPage({super.key});

  @override
  State<MainPage> createState() => _MainPageState();
}

class _MainPageState extends State<MainPage> {
  int currentIndex = 0;
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: HomePage(),
      bottomNavigationBar: BottomNavigationBar(
        items: [
          BottomNavigationBarItem(
            icon: SvgPicture.asset('assets/svg/ic_home.svg'),
            label: 'Home',
          ), // BottomNavigationBarItem
          BottomNavigationBarItem(
            icon: SvgPicture.asset('assets/svg/ic_favorite.svg'),
            label: 'Favorite',
          ), // BottomNavigationBarItem
          BottomNavigationBarItem(
            icon: SvgPicture.asset('assets/svg/ic_messages.svg'),
            label: 'Add Post',
          ),
        ],
      ),
    );
  }
}
```

To make the bottom navigation working doing simulation

```
final pages = [
  HomePage(),
  Center(
    child: Text('Favorite'),
  ),
  Center(
    child: Text('Add Post'),
  ),
  Center(
    child: Text('Messages'),
  ),
  Center(
    child: Text('User'),
  ),
];
}

class _MainPageState extends State<MainPage> {
  int currentIndex = 0;
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: pages[currentIndex], ✓
      bottomNavigationBar: BottomNavigationBar(
        items: [

```

Output: bottom navigation button working based on index

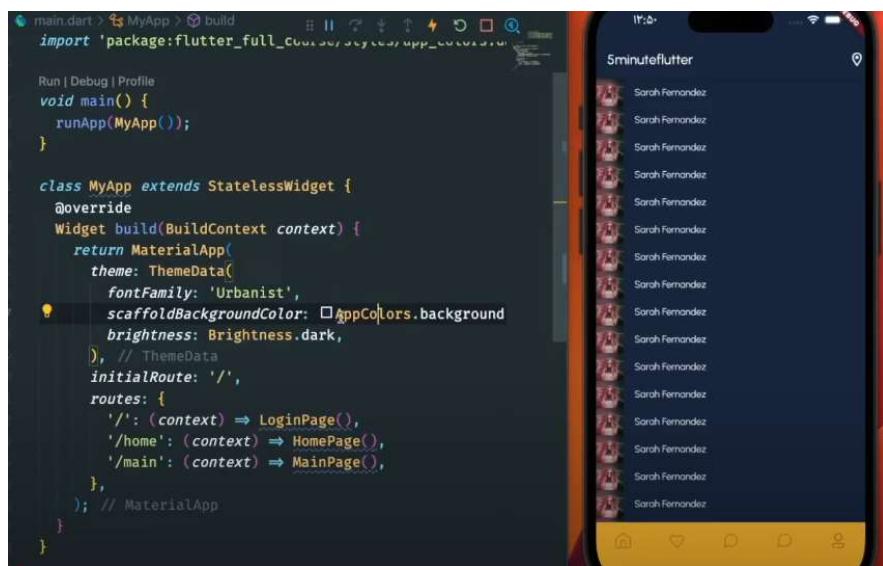


Creating file: app text.dart

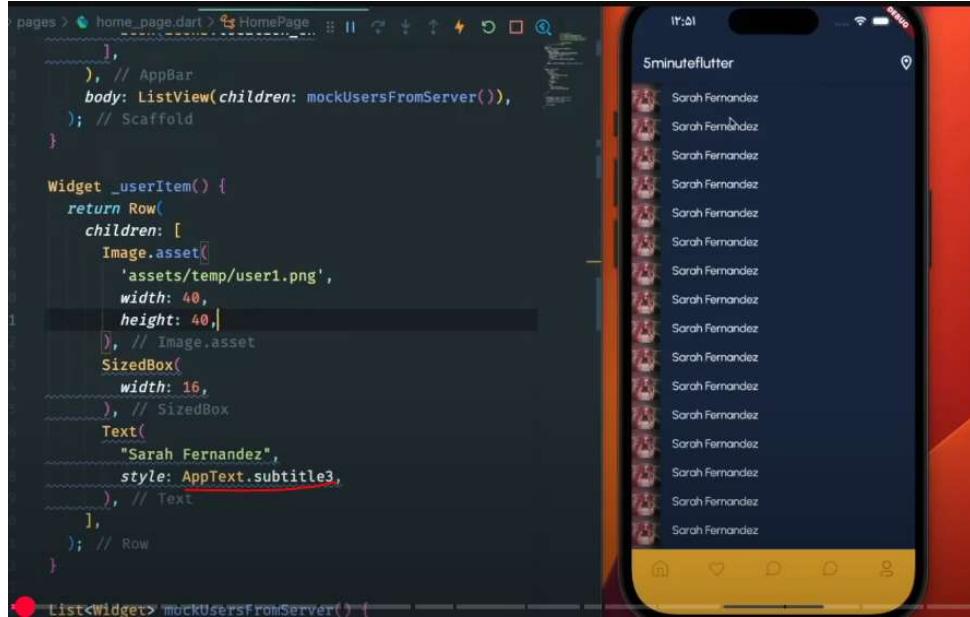
The screenshot shows the Android Studio interface with the following details:

- Project Structure:** The left sidebar shows the project structure under "FLUTTER_FULL COURSE". It includes "temp", "build", "ios", "lib" (containing "pages", "styles", "app_colors.dart", "app_text.dart", and "main.dart"), "linux", "macos", "test", "web", and "windows".
- Open Editors:** Below the project structure, there are three open editors:
 - "main_page.dart" in the "lib/pages" folder.
 - "main.dart" in the "lib" folder.
 - "app_colors.dart" in the "lib/styles" folder.
- Code Editor:** The main window displays the "app_text.dart" file from the "lib/styles" folder. The code defines several static const TextStyle variables for different font styles: header1, header2, subtitle1, subtitle2, subtitle3, and body1, each with a font size of 24 and a specific weight.
- Bottom Navigation:** The bottom navigation bar includes tabs for "OUTLINE", "PROBLEMS", "RUN", and "DEPENDENCIES".

- **Brightness inside Theme**



Using AppText() class



Create a folder named “components”

```
FLUTTER_FULL_COURSE
  ...
  lib > components > post_item.dart > PostItem
    import 'package:flutter/material.dart';
    import '../styles/app_text.dart';

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

      @override
      Widget build(BuildContext context) {
        return Row(
          children: [
            Image.asset(
              'assets/temp/user1.png',
              width: 40,
              height: 40,
            ),
            SizedBox(
              width: 16,
            ),
            Text(
              "Sarah Fernandez",
              style: AppText.subtitle3,
            ),
          ],
        );
      }
    }

  pages
    home_page.dart
    login_page.dart
    main_page.dart
  styles
    app_colors.dart

TIMELINE
OPEN EDITORS
  main_page.dart lib/pages
  main.dart lib
  home_page.dart lib/pages 1
  post_item.dart lib/components
  app_colors.dart lib/styles

OUTLINE
```

Using component



```
lib > pages > home_page.dart > HomePage
  15
  16
  17 @override
  18 Widget build(BuildContext context) {
  19   return Scaffold(
  20     appBar: AppBar(
  21       backgroundColor: AppColors.background,
  22       title: Text('5minuteflutter'),
  23       centerTitle: false,
  24       actions: [
  25         Icon(Icons.location_on_outlined),
  26       ],
  27     ), // AppBar
  28     body: ListView(children: mockUsersFromServer(),
  29   ); // Scaffold
  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
  73
  74
  75
  76
  77
  78
  79
  80
  81
  82
  83
  84
  85
  86
  87
  88
  89
  90
  91
  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
  117
  118
  119
  120
  121
  122
  123
  124
  125
  126
  127
  128
  129
  130
  131
  132
  133
  134
  135
  136
  137
  138
  139
  140
  141
  142
  143
  144
  145
  146
  147
  148
  149
  150
  151
  152
  153
  154
  155
  156
  157
  158
  159
  160
  161
  162
  163
  164
  165
  166
  167
  168
  169
  170
  171
  172
  173
  174
  175
  176
  177
  178
  179
  180
  181
  182
  183
  184
  185
  186
  187
  188
  189
  190
  191
  192
  193
  194
  195
  196
  197
  198
  199
  200
  201
  202
  203
  204
  205
  206
  207
  208
  209
  210
  211
  212
  213
  214
  215
  216
  217
  218
  219
  220
  221
  222
  223
  224
  225
  226
  227
  228
  229
  230
  231
  232
  233
  234
  235
  236
  237
  238
  239
  240
  241
  242
  243
  244
  245
  246
  247
  248
  249
  250
  251
  252
  253
  254
  255
  256
  257
  258
  259
  260
  261
  262
  263
  264
  265
  266
  267
  268
  269
  270
  271
  272
  273
  274
  275
  276
  277
  278
  279
  280
  281
  282
  283
  284
  285
  286
  287
  288
  289
  290
  291
  292
  293
  294
  295
  296
  297
  298
  299
  300
  301
  302
  303
  304
  305
  306
  307
  308
  309
  310
  311
  312
  313
  314
  315
  316
  317
  318
  319
  320
  321
  322
  323
  324
  325
  326
  327
  328
  329
  330
  331
  332
  333
  334
  335
  336
  337
  338
  339
  340
  341
  342
  343
  344
  345
  346
  347
  348
  349
  350
  351
  352
  353
  354
  355
  356
  357
  358
  359
  360
  361
  362
  363
  364
  365
  366
  367
  368
  369
  370
  371
  372
  373
  374
  375
  376
  377
  378
  379
  380
  381
  382
  383
  384
  385
  386
  387
  388
  389
  390
  391
  392
  393
  394
  395
  396
  397
  398
  399
  400
  401
  402
  403
  404
  405
  406
  407
  408
  409
  410
  411
  412
  413
  414
  415
  416
  417
  418
  419
  420
  421
  422
  423
  424
  425
  426
  427
  428
  429
  430
  431
  432
  433
  434
  435
  436
  437
  438
  439
  440
  441
  442
  443
  444
  445
  446
  447
  448
  449
  450
  451
  452
  453
  454
  455
  456
  457
  458
  459
  460
  461
  462
  463
  464
  465
  466
  467
  468
  469
  470
  471
  472
  473
  474
  475
  476
  477
  478
  479
  480
  481
  482
  483
  484
  485
  486
  487
  488
  489
  490
  491
  492
  493
  494
  495
  496
  497
  498
  499
  500
  501
  502
  503
  504
  505
  506
  507
  508
  509
  510
  511
  512
  513
  514
  515
  516
  517
  518
  519
  520
  521
  522
  523
  524
  525
  526
  527
  528
  529
  530
  531
  532
  533
  534
  535
  536
  537
  538
  539
  540
  541
  542
  543
  544
  545
  546
  547
  548
  549
  550
  551
  552
  553
  554
  555
  556
  557
  558
  559
  560
  561
  562
  563
  564
  565
  566
  567
  568
  569
  570
  571
  572
  573
  574
  575
  576
  577
  578
  579
  580
  581
  582
  583
  584
  585
  586
  587
  588
  589
  590
  591
  592
  593
  594
  595
  596
  597
  598
  599
  600
  601
  602
  603
  604
  605
  606
  607
  608
  609
  610
  611
  612
  613
  614
  615
  616
  617
  618
  619
  620
  621
  622
  623
  624
  625
  626
  627
  628
  629
  630
  631
  632
  633
  634
  635
  636
  637
  638
  639
  640
  641
  642
  643
  644
  645
  646
  647
  648
  649
  650
  651
  652
  653
  654
  655
  656
  657
  658
  659
  660
  661
  662
  663
  664
  665
  666
  667
  668
  669
  670
  671
  672
  673
  674
  675
  676
  677
  678
  679
  680
  681
  682
  683
  684
  685
  686
  687
  688
  689
  690
  691
  692
  693
  694
  695
  696
  697
  698
  699
  700
  701
  702
  703
  704
  705
  706
  707
  708
  709
  710
  711
  712
  713
  714
  715
  716
  717
  718
  719
  720
  721
  722
  723
  724
  725
  726
  727
  728
  729
  730
  731
  732
  733
  734
  735
  736
  737
  738
  739
  740
  741
  742
  743
  744
  745
  746
  747
  748
  749
  750
  751
  752
  753
  754
  755
  756
  757
  758
  759
  760
  761
  762
  763
  764
  765
  766
  767
  768
  769
  770
  771
  772
  773
  774
  775
  776
  777
  778
  779
  780
  781
  782
  783
  784
  785
  786
  787
  788
  789
  790
  791
  792
  793
  794
  795
  796
  797
  798
  799
  800
  801
  802
  803
  804
  805
  806
  807
  808
  809
  810
  811
  812
  813
  814
  815
  816
  817
  818
  819
  820
  821
  822
  823
  824
  825
  826
  827
  828
  829
  830
  831
  832
  833
  834
  835
  836
  837
  838
  839
  840
  841
  842
  843
  844
  845
  846
  847
  848
  849
  850
  851
  852
  853
  854
  855
  856
  857
  858
  859
  860
  861
  862
  863
  864
  865
  866
  867
  868
  869
  870
  871
  872
  873
  874
  875
  876
  877
  878
  879
  880
  881
  882
  883
  884
  885
  886
  887
  888
  889
  890
  891
  892
  893
  894
  895
  896
  897
  898
  899
  900
  901
  902
  903
  904
  905
  906
  907
  908
  909
  910
  911
  912
  913
  914
  915
  916
  917
  918
  919
  920
  921
  922
  923
  924
  925
  926
  927
  928
  929
  930
  931
  932
  933
  934
  935
  936
  937
  938
  939
  940
  941
  942
  943
  944
  945
  946
  947
  948
  949
  950
  951
  952
  953
  954
  955
  956
  957
  958
  959
  960
  961
  962
  963
  964
  965
  966
  967
  968
  969
  970
  971
  972
  973
  974
  975
  976
  977
  978
  979
  980
  981
  982
  983
  984
  985
  986
  987
  988
  989
  990
  991
  992
  993
  994
  995
  996
  997
  998
  999
  1000
  1001
  1002
  1003
  1004
  1005
  1006
  1007
  1008
  1009
  1010
  1011
  1012
  1013
  1014
  1015
  1016
  1017
  1018
  1019
  1020
  1021
  1022
  1023
  1024
  1025
  1026
  1027
  1028
  1029
  1030
  1031
  1032
  1033
  1034
  1035
  1036
  1037
  1038
  1039
  1040
  1041
  1042
  1043
  1044
  1045
  1046
  1047
  1048
  1049
  1050
  1051
  1052
  1053
  1054
  1055
  1056
  1057
  1058
  1059
  1060
  1061
  1062
  1063
  1064
  1065
  1066
  1067
  1068
  1069
  1070
  1071
  1072
  1073
  1074
  1075
  1076
  1077
  1078
  1079
  1080
  1081
  1082
  1083
  1084
  1085
  1086
  1087
  1088
  1089
  1090
  1091
  1092
  1093
  1094
  1095
  1096
  1097
  1098
  1099
  1100
  1101
  1102
  1103
  1104
  1105
  1106
  1107
  1108
  1109
  1110
  1111
  1112
  1113
  1114
  1115
  1116
  1117
  1118
  1119
  1120
  1121
  1122
  1123
  1124
  1125
  1126
  1127
  1128
  1129
  1130
  1131
  1132
  1133
  1134
  1135
  1136
  1137
  1138
  1139
  1140
  1141
  1142
  1143
  1144
  1145
  1146
  1147
  1148
  1149
  1150
  1151
  1152
  1153
  1154
  1155
  1156
  1157
  1158
  1159
  1160
  1161
  1162
  1163
  1164
  1165
  1166
  1167
  1168
  1169
  1170
  1171
  1172
  1173
  1174
  1175
  1176
  1177
  1178
  1179
  1180
  1181
  1182
  1183
  1184
  1185
  1186
  1187
  1188
  1189
  1190
  1191
  1192
  1193
  1194
  1195
  1196
  1197
  1198
  1199
  1200
  1201
  1202
  1203
  1204
  1205
  1206
  1207
  1208
  1209
  1210
  1211
  1212
  1213
  1214
  1215
  1216
  1217
  1218
  1219
  1220
  1221
  1222
  1223
  1224
  1225
  1226
  1227
  1228
  1229
  1230
  1231
  1232
  1233
  1234
  1235
  1236
  1237
  1238
  1239
  1240
  1241
  1242
  1243
  1244
  1245
  1246
  1247
  1248
  1249
  1250
  1251
  1252
  1253
  1254
  1255
  1256
  1257
  1258
  1259
  1260
  1261
  1262
  1263
  1264
  1265
  1266
  1267
  1268
  1269
  1270
  1271
  1272
  1273
  1274
  1275
  1276
  1277
  1278
  1279
  1280
  1281
  1282
  1283
  1284
  1285
  1286
  1287
  1288
  1289
  1290
  1291
  1292
  1293
  1294
  1295
  1296
  1297
  1298
  1299
  1300
  1301
  1302
  1303
  1304
  1305
  1306
  1307
  1308
  1309
  1310
  1311
  1312
  1313
  1314
  1315
  1316
  1317
  1318
  1319
  1320
  1321
  1322
  1323
  1324
  1325
  1326
  1327
  1328
  1329
  1330
  1331
  1332
  1333
  1334
  1335
  1336
  1337
  1338
  1339
  1340
  1341
  1342
  1343
  1344
  1345
  1346
  1347
  1348
  1349
  1350
  1351
  1352
  1353
  1354
  1355
  1356
  1357
  1358
  1359
  1360
  1361
  1362
  1363
  1364
  1365
  1366
  1367
  1368
  1369
  1370
  1371
  1372
  1373
  1374
  1375
  1376
  1377
  1378
  1379
  1380
  1381
  1382
  1383
  1384
  1385
  1386
  1387
  1388
  1389
  1390
  1391
  1392
  1393
  1394
  1395
  1396
  1397
  1398
  1399
  1400
  1401
  1402
  1403
  1404
  1405
  1406
  1407
  1408
  1409
  1410
  1411
  1412
  1413
  1414
  1415
  1416
  1417
  1418
  1419
  1420
  1421
  1422
  1423
  1424
  1425
  1426
  1427
  1428
  1429
  1430
  1431
  1432
  1433
  1434
  1435
  1436
  1437
  1438
  1439
  1440
  1441
  1442
  1443
  1444
  1445
  1446
  1447
  1448
  1449
  1450
  1451
  1452
  1453
  1454
  1455
  1456
  1457
  1458
  1459
  1460
  1461
  1462
  1463
  1464
  1465
  1466
  1467
  1468
  1469
  1470
  1471
  1472
  1473
  1474
  1475
  1476
  1477
  1478
  1479
  1480
  1481
  1482
  1483
  1484
  1485
  1486
  1487
  1488
  1489
  1490
  1491
  1492
  1493
  1494
  1495
  1496
  1497
  1498
  1499
  1500
  1501
  1502
  1503
  1504
  1505
  1506
  1507
  1508
  1509
  1510
  1511
  1512
  1513
  1514
  1515
  1516
  1517
  1518
  1519
  1520
  1521
  1522
  1523
  1524
  1525
  1526
  1527
  1528
  1529
  1530
  1531
  1532
  1533
  1534
  1535
  1536
  1537
  1538
  1539
  1540
  1541
  1542
  1543
  1544
  1545
  1546
  1547
  1548
  1549
  1550
  1551
  1552
  1553
  1554
  1555
  1556
  1557
  1558
  1559
  1560
  1561
  1562
  1563
  1564
  1565
  1566
  1567
  1568
  1569
  1570
  1571
  1572
  1573
  1574
  1575
  1576
  1577
  1578
  1579
  1580
  1581
  1582
  1583
  1584
  1585
  1586
  1587
  1588
  1589
  1590
  1591
  1592
  1593
  1594
  1595
  1596
  1597
  1598
  1599
  1600
  1601
  1602
  1603
  1604
  1605
  1606
  1607
  1608
  1609
  1610
  1611
  1612
  1613
  1614
  1615
  1616
  1617
  1618
  1619
  1620
  1621
  1622
  1623
  1624
  1625
  1626
  1627
  1628
  1629
  1630
  1631
  1632
  1633
  1634
  1635
  1636
  1637
  1638
  1639
  1640
  1641
  1642
  1643
  1644
  1645
  1646
  1647
  1648
  1649
  1650
  1651
  1652
  1653
  1654
  1655
  1656
  1657
  1658
  1659
  1660
  1661
  1662
  1663
  1664
  1665
  1666
  1667
  1668
  1669
  1670
  1671
  1672
  1673
  1674
  1675
  1676
  1677
  1678
  1679
  1680
  1681
  1682
  1683
  1684
  1685
  1686
  1687
  1688
  1689
  1690
  1691
  1692
  1693
  1694
  1695
  1696
  1697
  1698
  1699
  1700
  1701
  1702
  1703
  1704
  1705
  1706
  1707
  1708
  1709
  1710
  1711
  1712
  1713
  1714
  1715
  1716
  1717
  1718
  1719
  1720
  1721
  1722
  1723
  1724
  1725
  1726
  1727
  1728
  1729
  1730
  1731
  1732
  1733
  1734
  1735
  1736
  1737
  1738
  1739
  1740
  1741
  1742
  1743
  1744
  1745
  1746
  1747
  1748
  1749
  1750
  1751
  1752
  1753
  1754
  1755
  1756
  1757
  1758
  1759
  1760
  1761
  1762
  1763
  1764
  1765
  1766
  17
```

Adding Padding by wrapping with padding and giving the current return widget as child of Padding widget



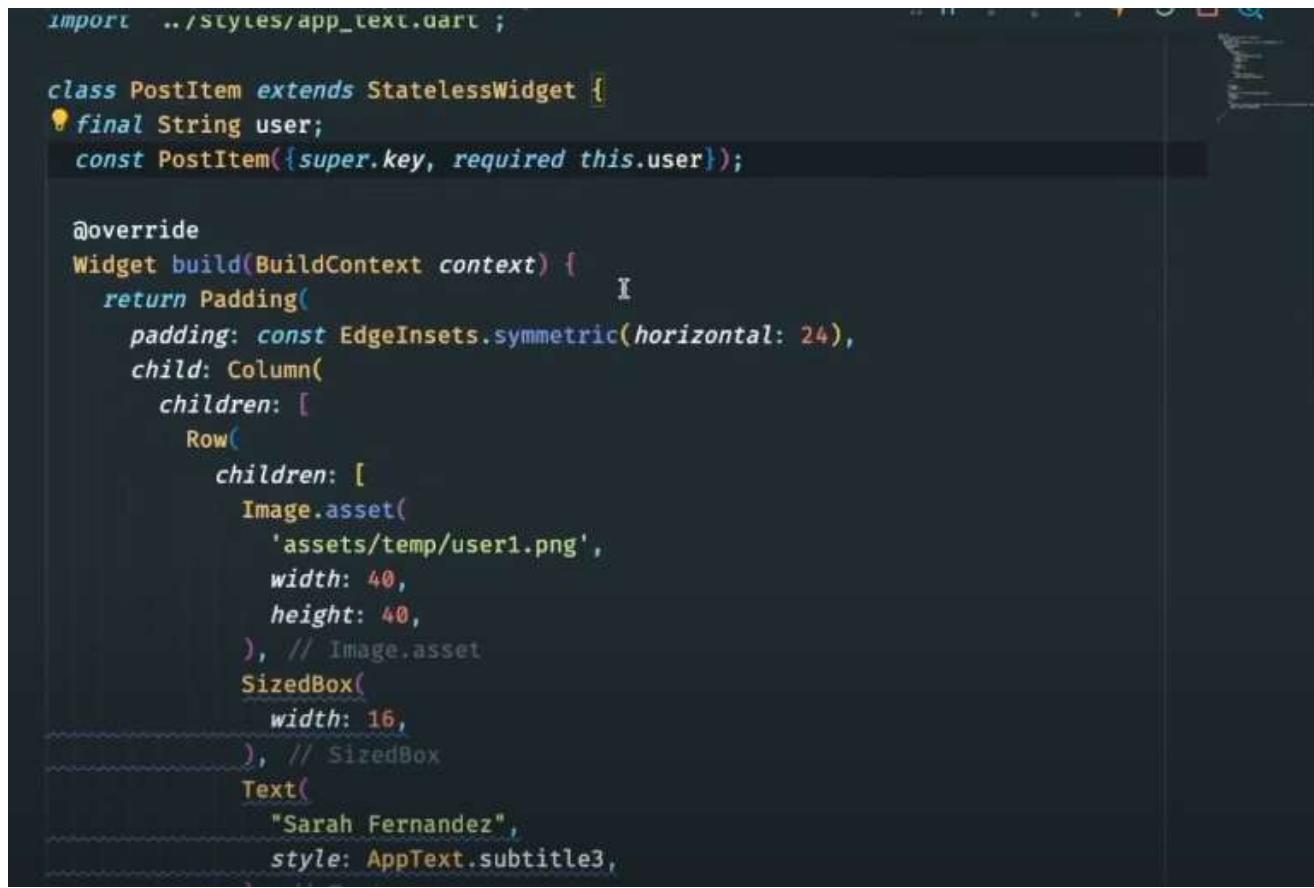
```
lib/components/post_item.dart:11:11: Error: The argument type 'Widget' can't be assigned to the parameter type 'EdgeInsets'.  
      padding: const EdgeInsets.symmetric(horizontal: 24),  
           ^  
lib/components/post_item.dart:11:11: Error: The argument type 'Widget' can't be assigned to the parameter type 'EdgeInsets'.  
      padding: const EdgeInsets.symmetric(horizontal: 24),  
           ^  
  
  1 import '.../styles/app_text.dart';  
  2  
  3 class PostItem extends StatelessWidget {  
  4   const PostItem({super.key});  
  5  
  6   @override  
  7   Widget build(BuildContext context) {  
  8     return Padding(  
  9       padding: const EdgeInsets.symmetric(horizontal: 24),  
10       child: Column(  
11         children: [  
12           Row(  
13             children: [  
14               Image.asset(  
15                 'assets/temp/user1.png',  
16                 width: 40,  
17                 height: 40,  
18               ), // Image.asset  
19               SizedBox(  
20                 width: 16,  
21               ), // SizedBox  
22             ],  
23           ), // Row  
24           SizedBox(  
25             height: 12,  
26           ), // SizedBox  
27         ],  
28       ), // Column  
29     );  
30   }  
31 }
```

Giving space between user name and post



```
lib/components/post_item.dart:11:11: Error: The argument type 'Widget' can't be assigned to the parameter type 'EdgeInsets'.  
      padding: const EdgeInsets.symmetric(horizontal: 24, vertical: 12),  
           ^  
lib/components/post_item.dart:11:11: Error: The argument type 'Widget' can't be assigned to the parameter type 'EdgeInsets'.  
      padding: const EdgeInsets.symmetric(horizontal: 24, vertical: 12),  
           ^  
  
  1 import '.../styles/app_text.dart';  
  2  
  3 class PostItem extends StatelessWidget {  
  4   const PostItem({super.key});  
  5  
  6   @override  
  7   Widget build(BuildContext context) {  
  8     return Padding(  
  9       padding: const EdgeInsets.symmetric(horizontal: 24, vertical: 12),  
10       child: Column(  
11         children: [  
12           Row(  
13             children: [  
14               Image.asset(  
15                 'assets/temp/user1.png',  
16                 width: 40,  
17                 height: 40,  
18               ), // Image.asset  
19               SizedBox(  
20                 width: 16,  
21               ), // SizedBox  
22               Text(  
23                 "Sarah Fernandez",  
24                 style: AppText.subtitle3,  
25               ), // Text  
26             ],  
27           ), // Row  
28           SizedBox(  
29             height: 12,  
30           ), // SizedBox  
31         ],  
32       ), // Column  
33     );  
34   }  
35 }
```

Updated PostItem component to take input parameter

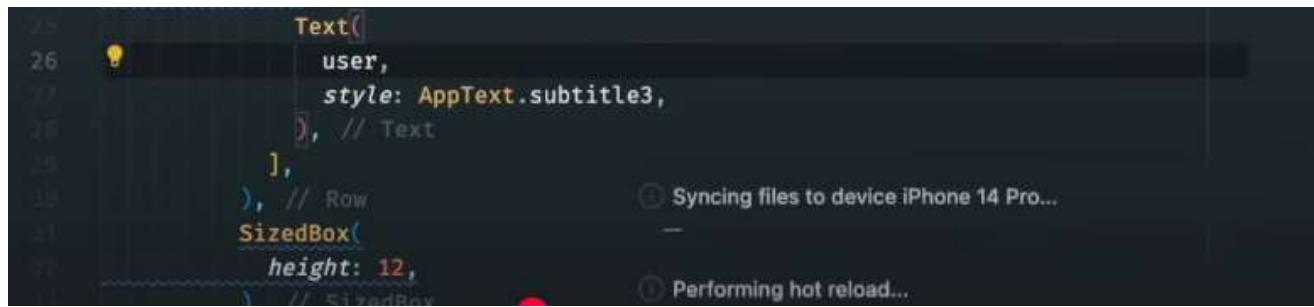


```
import 'package:flutter/material.dart';
import 'package:app_text/styles/app_text.dart';

class PostItem extends StatelessWidget {
    final String user;
    const PostItem({super.key, required this.user});

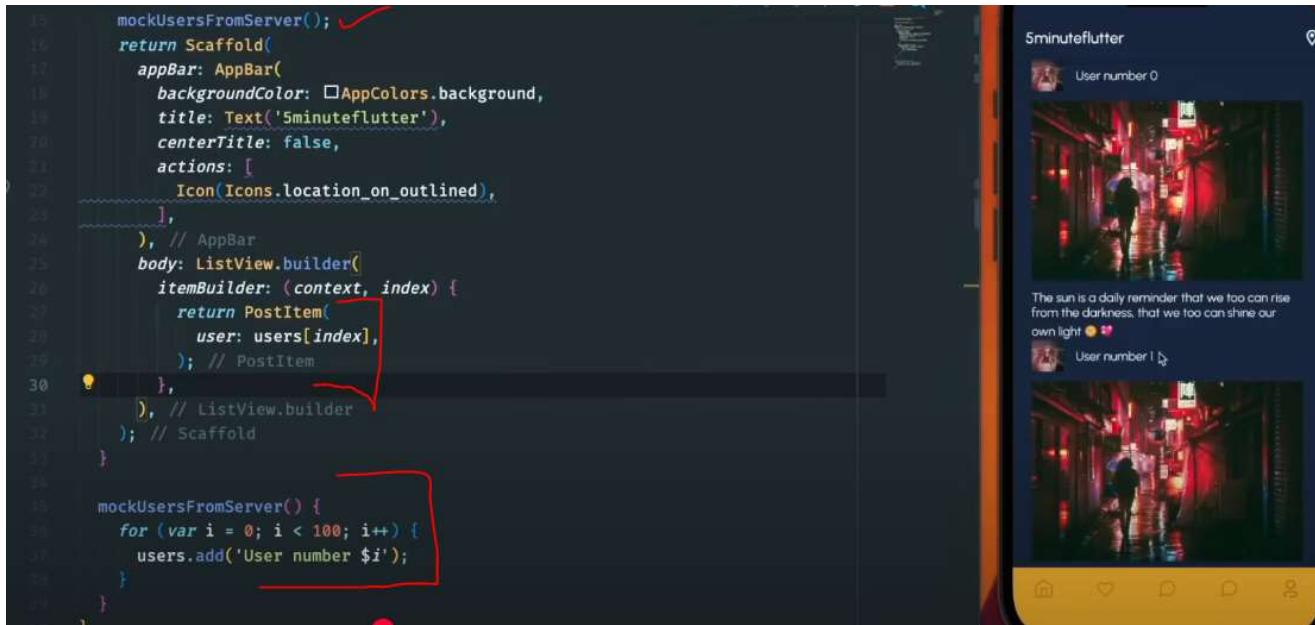
    @override
    Widget build(BuildContext context) {
        return Padding(
            padding: const EdgeInsets.symmetric(horizontal: 24),
            child: Column(
                children: [
                    Row(
                        children: [
                            Image.asset(
                                'assets/temp/user1.png',
                                width: 40,
                                height: 40,
                            ), // Image.asset
                            SizedBox(
                                width: 16,
                            ), // SizedBox
                            Text(
                                "Sarah Fernandez",
                                style: AppText.subtitle3,
                            ),
                        ],
                    ),
                ],
            ),
        );
    }
}
```

Instead of name “Sarah Fernandez” we used input variable “user”

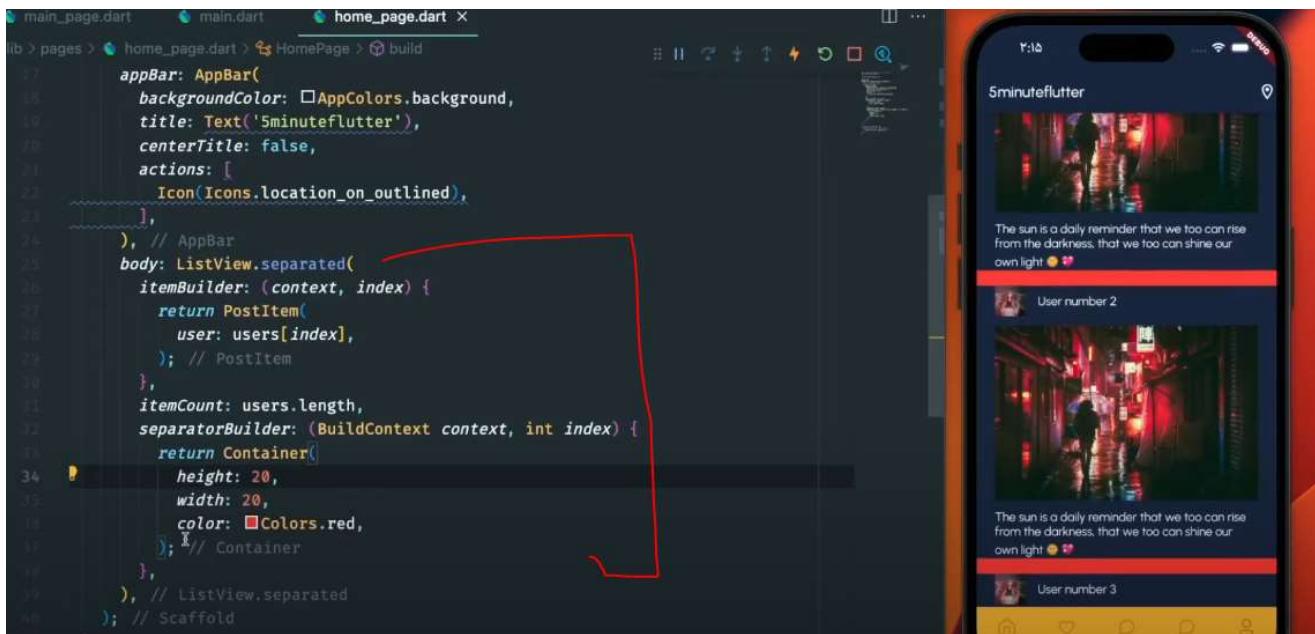


```
25
26     Text(
27         user,
28         style: AppText.subtitle3,
29     ), // Text
30 ],
31 ), // Row
32 SizedBox(
33     height: 12,
34 ), // SizedBox
```

See



To give space between user and the previous post



Removing red color and replacing container with SizedBox

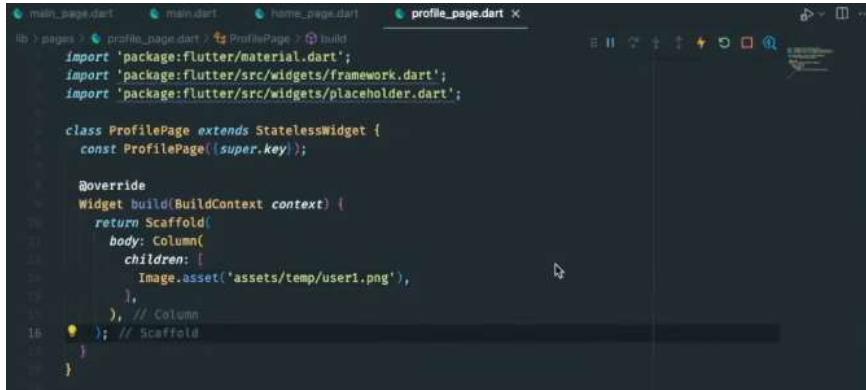


The screenshot shows a Flutter application running on an iPhone X simulator. The app has a dark theme. It displays a feed of two posts. Each post includes a user profile picture, a placeholder for 'User number 0' or 'User number 1', a large image of a person walking in a city at night, and a quote. The quote for the first post is: "The sun is a daily reminder that we too can rise from the darkness, that we too can shine our own light 😊☀️". The quote for the second post is: "The sun is a daily reminder that we too can rise from the darkness, that we too can shine our own light 😊☀️".

```
lib> pages > home_page.dart > HomePage > build
  title: Text('5minuteflutter'),
  centerTitle: false,
  actions: [
    Icon(Icons.location_on_outlined),
  ],
), // AppBar
body: ListView.separated(
  itemBuilder: (context, index) {
    return PostItem(
      user: users[index],
    ); // PostItem
  },
  itemCount: users.length,
  separatorBuilder: (BuildContext context, int index) {
    return SizedBox(
      height: 24,
    );
  },
), // ListView.separated
); // Scaffold
}

mockUsersFromServer() {
  for (var i = 0; i < 100; i++) {
    User user = User();
    user.id = i;
    user.name = 'User number $i';
    user.profileImage =
        'https://randomuser.me/api/portraits/men/$(i % 10).jpg';
    users.add(user);
  }
}
```

Creating profile_page.dart

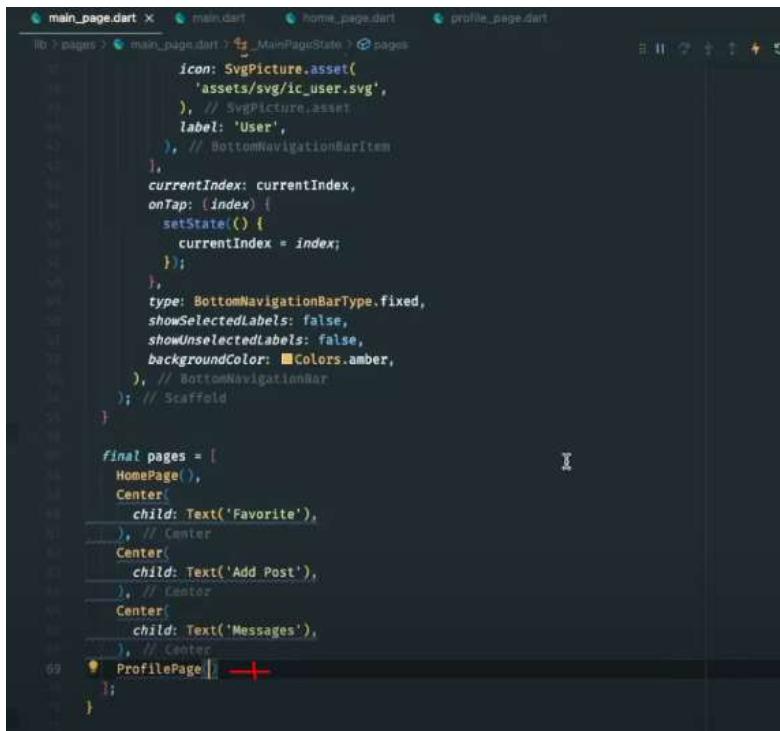


```
lib>pages>profile_page.dart>ProfilePage>lib
import 'package:flutter/material.dart';
import 'package:flutter/src/widgets/framework.dart';
import 'package:flutter/src/widgets/placeholder.dart';

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

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: Column(
        children: [
          Image.asset('assets/temp/user1.png'),
        ],
      ),
    );
  }
}
```

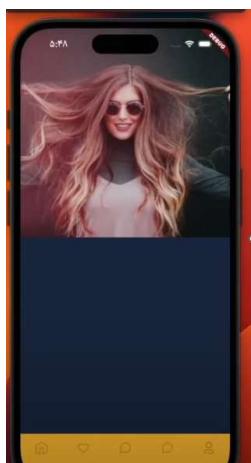
Update bottom sheet for profile screen



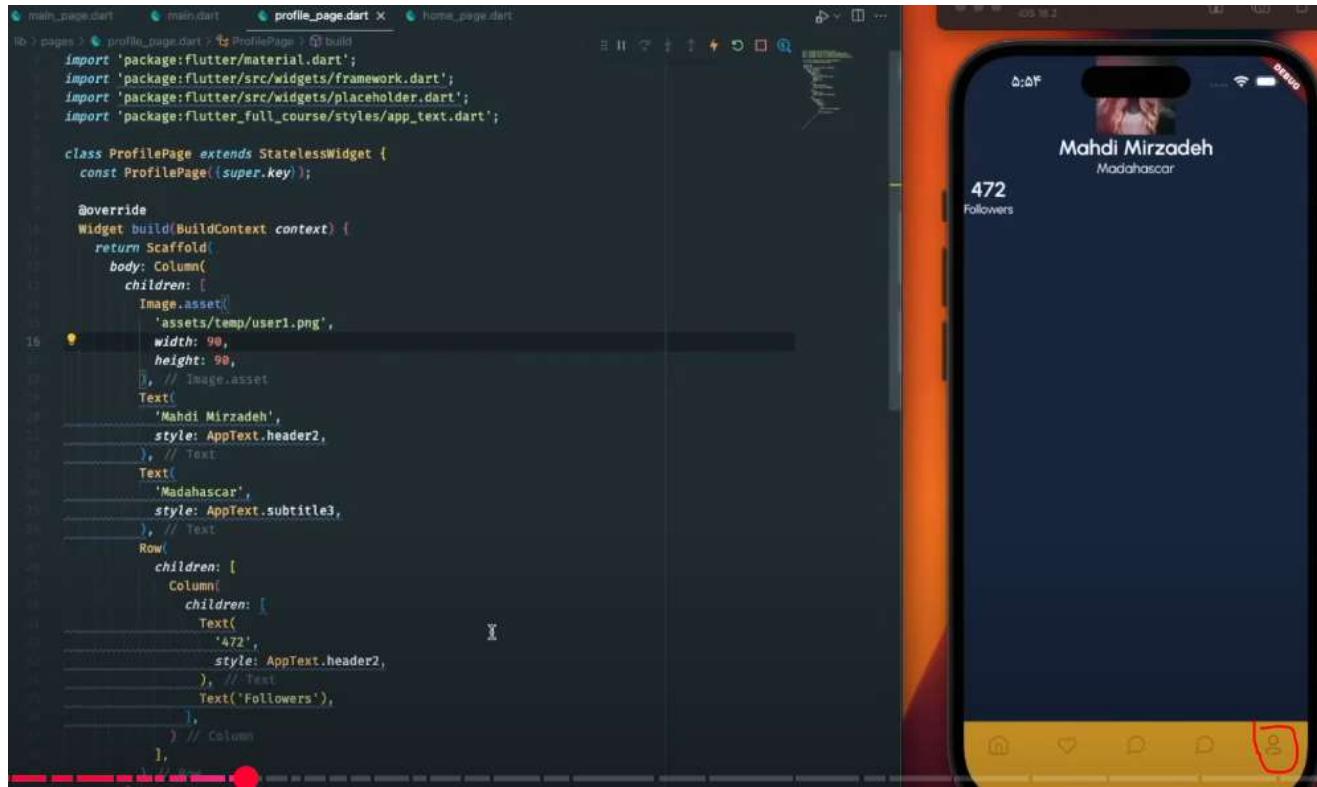
```
lib>pages>main_page.dart>_MainPageState>pages
icon: SvgPicture.asset(
  'assets/svg/ic_user.svg',
), // SvgPicture.asset
label: 'User',
), // BottomNavigationBarItem
),
currentIndex: currentIndex,
onTap: (index) {
  setState(() {
  currentIndex = index;
});
},
type: BottomNavigationBarType.fixed,
showSelectedLabels: false,
showUnselectedLabels: false,
backgroundColor: Colors.amber,
), // BottomNavigationBarItem
); // Scaffold
}

final pages = [
  HomePage(),
  Center(
    child: Text('Favorite'),
  ), // Center
  Center(
    child: Text('Add Post'),
  ), // Center
  Center(
    child: Text('Messages'),
  ), // Center
  ProfilePage(), // ProfilePage
];
}
```

Output

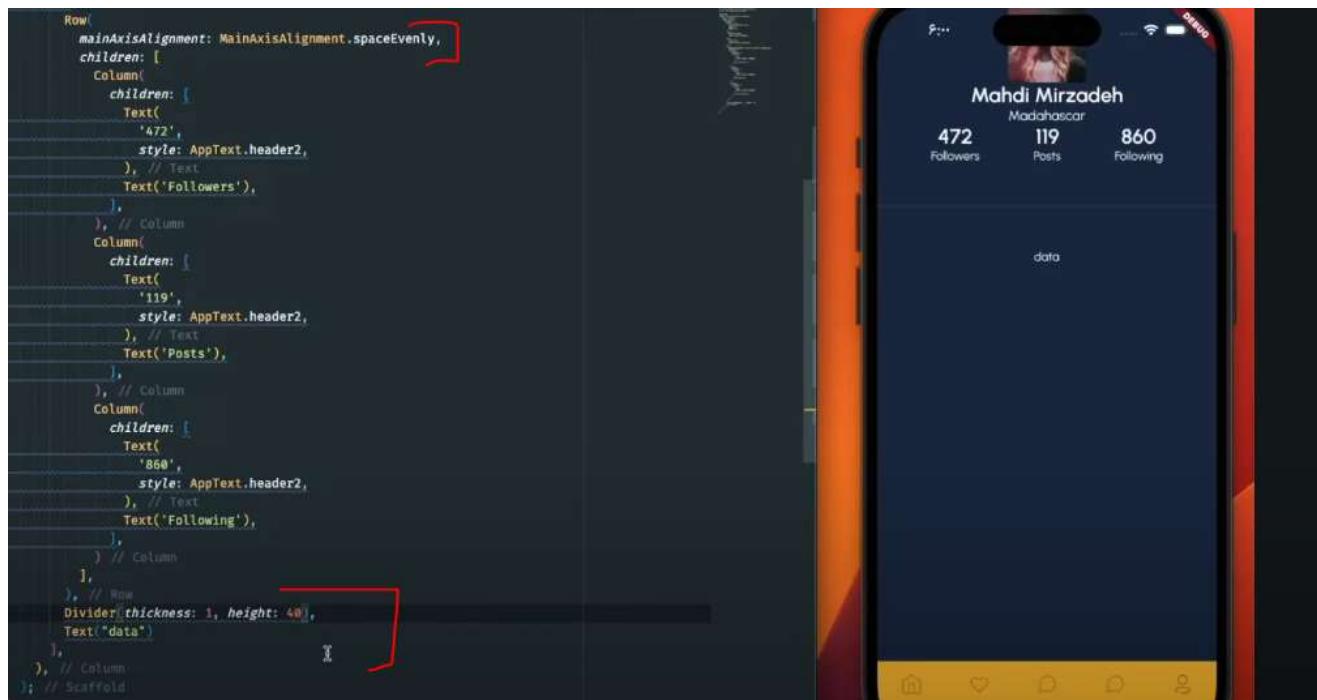


Updating profilePage

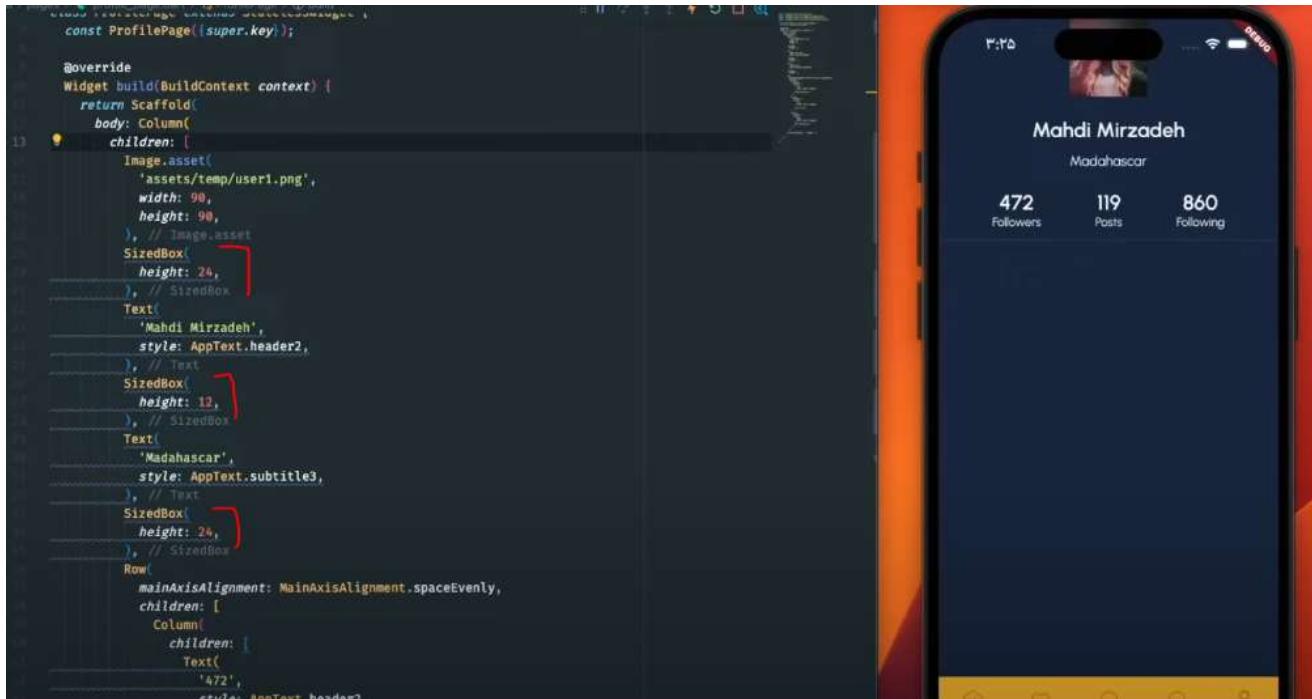


- [Divider\(thickness:int,color:Color.colorName,...\)](#)

Aligning item spaceEvenly



Adding more space between widgets



Creating a file toolbar.dart that will contain functional small components which are fixed such as AppBar

The image shows a Flutter project structure on the left and a code editor on the right. The code editor displays the contents of the toolbar.dart file, which defines a StatelessWidget named Toolbar. This widget takes a title and a list of actions as parameters and returns an AppBar with a background color and the specified title and actions.

```
lib> components> toolbar.dart> Toolbar> build
import 'package:flutter/material.dart';
import 'package:flutter_full_course/styles/app_colors.dart';

class Toolbar extends StatelessWidget {
  final String title;
  final List<Widget>? actions;
  const Toolbar({super.key, required this.title, this.actions});

  @override
  Widget build(BuildContext context) {
    return AppBar(
      backgroundColor: AppColors.background,
      title: Text(title),
      centerTitle: false,
      actions: actions,
    ); // AppBar
  }
}
```

Again updating before we can use it since AppBar accept "PreferredSizeWidget" because Scaffold don't know what size to give to AppBar

```
lib> components> toolbar.dart > Toolbar > preferredSize
import 'package:flutter/material.dart';
import 'package:flutter_full_course/styles/app_colors.dart';

class Toolbar extends StatelessWidget implements PreferredSizeWidget {
  final String title;
  final List<Widget>? actions;
  const Toolbar({super.key, required this.title, this.actions});

  @override
  Widget build(BuildContext context) {
    return AppBar(
      backgroundColor: AppColors.background,
      title: Text(title),
      centerTitle: false,
      actions: actions,
    ); // AppBar
  }

  @override
  Size get preferredSize => const Size.fromHeight(60);
}
```

Using



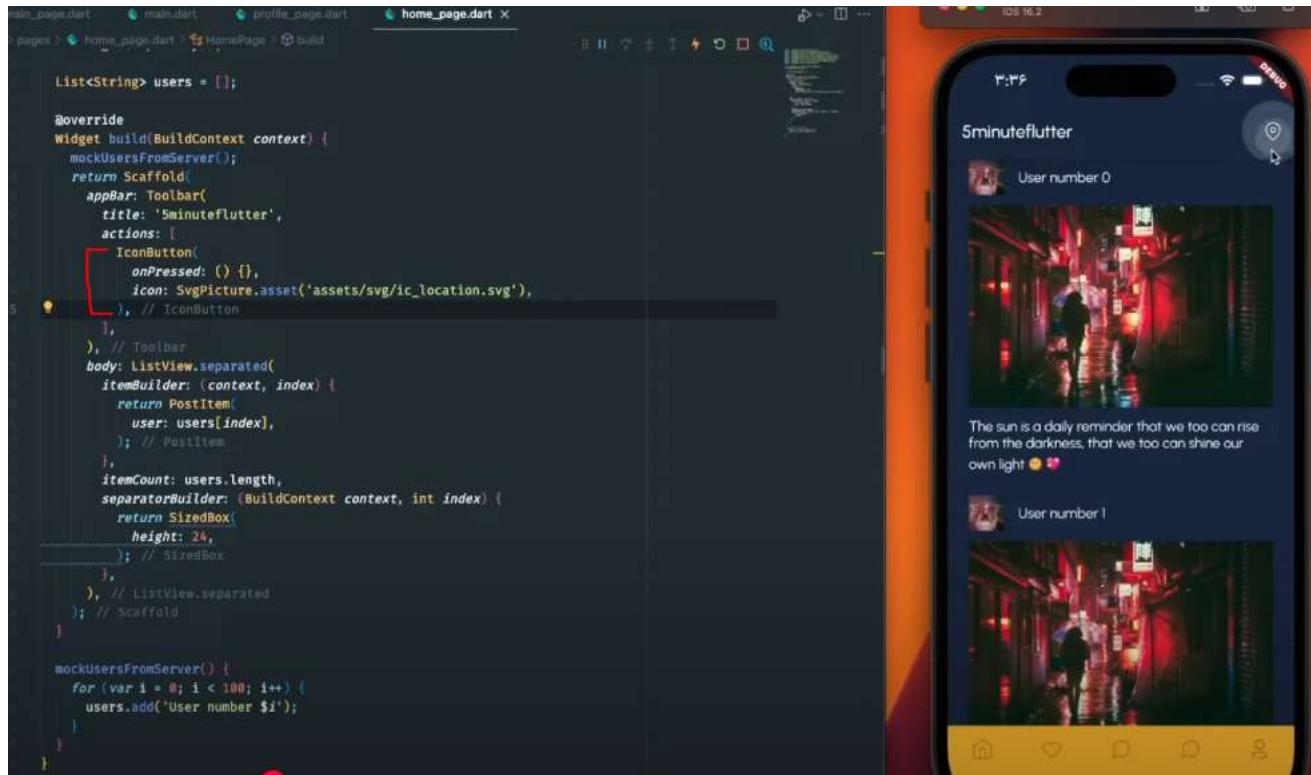
```
lib> home_page.dart > HomePage > Scaffold
class HomePage extends StatelessWidget {
  HomePage({super.key});

  List<String> users = [];

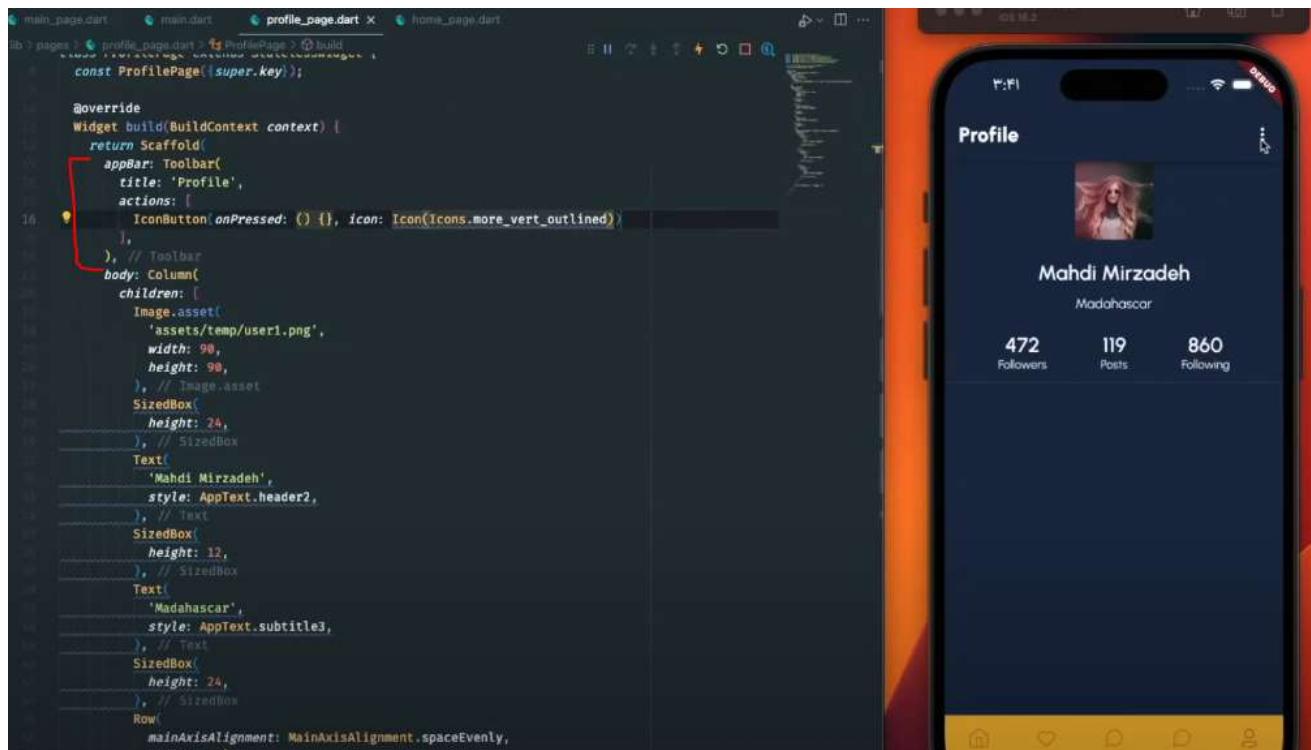
  @override
  Widget build(BuildContext context) {
    mockUsersFromServer();
    return Scaffold(
      appBar: Toolbar(
        title: '5minuteflutter',
        actions: [SvgPicture.asset('assets/svg/ic_location.svg')],
      ), // Toolbar
      body: ListView.separated(
        itemBuilder: (context, index) {
          return PostItem(
            user: users[index],
          ); // PostItem
        },
        itemCount: users.length,
        separatorBuilder: (BuildContext context, int index) {
          return SizedBox(
            height: 24,
          ); // SizedBox
        },
      ), // ListView.separated
    ); // Scaffold
  }
}
```

Replacing static image inside actions with the button called

- `IconButton(onPressed:{},icon:"name")`

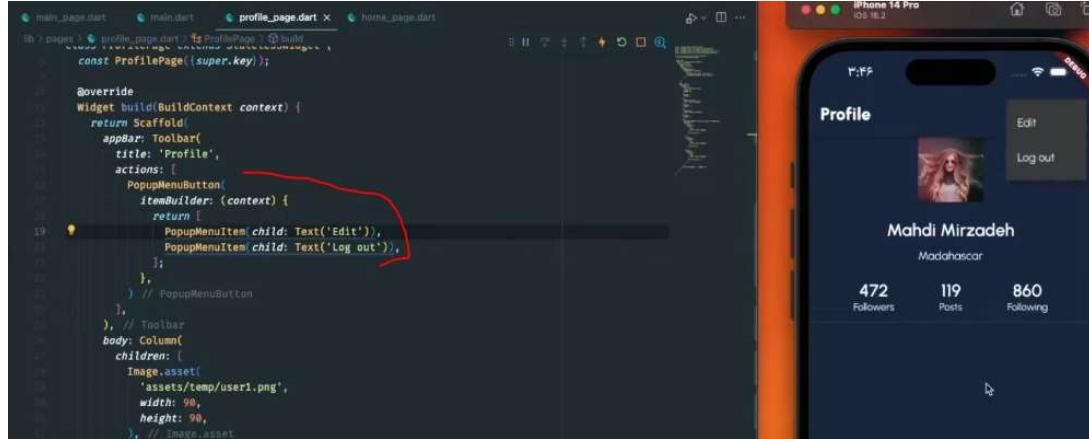


Similarly updating the profile page



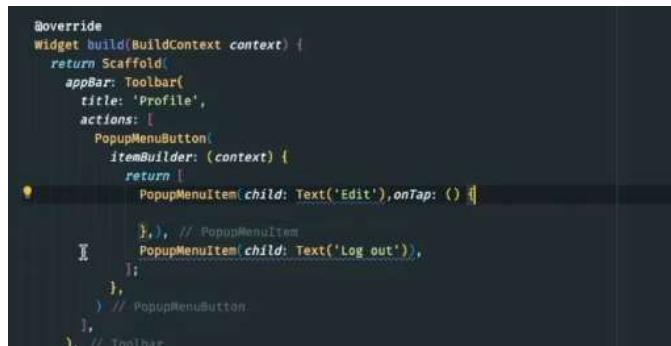
Replacing the IconButton with

- PopupMenuButton() [to show us the list or dropdown]

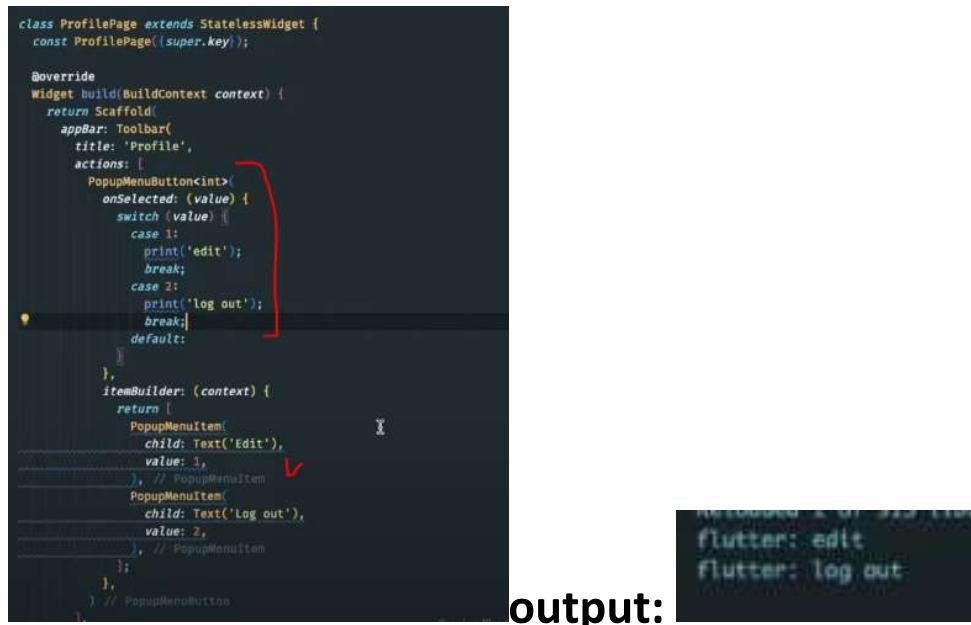


To handle press we have two options

- a) onTap()



- b) value



Or instead of passing int we can use enums

```
import 'package:flutter/material.dart';
import 'package:flutter_full_course/components/toolbar.dart';
import 'package:flutter_full_course/styles/app_text.dart';

enum ProfileMenu{
    edit,
    logout
}

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

    @override
    Widget build(BuildContext context) {
        return Scaffold(
            appBar: Toolbar(
                title: 'Profile',
                actions: [
                    PopupMenuButton<ProfileMenu>(
                        onSelected: (value) {
                            switch (value) {
                                case ProfileMenu.edit:
                                    print('edit');
                                    break;
                                case ProfileMenu.logout:
                                    print('log out');
                                    break;
                                default:
                            }
                        },
                        itemBuilder: (context) {
                            return [
                                PopupMenuItem(
                                    child: Text('Edit'),
                                    value: ProfileMenu.edit,
                                ), // PopupMenuItem
                                PopupMenuItem(
                                    child: Text('Log out'),
                                    value: ProfileMenu.logout,
                                ), // PopupMenuItem
                            ];
                        },
                    ), // PopupMenuButton
                ],
            ),
        );
    }
}
```

- Make 3 dots vertical or horizontal

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

    @override
    Widget build(BuildContext context) {
        return Scaffold(
            appBar: Toolbar(
                title: 'Profile',
                actions: [
                    PopupMenuButton<ProfileMenu>(
                        onSelected: (value) {
                            switch (value) {
                                case ProfileMenu.edit:
                                    print('edit');
                                    break;
                                case ProfileMenu.logout:
                                    print('log out');
                                    break;
                                default:
                            }
                        },
                        icon: const Icon(Icons.more_vert_rounded),
                        itemBuilder: (context) {
                            return [
                                PopupMenuItem(
                                    child: Text('Edit'),
                                    value: ProfileMenu.edit,
                                ), // PopupMenuItem
                                PopupMenuItem(
                                    child: Text('Log out'),
                                    value: ProfileMenu.logout,
                                ), // PopupMenuItem
                            ];
                        },
                    ),
                ],
            ),
        );
    }
}
```

Giving radius to profileImage



```
    icon: const Icon(Icons.more_vert_rounded),
    itemBuilder: (context) {
      return [
        PopupMenuItem(
          child: Text('Edit'),
          value: ProfileMenu.edit,
        ), // PopupMenuItem
        PopupMenuItem(
          child: Text('Log out'),
          value: ProfileMenu.logout,
        ), // PopupMenuItem
      ];
    },
  ); // PopupMenuButton
},
); // Toolbar
body: Column(
  children: [
    Container(
      padding: const EdgeInsets.all(20),
      decoration: BoxDecoration(
        color: Colors.red,
        borderRadius: BorderRadius.only topLeft: Radius.circular(16),
      ),
      child: Image.asset(
        'assets/temp/user1.png',
        width: 90,
        height: 90,
      ), // Image.asset
    ), // Container
  ],
), // Column
); // Scaffold
); // MainWidget
```

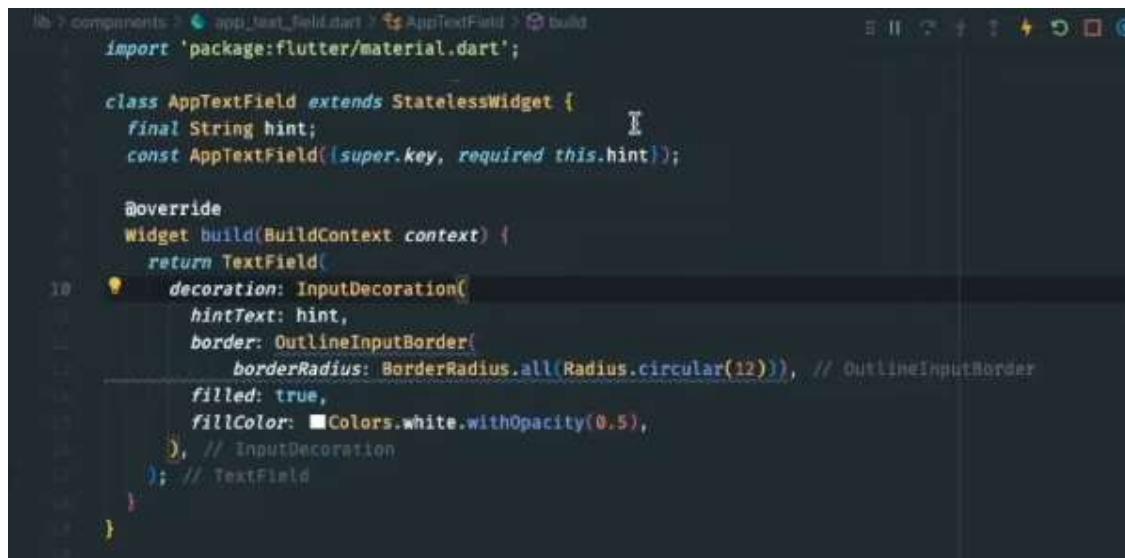
But with Container we are unable to clip the image so using

- ClipRRect()



```
    icon: const Icon(Icons.more_vert_rounded),
    itemBuilder: (context) {
      return [
        PopupMenuItem(
          child: Text('Edit'),
          value: ProfileMenu.edit,
        ), // PopupMenuItem
        PopupMenuItem(
          child: Text('Log out'),
          value: ProfileMenu.logout,
        ), // PopupMenuItem
      ];
    },
  ); // PopupMenuButton
},
); // Toolbar
body: Column(
  children: [
    ClipRRect(
      borderRadius: BorderRadius.all(Radius.circular(16)),
      child: Image.asset(
        'assets/temp/user1.png',
        width: 90,
        height: 90,
      ), // Image.asset
    ), // ClipRRect
    SizedBox(
      height: 24,
```

app_text_field.dart (to reuse it)



```
lib> components > app_text_field.dart > AppTextField > build
import 'package:flutter/material.dart';

class AppTextField extends StatelessWidget {
    final String hint;
    const AppTextField({super.key, required this.hint});

    @override
    Widget build(BuildContext context) {
        return TextField(
            decoration: InputDecoration(
                hintText: hint,
                border: OutlineInputBorder(
                    borderRadius: BorderRadius.all(Radius.circular(12))), // OutlineInputBorder
                filled: true,
                fillColor: Colors.white.withOpacity(0.5),
            ), // InputDecoration
        ); // TextField
    }
}
```

edit_profile.dart



```
lib> pages > edit_profile_page.dart > ...
import 'package:flutter/material.dart';
import 'package:flutter_full_course/components/app_text_field.dart';
import 'package:flutter_full_course/components/toolbar.dart';
|
class EditProfilePage extends StatelessWidget {
    const EditProfilePage({super.key});

    @override
    Widget build(BuildContext context) {
        return Scaffold(
            appBar: Toolbar(title: 'Edit Profile'),
            body: Column(
                children: [
                    AppTextField(hint: 'First name'),
                    AppTextField(hint: 'Last name'),
                    AppTextField(hint: 'Phone number'),
                    AppTextField(hint: 'Location')
                ],
            ), // Column
        ); // Scaffold
    }
}
```

Updating profile page edit button in tab from 3 dot



A screenshot of a Flutter development environment. On the left, the code editor shows the `profile_page.dart` file. A red arrow points to the line of code where `Navigator.of(context).pushNamed('/edit_profile');` is located. On the right, a mobile phone screen displays a profile page for 'Mahdi Mirzadeh'. The profile picture is a woman with long hair. Below it, the name 'Mahdi Mirzadeh' is followed by the handle 'Madahascar'. At the bottom, there are statistics: 472 Followers, 119 Posts, and 860 Following. To the right of the stats is an 'Edit' button.

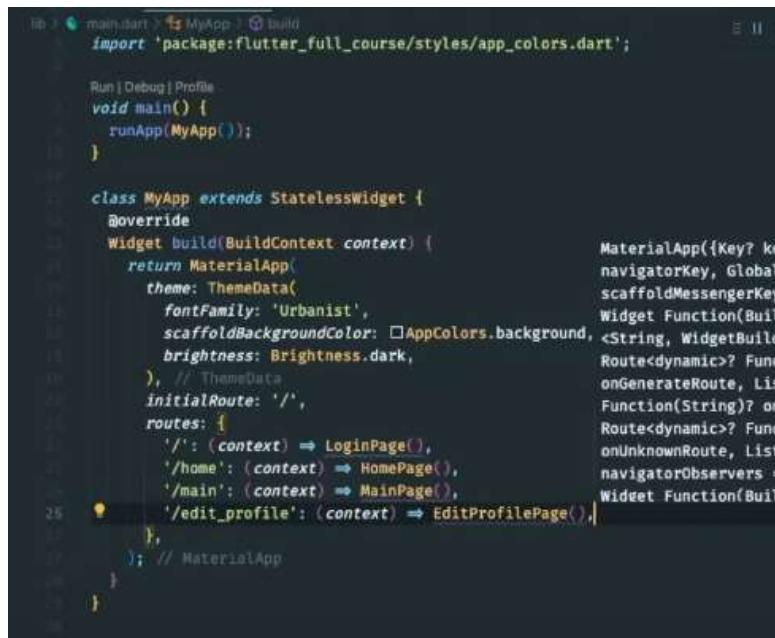
```
lib>pages>profile_page.dart>ProfilePage>@build
import 'package:flutter/material.dart';
import 'package:flutter_full_course/components/toolbar.dart';
import 'package:flutter_full_course/styles/app_text.dart';

enum ProfileMenu { edit, logout }

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

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: Toolbar(
        title: 'Profile',
        actions: [
          PopupMenuButton<ProfileMenu>(
            onSelected: (value) {
              switch (value) {
                case ProfileMenu.edit:
                  Navigator.of(context).pushNamed('/edit_profile');
                  break;
                case ProfileMenu.logout:
                  print('log out');
                  break;
                default:
              }
            },
            icon: const Icon(Icons.more_vert_rounded),
          ),
        ],
      ),
    );
  }
}
```

Updating route in main.dart



A screenshot of the `main.dart` file. A red arrow points to the line of code where `'/edit_profile': (context) => EditProfilePage(),` is added to the `routes:` list. This adds a new route for editing the profile.

```
lib>main.dart>MyApp>build
import 'package:flutter_full_course/styles/app_colors.dart';

Run | Debug | Profile
void main() {
  runApp(MyApp());
}

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      theme: ThemeData(
        fontFamily: 'Urbanist',
        scaffoldBackgroundColor: AppColors.background,
        brightness: Brightness.dark,
      ),
      initialRoute: '/',
      routes: [
        '/': (context) => LoginPage(),
        '/home': (context) => HomePage(),
        '/main': (context) => MainPage(),
        '/edit_profile': (context) => EditProfilePage(),
      ],
    );
  }
}
```

Output



>> AppString.dart for Future multilanguage use

```
lib/config/app_string.dart
class AppStrings {
    static const appName = '5minuteflutter';
    static const helloWelcome = 'Hello, welcome back!';
    static const loginToContinue = 'Login to continue';
    static const username = 'Username';
    static const password = 'Password';
    static const forgotPassword = 'Forgot password?';
    static const orSignInWith = 'Or sign in with';
    static const loginWithGoogle = 'Login with Google';
    static const loginWithFacebook = 'Login with Facebook';
    static const dontHaveAccount = "Don't have account? ";
    static const signup = 'Sign up';
    static const home = 'Home';
    static const favorites = 'Favorites';
    static const add = 'Add';
    static const messages = 'Messages';
    static const user = 'User';
    static const followers = 'Followers';
    static const posts = 'Posts';
    static const following = 'Following';
    static const profile = 'Profile';
    static const editProfile = 'Edit Profile';
    static const firstName = 'First name';
    static const lastName = 'Last name';
    static const phoneNumber = 'Phone number';
    static const location = 'Location';
}
```

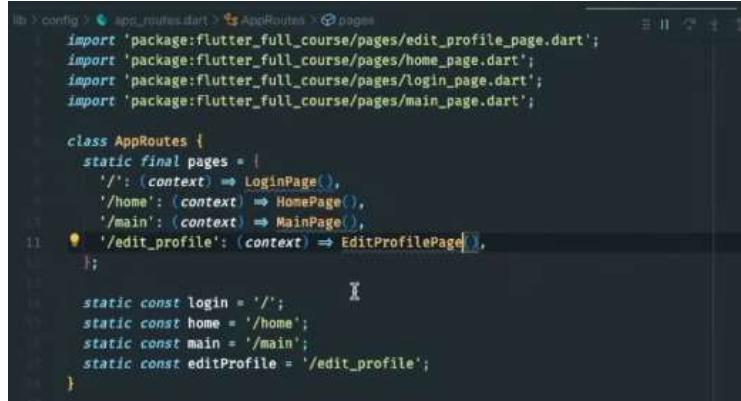
Using

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

    @override
    Widget build(BuildContext context) {
        return Scaffold(
            body: SingleChildScrollView(
                child: SizedBox(
                    height: MediaQuery.of(context).size.height,
                    child: Padding(
                        padding: EdgeInsets.all(24),
                        child: Column(
                            children: [
                                Spacer(),
                                Text(
                                    AppStrings.helloWelcome,
                                    style: TextStyle(
                                        color: Colors.white,
                                        fontSize: 22,
                                        fontWeight: FontWeight.bold),
                                ),
                                Text(
                                    AppStrings.loginToContinue,
                                    style: TextStyle(
                                        color: Colors.white,
                                        fontSize: 16,
                                        fontWeight: FontWeight.normal),
                                ),
                            ],
                        ),
                    ),
                ),
            ),
        );
    }
}
```

>> All routes in one place

to make all our screen navigation variable in one class for better handling



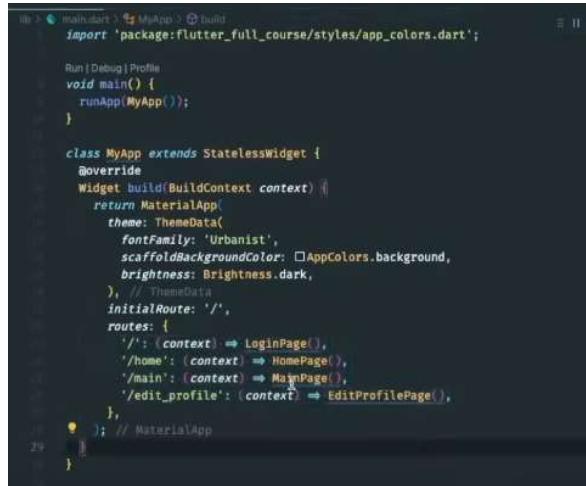
```
lib> config > app_routes.dart > AppRoutes > pages
import 'package:flutter_full_course/pages/edit_profile_page.dart';
import 'package:flutter_full_course/pages/home_page.dart';
import 'package:flutter_full_course/pages/login_page.dart';
import 'package:flutter_full_course/pages/main_page.dart';

class AppRoutes {
  static final pages = {
    '/': (context) => LoginPage(),
    '/home': (context) => HomePage(),
    '/main': (context) => MainPage(),
    '/edit_profile': (context) => EditProfilePage(),
  };

  static const login = '/';
  static const home = '/home';
  static const main = '/main';
  static const editProfile = '/edit_profile';
}
```

Using

Instead of

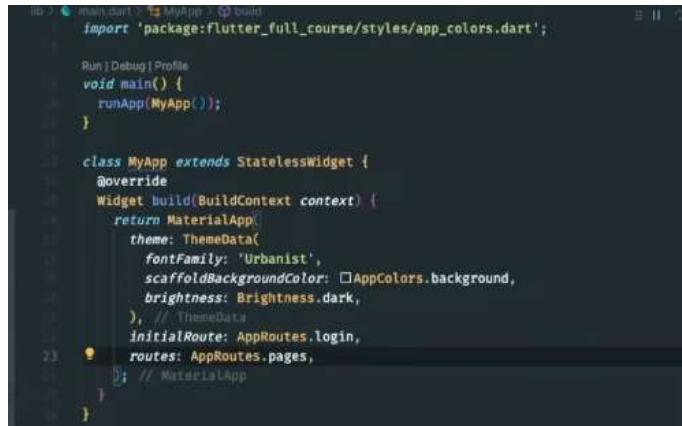


```
lib> main.dart > MyApp > build
import 'package:flutter_full_course/styles/app_colors.dart';

Run | Debug | Profile
void main() {
  runApp(MyApp());
}

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      theme: ThemeData(
        fontFamily: 'Urbanist',
        scaffoldBackgroundColor: AppColors.background,
        brightness: Brightness.dark,
      ),
      initialRoute: '/',
      routes: {
        '/': (context) => LoginPage(),
        '/home': (context) => HomePage(),
        '/main': (context) => MainPage(),
        '/edit_profile': (context) => EditProfilePage(),
      },
    );
  }
}
```

Using



```
lib> main.dart > MyApp > build
import 'package:flutter_full_course/styles/app_colors.dart';

Run | Debug | Profile
void main() {
  runApp(MyApp());
}

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      theme: ThemeData(
        fontFamily: 'Urbanist',
        scaffoldBackgroundColor: AppColors.background,
        brightness: Brightness.dark,
      ),
      initialRoute: AppRoutes.login,
      routes: AppRoutes.pages,
    );
  }
}
```

And change

```
// ... code here
SizedBox(
  height: 48,
  width: double.infinity,
  child: ElevatedButton(
    onPressed: () {
      Navigator.of(context).pushReplacementNamed('/main');
    },
    style: ElevatedButton.styleFrom(
      backgroundColor: Colors.amber,
      foregroundColor: Colors.black,
    ),
    child: Text(AppStrings.login),
  ), // ElevatedButton
), // SizedBox
// ... code here
```

To

```
SizedBox(
  height: 32,
), // SizedBox
SizedBox(
  height: 48,
  width: double.infinity,
  child: ElevatedButton(
    onPressed: () {
      Navigator.of(context).pushReplacementNamed(AppRoutes.main);
    },
    style: ElevatedButton.styleFrom(
      backgroundColor: Colors.amber,
      foregroundColor: Colors.black,
    ),
    child: Text(AppStrings.login),
  ), // ElevatedButton
), // SizedBox
// ... code here
```

>> Similary making class for Icon Strings

```
lib/config/app_icons.dart
```

```
class AppIcons {
  static const _png = 'assets/images/';
  static const _svg = 'assets/svg/';

  static const icFacebook = '${_png}facebook.png';
  static const icGoogle = '${_png}google.png';

  static const icLocation = '${_svg}ic_location.svg';
  static const icHome = '${_svg}ic_home.svg';
  static const icFavorite = '${_svg}ic_favorite.svg';
  static const icAdd = '${_svg}ic_add.svg';
  static const icMessage = '${_svg}ic_messages.svg';
  static const icUser = '${_svg}ic_user.svg';
}
```

Using:From

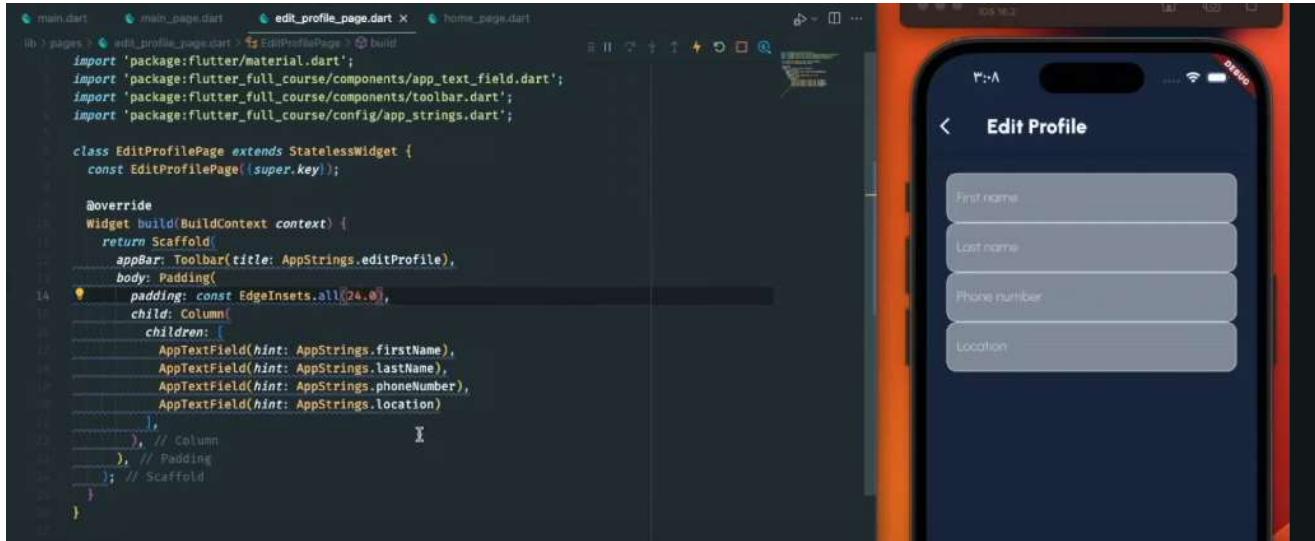
```
child: Row(
  mainAxisAlignment: MainAxisAlignment.center,
  children: [
    Image.asset(
      'assets/images/facebook.png',
      width: 22,
      height: 22,
    ), // Image.asset
    SizedBox(
      width: 8,
    ), // SizedBox
    Text(AppStrings.loginWithFacebook),
  ],
), // Row // ElevatedButton
```

To

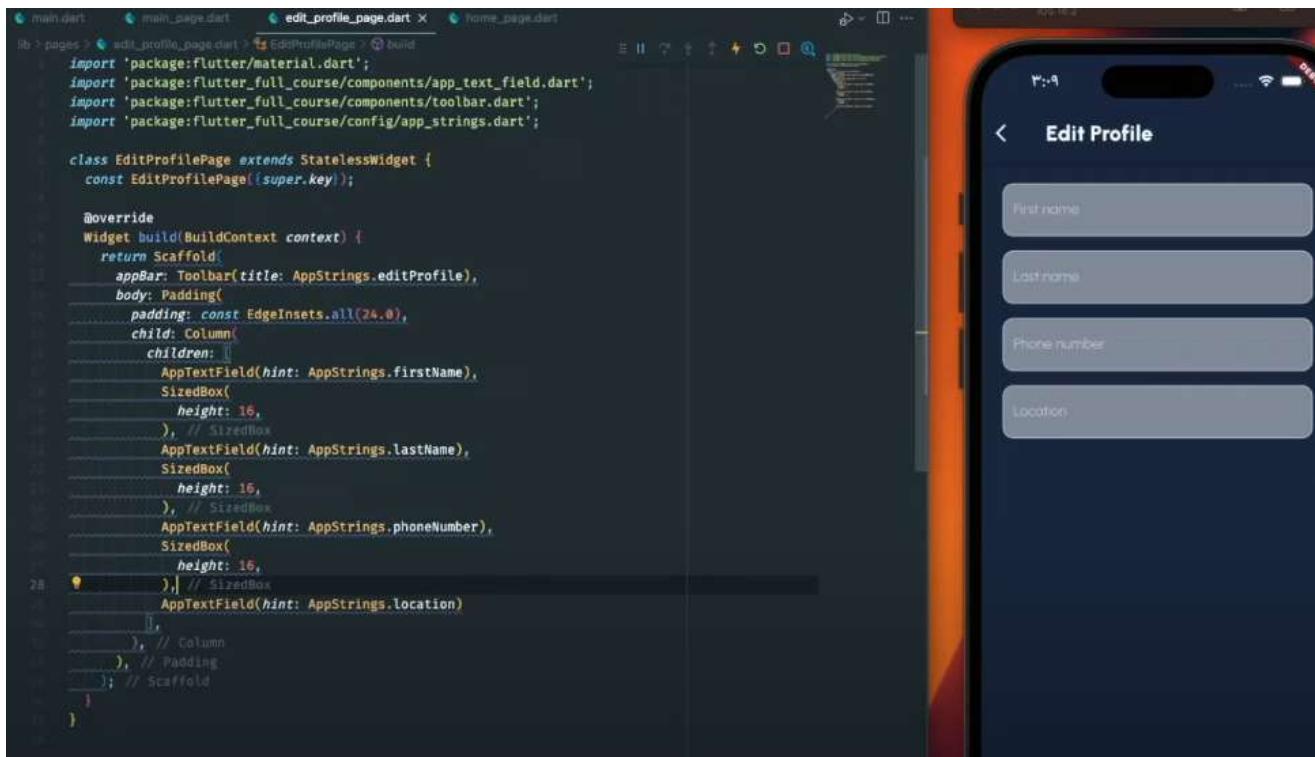
```
child: Row(
  mainAxisAlignment: MainAxisAlignment.center,
  children: [
    Image.asset(
      AppIcons.ic_facebook,
      width: 22,
      height: 22,
    ), // Image.asset
    SizedBox(
      width: 8,
    ), // SizedBox
    Text(AppStrings.loginWithFacebook),
  ],
), // Row // ElevatedButton
```

Coming back to Edit Profile page

Giving padding



Giving space between inputs using SizedBox



>> converting using avatar code to component

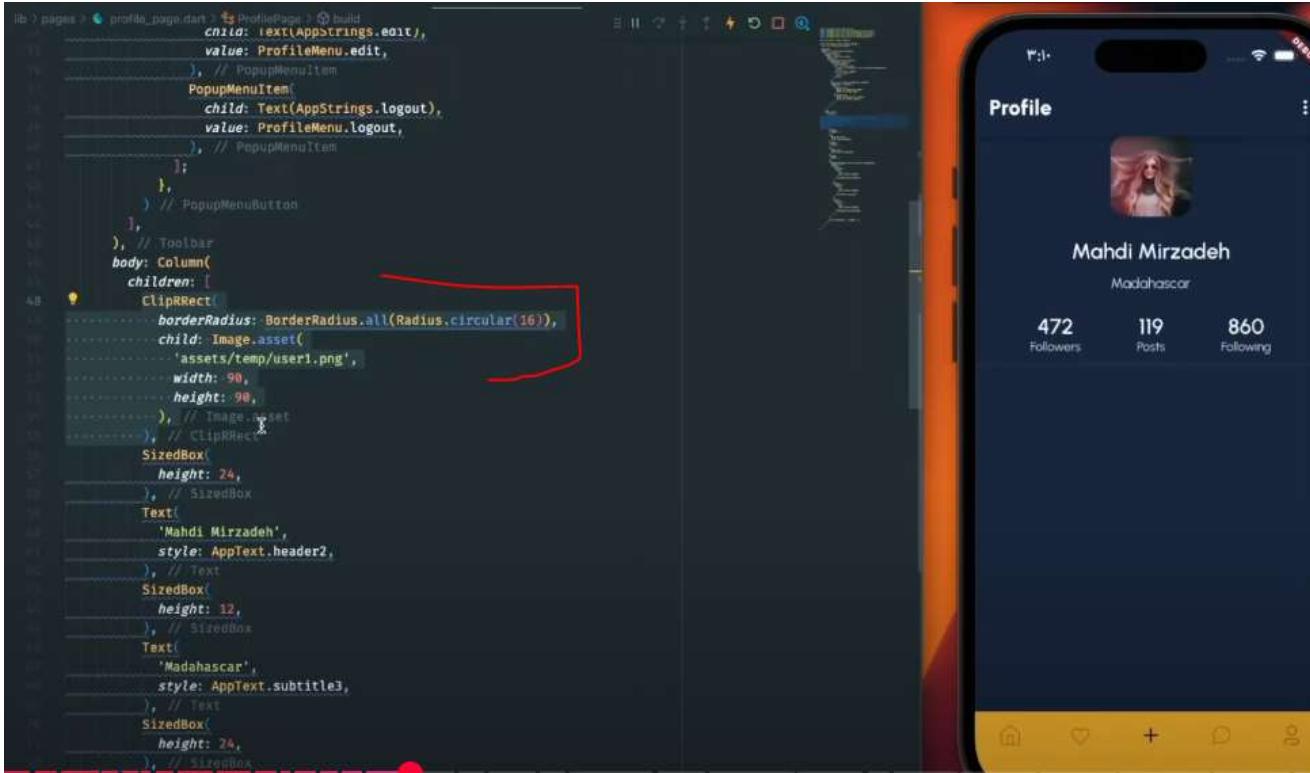
Component

```
lib > components > user_avatar.dart > UserAvatar > build
import 'package:flutter/material.dart';

class UserAvatar extends StatelessWidget {
  final double size;
  const UserAvatar({super.key, this.size = 40});

  @override
  Widget build(BuildContext context) {
    return ClipRRect(
      borderRadius: BorderRadius.all(Radius.circular(16)),
      child: Image.asset(
        'assets/temp/user1.png',
        width: size,
        height: size,
      ), // Image.asset
    ); // ClipRRect
  }
}
```

Using From



TO

The image displays a split-screen view. On the left, a code editor shows the Dart code for a profile page. On the right, a mobile phone screen shows the resulting user interface.

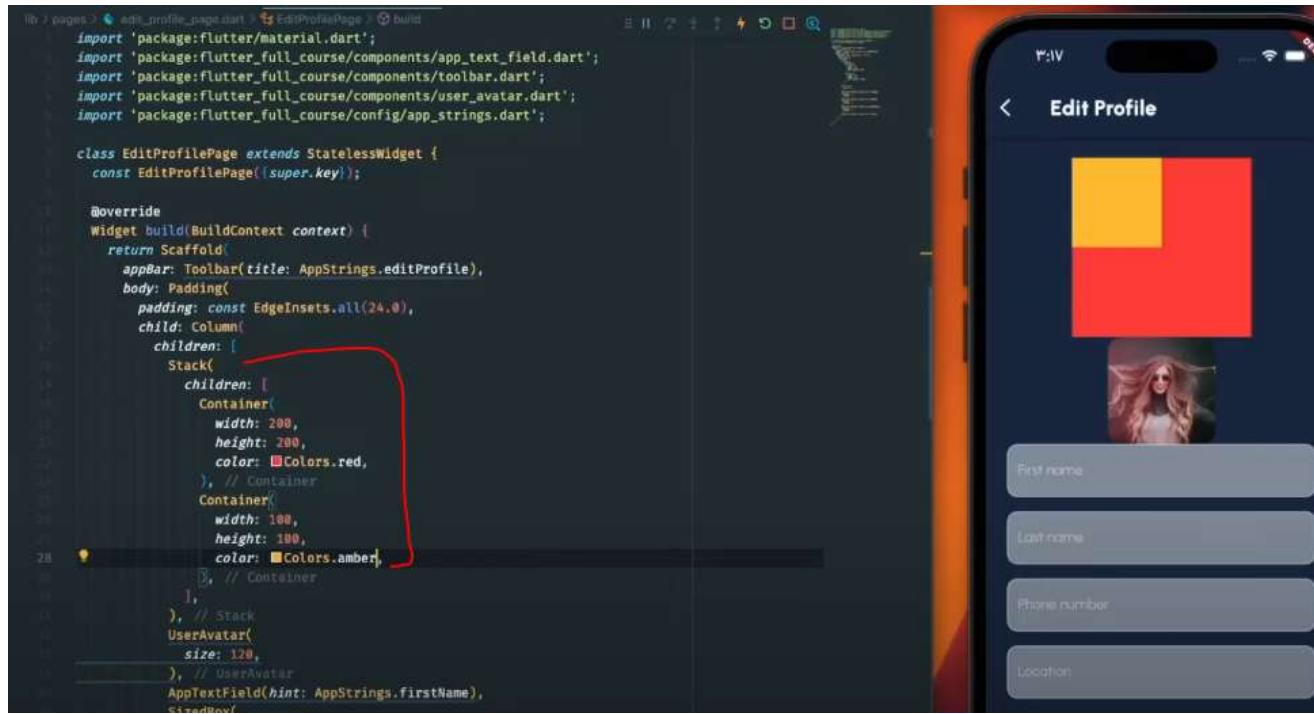
Code (ProfilePage.dart):

```
lib/pages/profile_page.dart
ProfilePage extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('Profile'),
        actions: [
          PopupMenuButton(
            value: ProfileMenuItem.edit,
            child: Text(AppStrings.logout),
            value: ProfileMenuItem.logout,
          ),
        ],
      ),
      body: Column(
        children: [
          UserAvatar(
            size: 96,
          ),
          SizedBox(
            height: 24,
          ),
          Text(
            'Mahdi Mirzadeh',
            style: AppText.header2,
          ),
          SizedBox(
            height: 12,
          ),
          Text(
            'Madagascar',
            style: AppText.body,
          ),
        ],
      ),
    );
  }
}
```

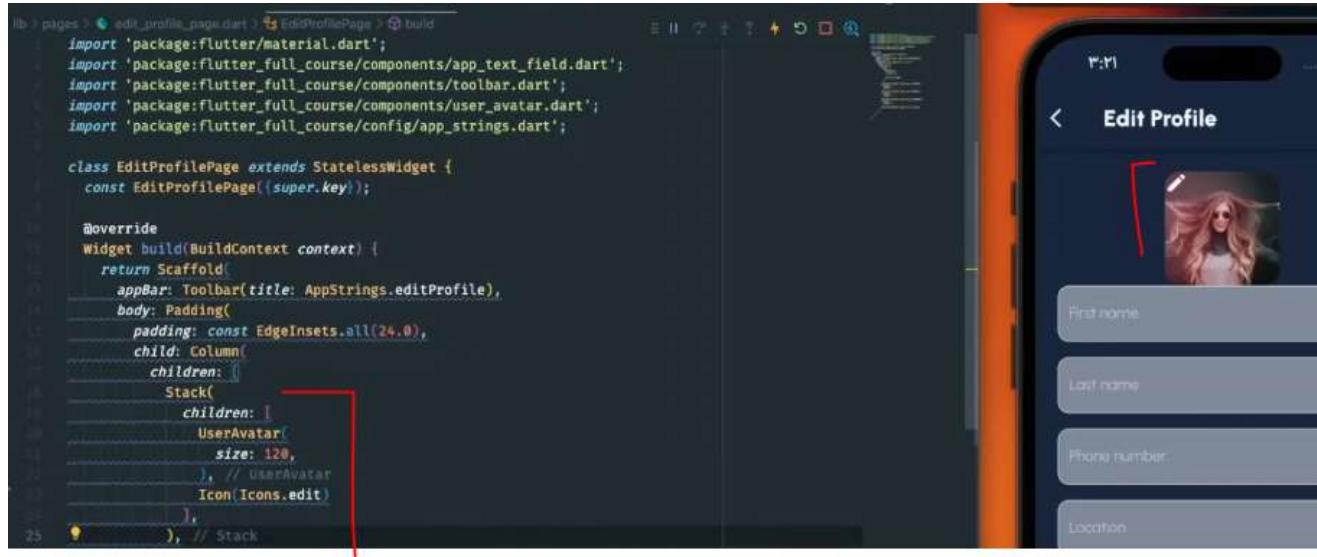
User Interface (iPhone X Mockup):

- Profile Picture:** A placeholder image of a person with long hair.
- Name:** Mahdi Mirzadeh
- Location:** Madagascar
- Statistics:**
 - Followers: 472
 - Posts: 119
 - Following: 860

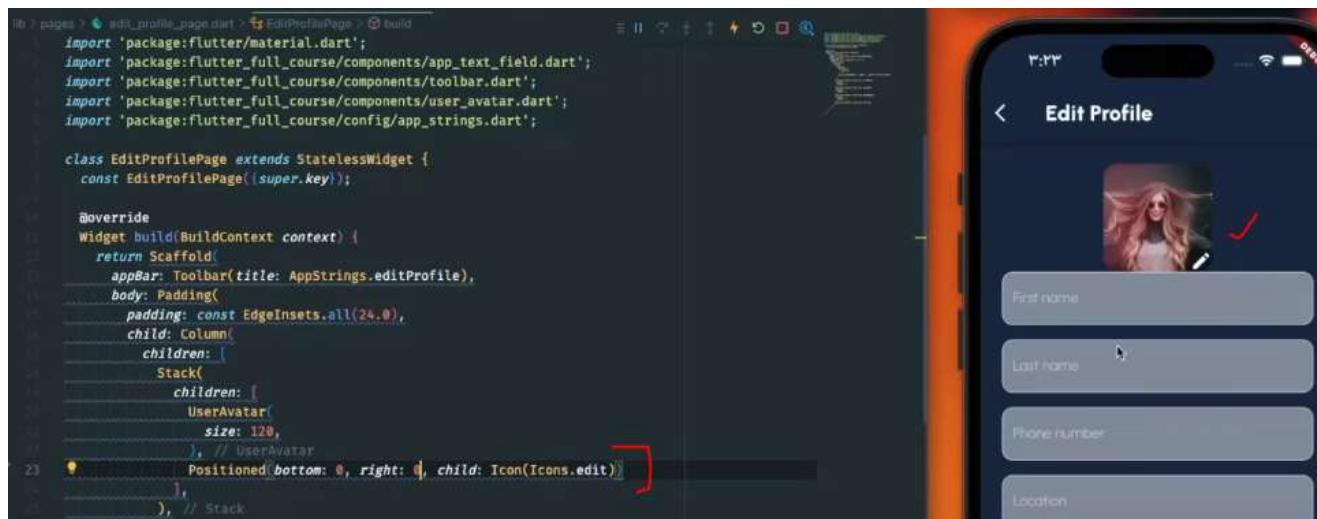
- **Stack()** [to position widget over each other instead of below or side to each other]



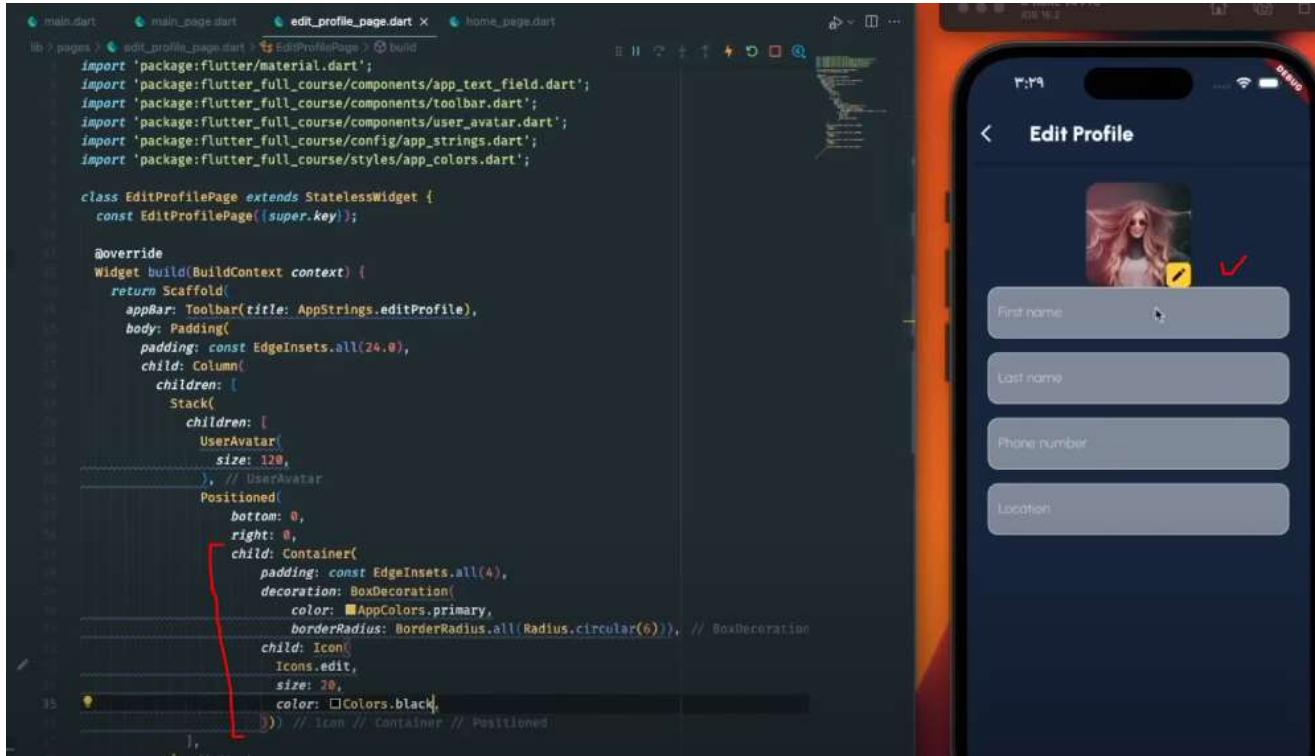
>Adding Pencil over profile avatar using Stack()



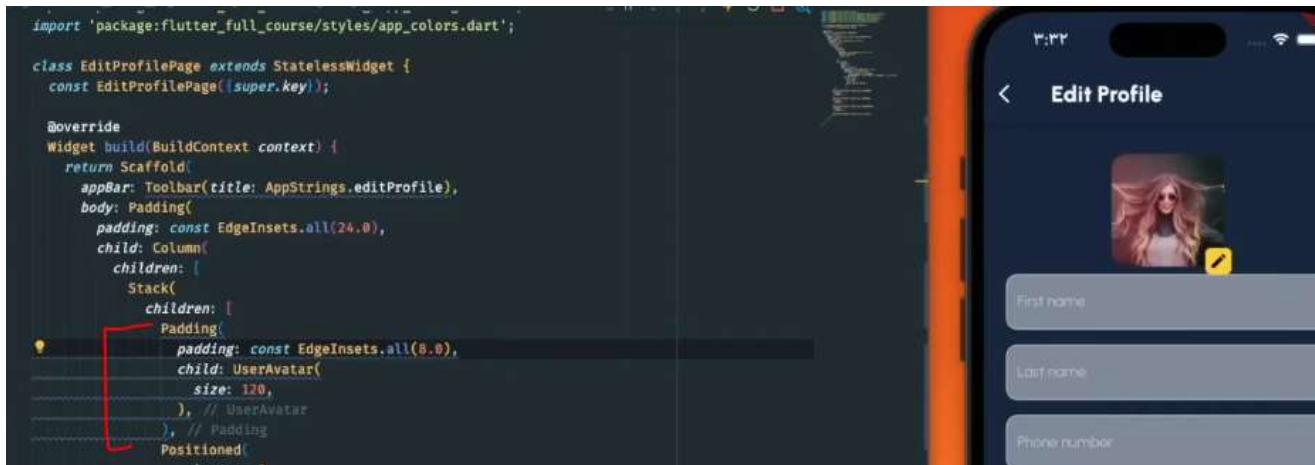
- Positioned(**top:10, left:30, right:x, bottom:y**)
- To center do **left:0 and right:0, top:0, bottom:0**



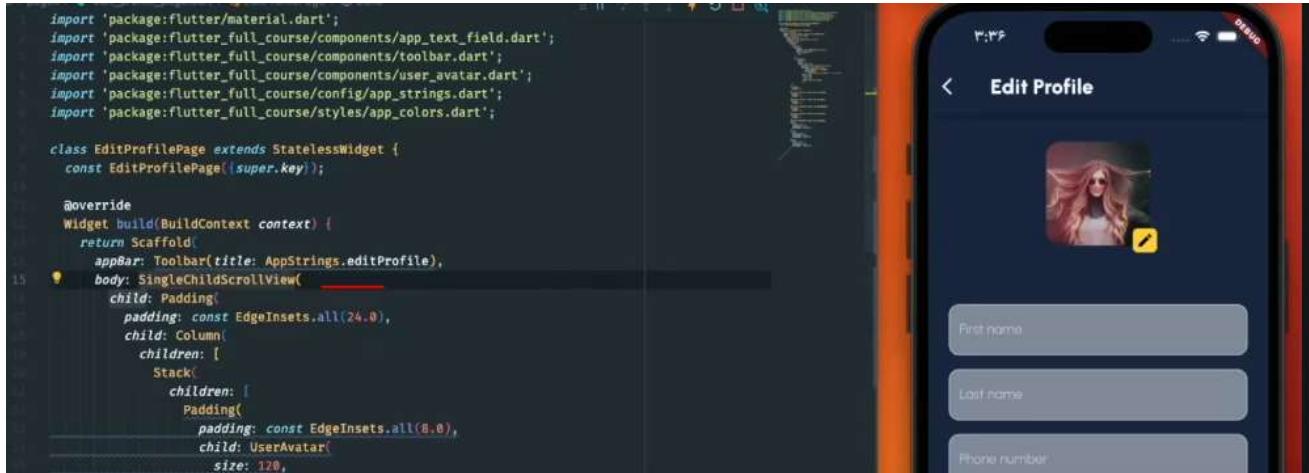
Giving padding color and color to pencil



Giving padding to avatar to show floating pencil effect



To avoid overflow of pixels using: SingleChildScrollView



- Radio buttons
 - a) Defining enums

```
import 'package:flutter_full_course/components/app_text_field.dart';
import 'package:flutter_full_course/components/toolbar.dart';
import 'package:flutter_full_course/components/user_avatar.dart';
import 'package:flutter_full_course/config/app_strings.dart';
import 'package:flutter_full_course/styles/app_colors.dart';

enum Gender {none, male, female, other}

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

  var gender = Gender.none;
```

Remove const

```
lib/pages/edit_profile_page.dart:11:11: Error: Constant expression expected.
  var gender = Gender.none;
          ^
```

```
import 'package:flutter_full_course/components/app_text_field.dart';
import 'package:flutter_full_course/components/toolbar.dart';
import 'package:flutter_full_course/components/user_avatar.dart';
import 'package:flutter_full_course/config/app_strings.dart';
import 'package:flutter_full_course/styles/app_colors.dart';

enum Gender {none, male, female, other}

class EditProfilePage extends StatelessWidget {
  EditProfilePage({super.key});

  var gender = Gender.none;
```

Convert to stateful widget

```
lib/pages/edit_profile_page.dart
```

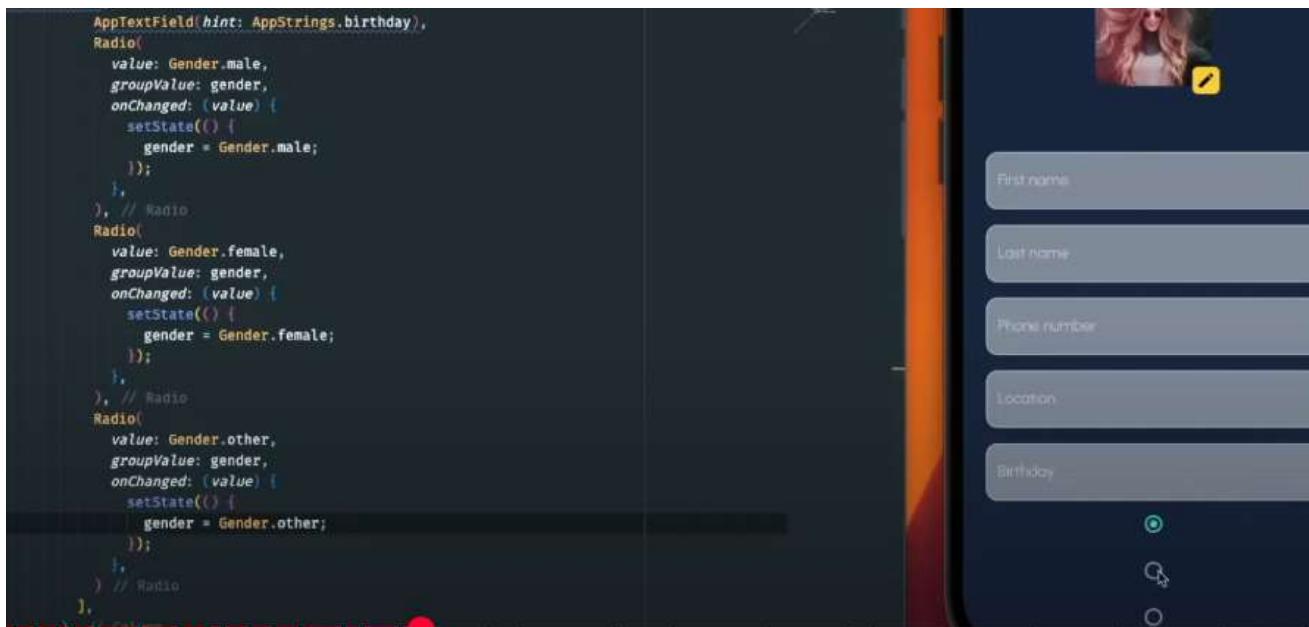
```
import 'package:flutter_full_course/components/toolbar.dart';
import 'package:flutter_full_course/components/user_avatar.dart';
import 'package:flutter_full_course/config/app_strings.dart';
import 'package:flutter_full_course/styles/app_colors.dart';

enum Gender { none, male, female, other }

10 class EditProfilePage extends StatefulWidget {
  EditProfilePage({super.key});

  @override
  State<EditProfilePage> createState() // UserAvatar => _ // PaddingE // BoxDecorationditPDr /;
}

class _EditProfilePageState extends State<EditProfilePage> {
  var gender = Gender.none;
```



Adding title

Method 1: using radioTile

```
AppTextField(hint: AppStrings.birthday),
RadioListTile(
  title: Text('Male'),
  value: Gender.male,
  groupValue: gender,
  onChanged: (value) {
    setState(() {
      gender = Gender.male;
    });
  },
), // RadioListTile
```

Output: will take full row



Method 2: using Row

```
Row(  
  children: [  
    Radio(  
      value: Gender.female,  
      groupValue: gender,  
      onChanged: (value) {  
        setState(() {  
          gender = Gender.female;  
        });  
      },  
    ), // Radio  
    Text('Female')  
)
```

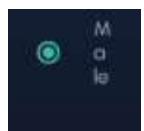
Output: will take particular space



Method 3: giving size to radiotile

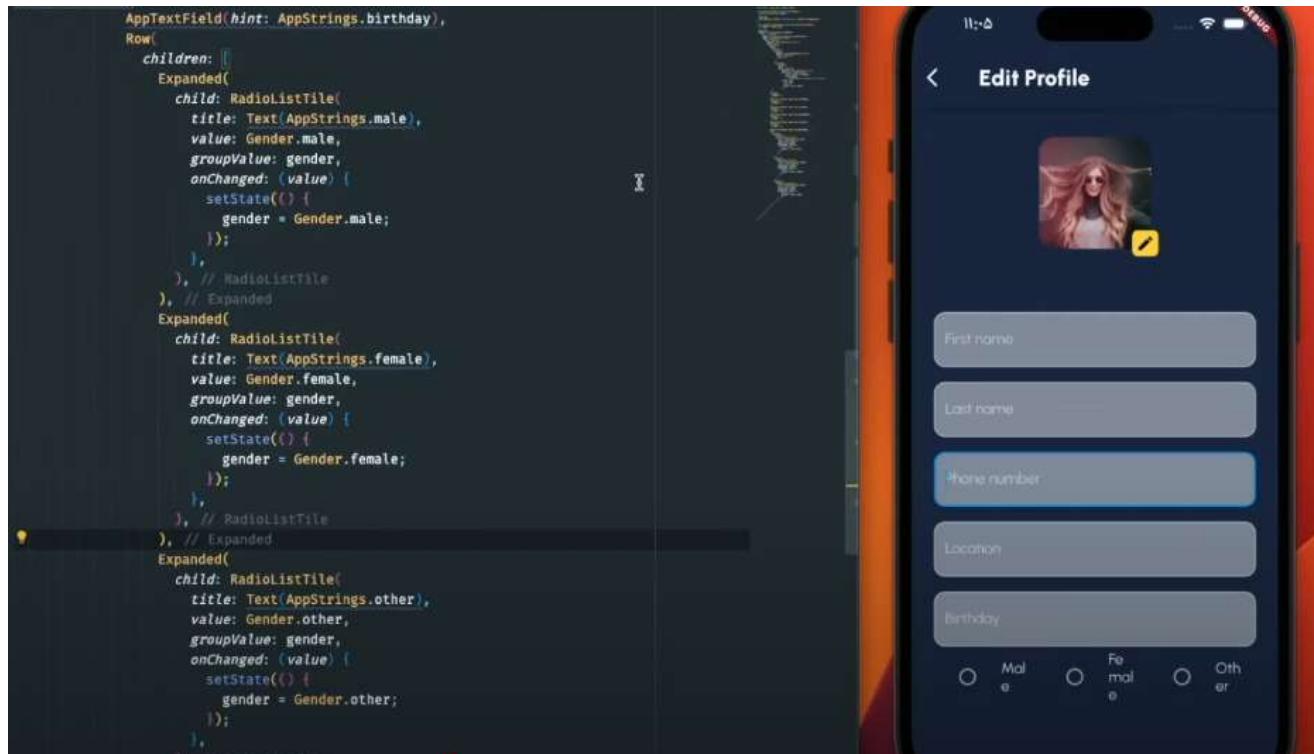
```
SizedBox(  
  width: 100,  
  child: RadioListTile(  
    title: Text(AppStrings.other),  
    value: Gender.other,  
    groupValue: gender,  
    onChanged: (value) {  
      setState(() {  
        gender = Gender.other;  
      });  
    },  
  ), // RadioListTile  
) // SizedBox
```

Output

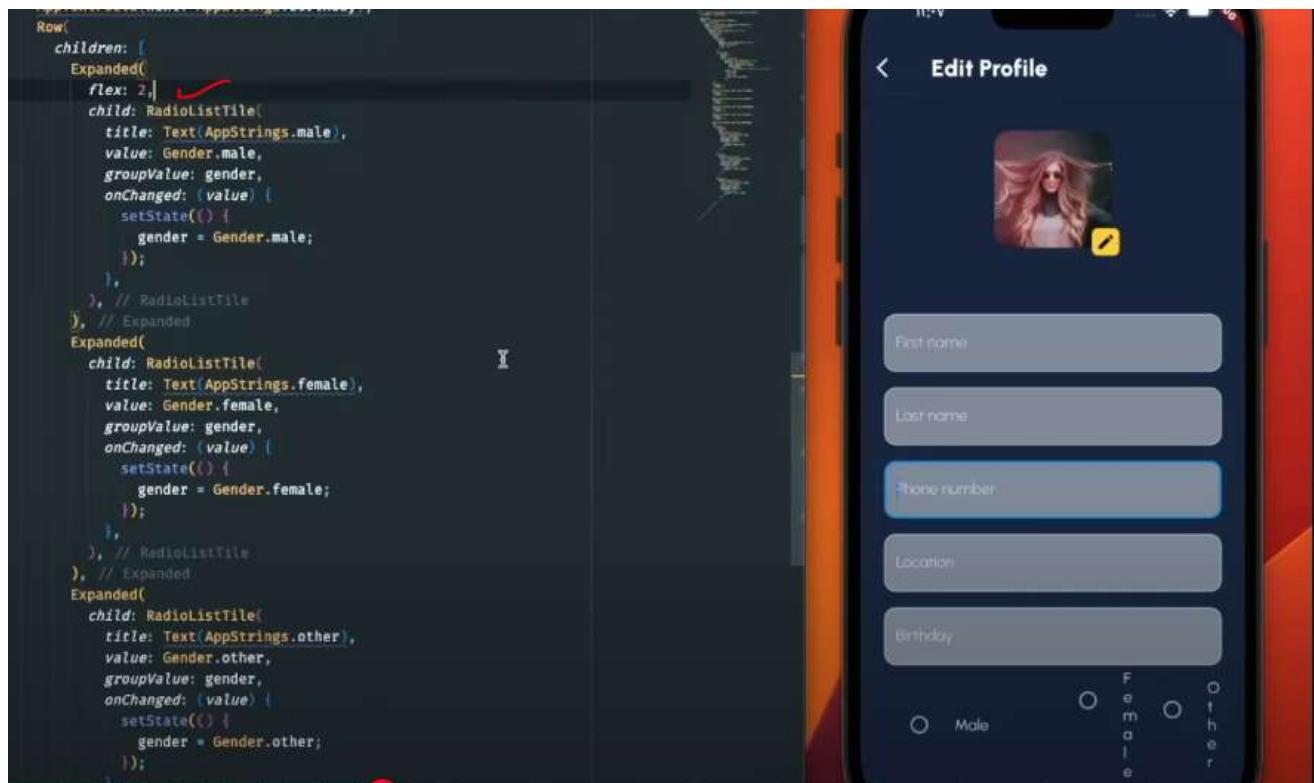


Method 4: Using Expanded()

- **Expanded() [divides space equally w.r.t parent widget]**



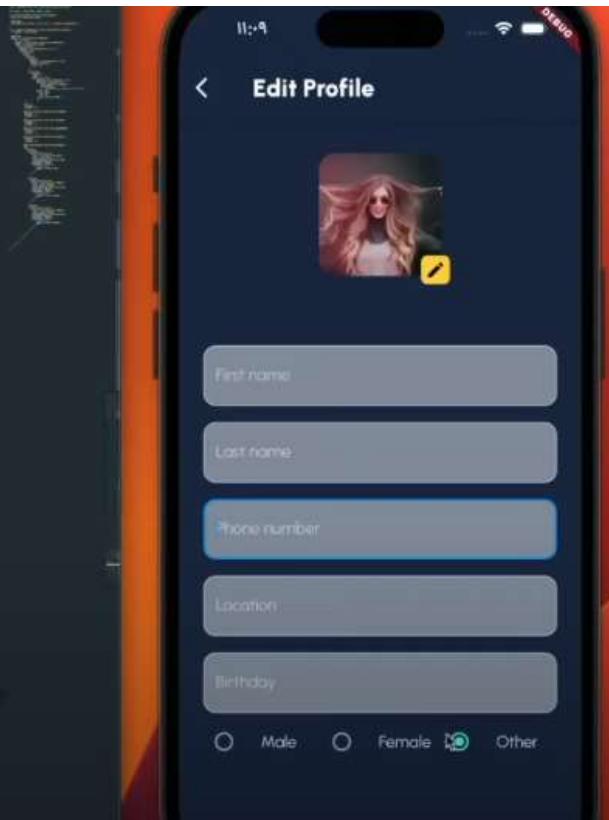
- **Using flex:**



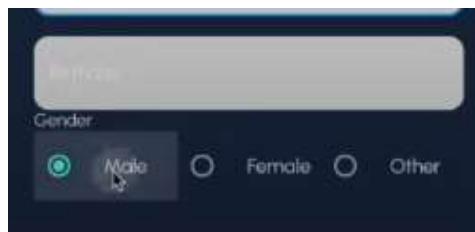
>> removing contentPadding in radioTile to adjust all items in row

```
        child: RadioListTile(
            title: Text(AppStrings.male),
            value: Gender.male,
            contentPadding: EdgeInsets.zero,
            groupValue: gender,
            onChanged: (value) {
                setState(() {
                    gender = Gender.male;
                });
            },
        ), // RadioListTile
    ), // Expanded
    Expanded(
        child: RadioListTile(
            title: Text(AppStrings.female),
            value: Gender.female,
            contentPadding: EdgeInsets.zero,
            groupValue: gender,
            onChanged: (value) {
                setState(() {
                    gender = Gender.female;
                });
            },
        ), // RadioListTile
    ), // Expanded
    Expanded(
        child: RadioListTile(
            title: Text(AppStrings.other),
            value: Gender.other,
            contentPadding: EdgeInsets.zero,
            groupValue: gender,
            onChanged: (value) {
                setState(() {
                    gender = Gender.other;
                });
            },
        ), // RadioListTile
    ), // Expanded
); // Column

```



>>removing extra padding around the button area

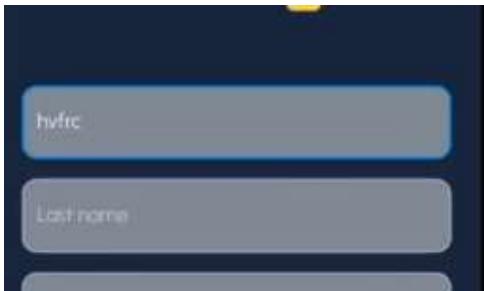


- **visualDensity**

```
Row(
  children: [
    Expanded(
      child: RadioListTile<
        title: Text(AppStrings.male),
        value: Gender.male,
        visualDensity: const VisualDensity(
          horizontal: VisualDensity.minimumDensity,
          vertical: VisualDensity.minimumDensity), // VisualDensity
        contentPadding: EdgeInsets.zero,
        groupValue: gender,
        onChanged: (value) {

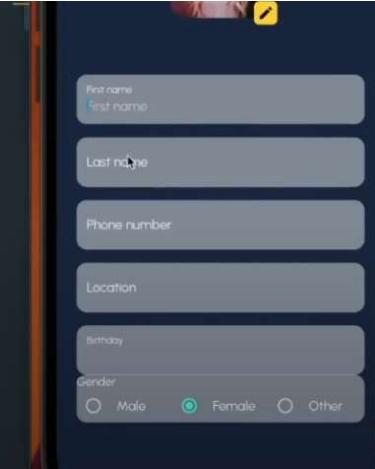
```

>> Showing label/hint by reducing its size to top



- labelText and labelStyle

```
@override
Widget build(BuildContext context) {
  return TextField(
    decoration: InputDecoration(
      hintText: hint,
      labelText: hint,
      labelStyle: TextStyle(
        color: Colors.white,
      ), // TextStyle
      border: const UnderlineInputBorder(
        borderRadius: BorderRadius.all(Radius.circular(12)), // UnderlineInputBorder
        focusedBorder: const UnderlineInputBorder(
          borderSide: BorderSide(color: Colors.transparent),
          borderRadius: BorderRadius.all(Radius.circular(12))), // UnderlineInputBorder
        filled: true,
        fillColor: AppColors.fieldColor,
      ), // InputDecoration
    ); // TextField
}
```



➤ As we know we have a file for styles

```
> styles > app_text.dart > AppText > body1
  );
  static const subtitle1 = TextStyle(
    fontSize: 16,
    fontWeight: FontWeight.w600,
  );
  static const subtitle2 = TextStyle(
    fontSize: 16,
    fontWeight: FontWeight.w500,
  );
  static const subtitle3 = TextStyle(
    fontSize: 16,
    fontWeight: FontWeight.w400,
  );
  static const body1 = TextStyle(
    fontSize: 14,
    fontWeight: FontWeight.w500,
  );
  static const body2 = TextStyle(
    fontSize: 14,
    fontWeight: FontWeight.w400,
  );
}
```

if u want to use all styles of body1 and want to override or add more style use:

- **copyWith(fontSize:4)**

```
children: [
  Text(
    AppStrings.gender,
    style: AppText.body1.copyWith(
      fontSize: 12,
    ),
  ), // Text
  Row(
```

➤ Custom Bottom Navigation



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

  @override
  State<MyBottomNavigation> createState() => _MyBottomNavigationState();
}

class _MyBottomNavigationState extends State<MyBottomNavigation> {
  @override
  Widget build(BuildContext context) {
    return Container(
      height: 87,
      margin: EdgeInsets.all(24),
      child: Stack(
        children: [
          Container(
            decoration: BoxDecoration(
              color: Colors.white,
              borderRadius: BorderRadius.circular(25)),
          ),
          Container(
            child: Stack(
              children: [
                Container(
                  height: 100,
                  width: 100,
                  decoration: BoxDecoration(
                    color: Colors.white,
                    borderRadius: BorderRadius.circular(50),
                    border: Border.all(color: Colors.black, width: 2),
                  ),
                ),
                Container(
                  height: 100,
                  width: 100,
                  decoration: BoxDecoration(
                    color: Colors.white,
                    borderRadius: BorderRadius.circular(50),
                    border: Border.all(color: Colors.black, width: 2),
                  ),
                ),
              ],
            ),
          ),
        ],
      ),
    );
  }
}
```

Applying



```
lib > pages > main_page.dart > _MainPageState
import 'package:flutter_full_course/config/app_strings.dart';
import 'package:flutter_full_course/pages/home_page.dart';
import 'package:flutter_full_course/pages/profile_page.dart';
import 'package:flutter_svg/svg.dart';

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

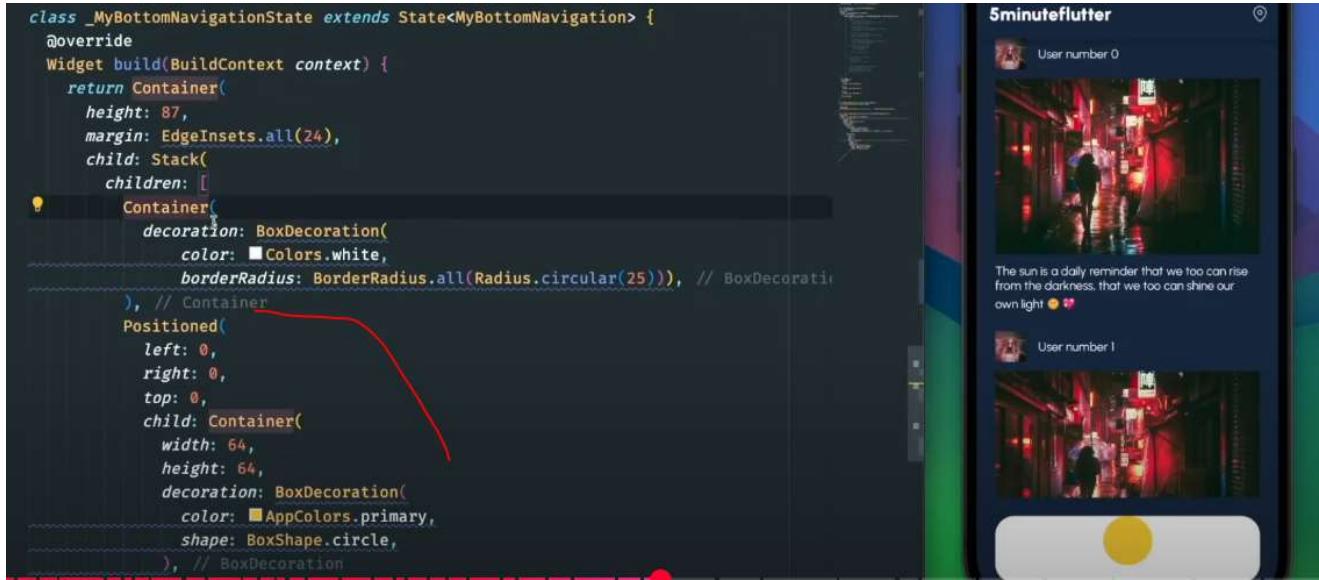
  @override
  State<MainPage> createState() => _MainPageState();
}

class _MainPageState extends State<MainPage> {
  int currentIndex = 0;
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: pages[currentIndex],
      bottomNavigationBar: MyBottomNavigation(),
    );
  }
}
```

Output

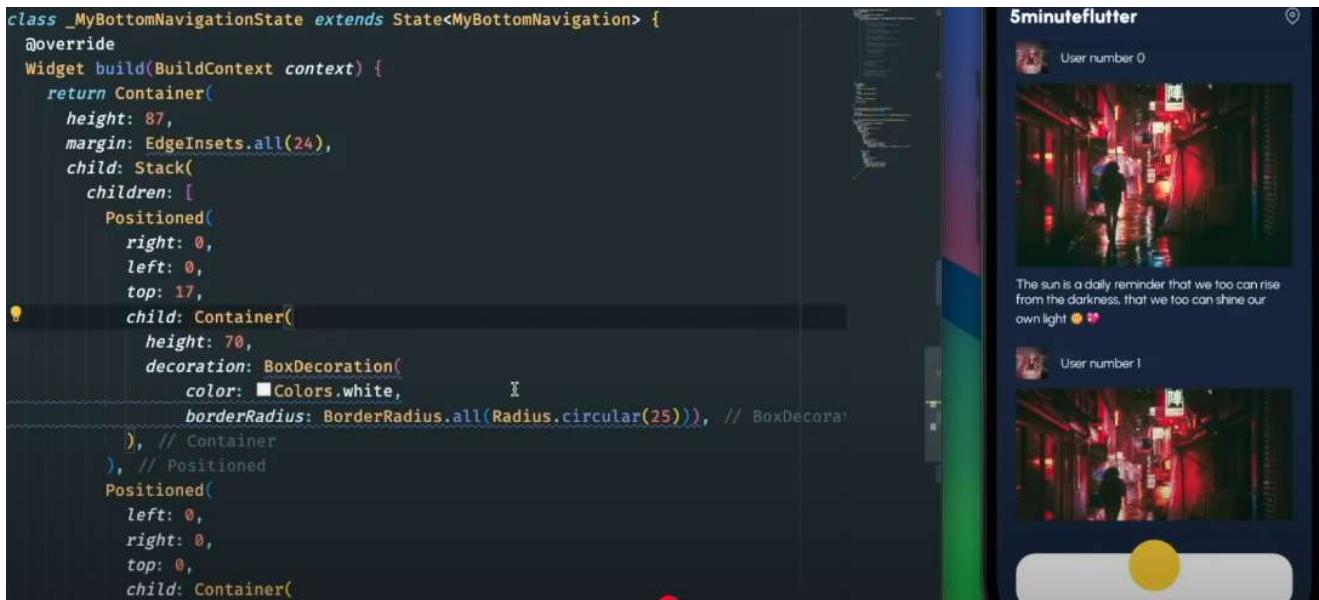


Updating



```
class _MyBottomNavigationState extends State<MyBottomNavigation> {
  @override
  Widget build(BuildContext context) {
    return Container(
      height: 87,
      margin: EdgeInsets.all(24),
      child: Stack(
        children: [
          Container(
            decoration: BoxDecoration(
              color: Colors.white,
              borderRadius: BorderRadius.all(Radius.circular(25))), // BoxDecoration
          ), // Container
          Positioned(
            left: 0,
            right: 0,
            top: 0,
            child: Container(
              width: 64,
              height: 64,
              decoration: BoxDecoration(
                color: AppColors.primary,
                shape: BoxShape.circle,
              ), // BoxDecoration
            ), // Container
        ],
      ), // Stack
    ), // Container
  }
}
```

Stack top:17



```
class _MyBottomNavigationState extends State<MyBottomNavigation> {
  @override
  Widget build(BuildContext context) {
    return Container(
      height: 87,
      margin: EdgeInsets.all(24),
      child: Stack(
        children: [
          Positioned(
            right: 0,
            left: 0,
            top: 17,
            child: Container(
              height: 70,
              decoration: BoxDecoration(
                color: Colors.white,
                borderRadius: BorderRadius.all(Radius.circular(25))), // BoxDecoration
            ), // Container
          ), // Positioned
          Positioned(
            left: 0,
            right: 0,
            top: 0,
            child: Container(

```

➤ Adding svg icons to bottom navigation

```
top: 17,
child: Container(
height: 70,
decoration: BoxDecoration(
color: Colors.white,
borderRadius: BorderRadius.all(Radius.circular(25))), // BoxDecoration
child: Row(
children: [
Expanded(
child: IconButton(
onPressed: () {},  
icon: SvgPicture.asset(AppIcons.icHome)), // IconButton /),
Expanded(
child: IconButton(
onPressed: () {},  
icon: SvgPicture.asset(AppIcons.icHome)), // IconButton /),
Expanded(
child: IconButton(
onPressed: () {},  
icon: SvgPicture.asset(AppIcons.icHome)), // IconButton /),
Expanded(
child: IconButton(
onPressed: () {},  
icon: SvgPicture.asset(AppIcons.icHome)), // IconButton /),
],
),
```



We can Applying Spacer() for extra space or use empty Expanded

- Adding valueChanged<int> onTap;

```
class MyBottomNavigation extends StatefulWidget {
  final int currentIndex;
  final ValueChanged<int> onTap;
  const MyBottomNavigation({super.key});

  @override
  State<MyBottomNavigation> createState() => _MyBottomNavigationState();
}

class _MyBottomNavigationState extends State<MyBottomNavigation> {
  @override
  Widget build(BuildContext context) {
    return Container(
      height: 87,
      margin: EdgeInsets.all(24),
      child: Stack(
        children: [
          Positioned(
            right: 0,
```

Converting widget to stateless

```
class MyBottomNavigation extends StatelessWidget {
  final int currentIndex;
  final ValueChanged<int> onTap;
  const MyBottomNavigation({super.key, required this.currentIndex, required this.onTap});

  @override
  Widget build(BuildContext context) {
    return Container(
      height: 87,
      margin: EdgeInsets.all(24),
      child: Stack(
        children: [
          Positioned(
            right: 0,
```

```
@override
Widget build(BuildContext context) {
  return Container(
    height: 87,
    margin: EdgeInsets.all(24),
    child: Stack(
      children: [
        Positioned(
          right: 0,
          left: 0,
          top: 17,
          child: Container(
            height: 70,
            decoration: BoxDecoration(
              color: Colors.white,
              borderRadius: BorderRadius.all(Radius.circular(25))), // BoxDecoration
            child: Row(
              children: [
                Expanded(
                  child: IconButton(
                    onPressed: () {
```

Defining enums for onTap

```
enum Menus{
  home,
  favorite,
  add,
  messages,
  user,
}
class MyBottomNavigation extends StatelessWidget {
  final int currentIndex;
  final ValueChanged<Menus> onTap;
  const MyBottomNavigation({
    super.key,
    required this.currentIndex,
    required this.onTap,
  });
}
```

Applying

```

child: Row(
  children: [
    Expanded(
      child: IconButton(
        onPressed: () => onTap(Menu.home),
        icon: SvgPicture.asset(AppIcons.icHome))), // IconButton /
    Expanded(
      child: IconButton(
        onPressed: () => onTap(Menu.favorite),
        icon: SvgPicture.asset(AppIcons.icFavorite))), // IconButton /
    Spacer(),
  ],
)

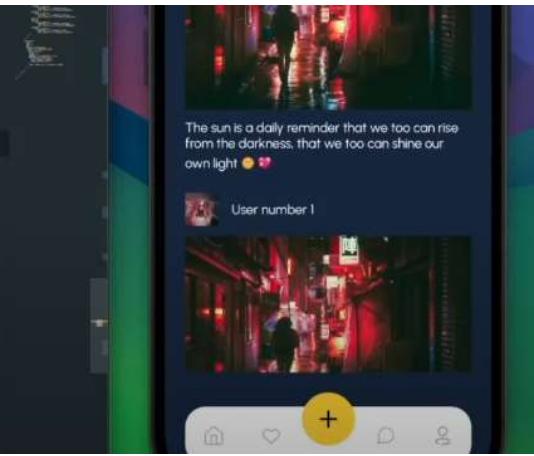
```

- [GestureDetector\(\)](#)

```

Positioned(
  left: 0,
  right: 0,
  top: 0,
  child: GestureDetector(
    onTap: () => onTap(Menu.add),
    child: Container(
      width: 64,
      height: 64,
      padding: const EdgeInsets.all(16),
      decoration: BoxDecoration(
        color: AppColors.primary,
        shape: BoxShape.circle,
      ), // BoxDecoration
      child: SvgPicture.asset(AppIcons.icAdd),
    ), // Container
  ), // GestureDetector
)

```



Handling enums update

```

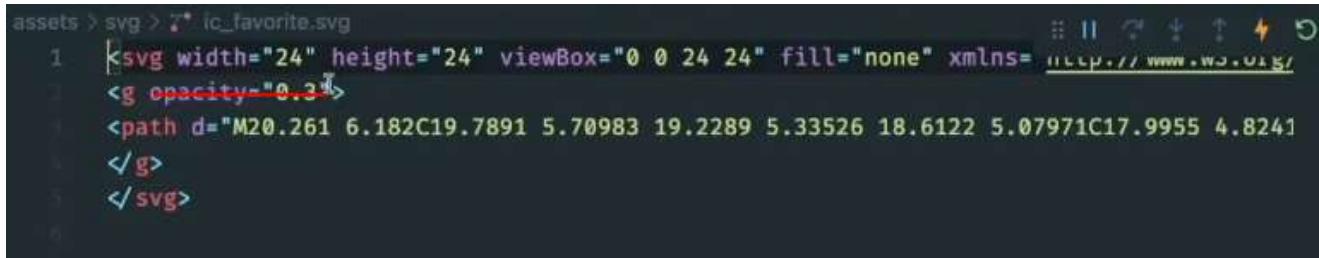
class _ MainPageState extends State<MainPage> {
  Menus currentIndex = Menus.home;
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: pages[currentIndex.index],
      bottomNavigationBar: MyBottomNavigation(
        currentIndex: currentIndex.index,
        onTap: (value) {
          setState(() {
            currentIndex = value;
          });
        },
      ), // MyBottomNavigation
    ); // Scaffold
  }
}

```

Output: bottom navigation buttons working

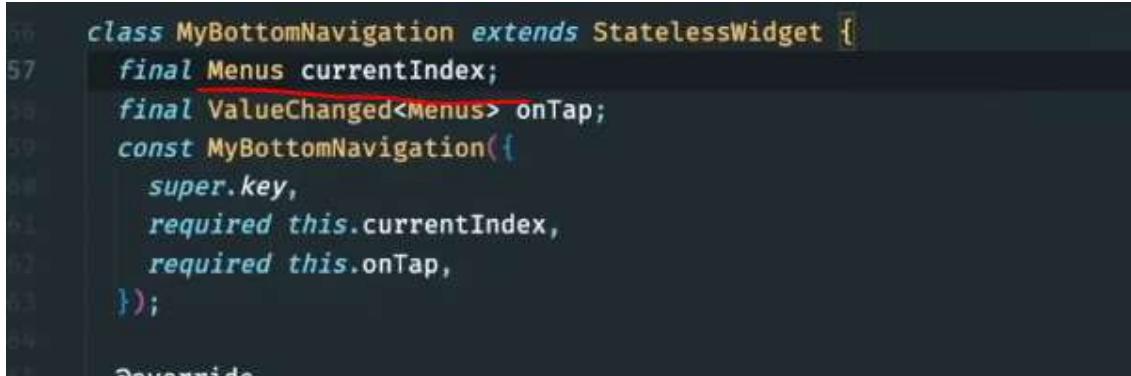
- Changing color of the selected icon in bottom

Deleting opacity from svg because we will adjust from flutter



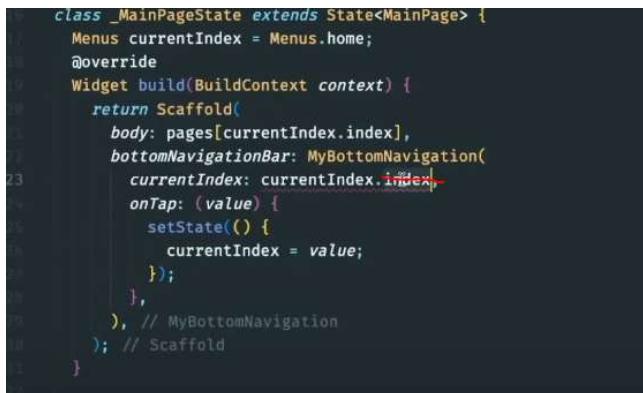
```
assets > svg > ic_favorite.svg
1  <svg width="24" height="24" viewBox="0 0 24 24" fill="none" xmlns="http://www.w3.org/1
2  <g>
3    <path d="M20.261 6.182C19.7891 5.70983 19.2289 5.33526 18.6122 5.07971C17.9955 4.8241
4    </g>
5  </svg>
```

Updating int to Menus



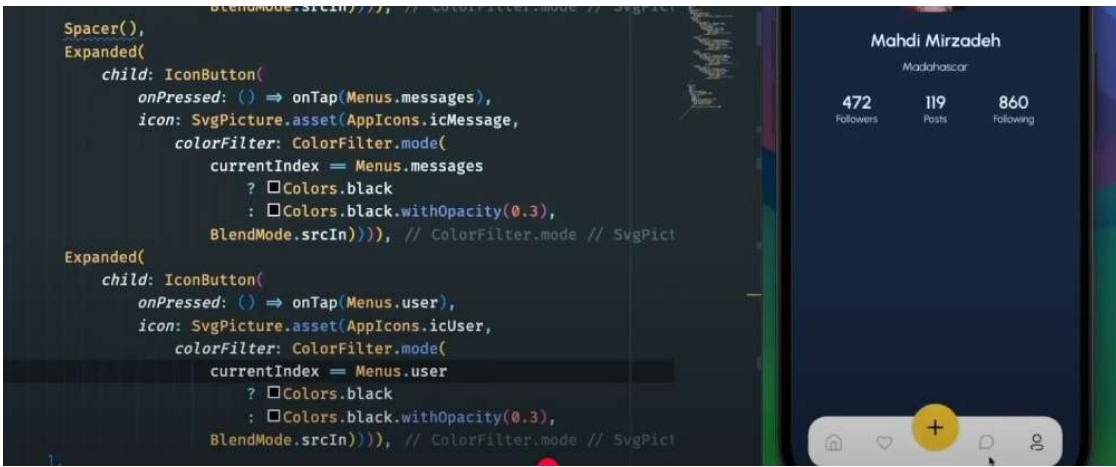
```
56
57  class MyBottomNavigation extends StatelessWidget {
58    final Menus currentIndex;
59    final ValueChanged<Menus> onTap;
60    const MyBottomNavigation({
61      super.key,
62      required this.currentIndex,
63      required this.onTap,
64    });
65
66    @override
```

Removing index

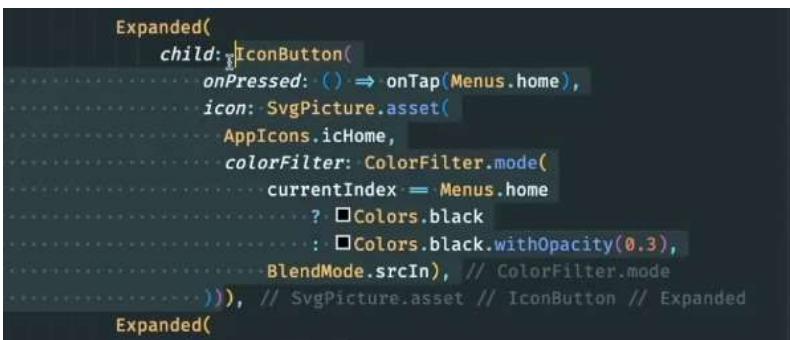


```
1  class _MainPageState extends State<MainPage> {
2    Menus currentIndex = Menus.home;
3    @override
4    Widget build(BuildContext context) {
5      return Scaffold(
6        body: pages[currentIndex.index],
7        bottomNavigationBar: MyBottomNavigation(
8          currentIndex: currentIndex.index,
9          onTap: (value) {
10            setState(() {
11              currentIndex = value;
12            });
13          },
14        ), // MyBottomNavigation
15      ); // Scaffold
16    }
17 }
```

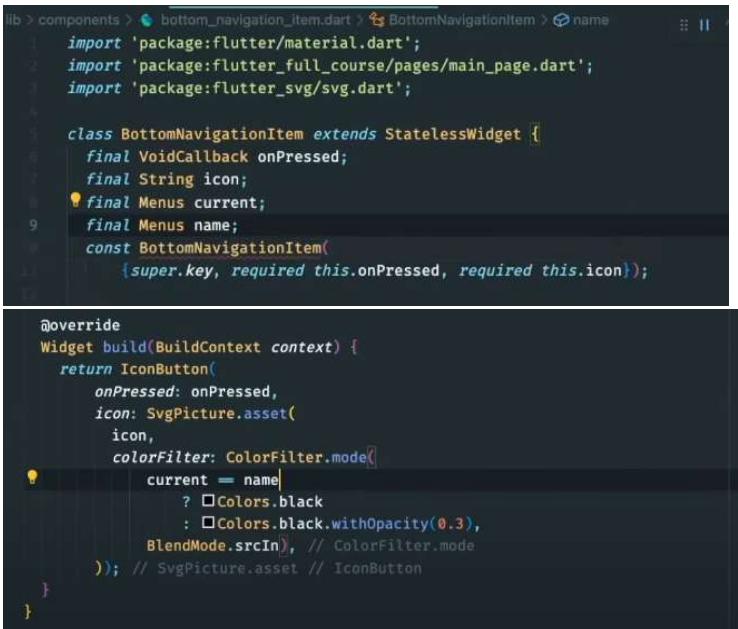
Working



Refactoring code and making new component



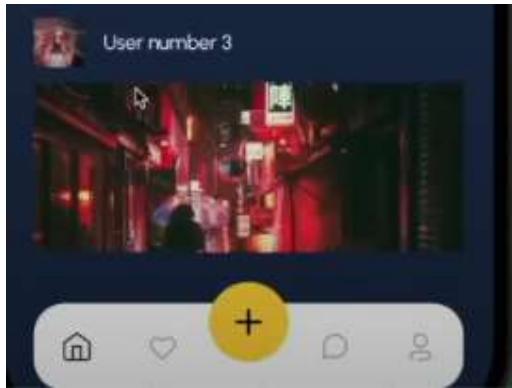
Component



Using

```
children: [
  Expanded(
    child: BottomNavigationItem(
      onPressed: () => onTap(Menu.home),
      icon: AppIcons.icHome,
      current: currentIndex,
      name: Menu.home), // BottomNavigationItem
  ), // Expanded
  Expanded(
    child: BottomNavigationItem(
      onPressed: () => onTap(Menu.favorite),
      icon: AppIcons.icFavorite,
      current: currentIndex,
      name: Menu.favorite), // BottomNavigationItem
  ), // Expanded
  Spacer(),
]
```

- Making the whole device height except the bottom navigation



- **Flutter map :**

The screenshot shows the pub.dev package page for `google_maps_flutter` version 2.3.0. The page includes the following details:

- Version:** 2.3.0
- Published:** 40 hours ago
- Publisher:** flutter.dev
- Languages:** Dart 3 compatible
- Platforms:** SDK, FLUTTER, PLATFORM, ANDROID, IOS, WEB
- Metrics:** 3.3K likes, 3317 lines, 140 pub points, 100% popularity
- Readme:** Link
- Changelog:** Link
- Example:** Link
- Installing:** Link
- Versions:** Link
- Scores:** Link
- Description:** A Flutter plugin that provides a Google Maps widget.
- Support:** SDK 20+, iOS 11+, Same as Flutter's
- Metadata:** A Flutter plugin for integrating Google Maps in iOS and Android applications.
- Repository (GitHub):** Link
- View/report issues:** Link
- Contributing:** Link

But we don't have api key

So using open source map

The screenshot shows the pub.dev package page for `flutter_map` version 5.0.0 and its corresponding documentation site.

flutter_map 5.0.0

- Published:** 3 days ago
- Publisher:** fleaflet.dev
- Languages:** Dart 3 compatible
- Platforms:** SDK, FLUTTER, PLATFORM, ANDROID, IOS, LINUX, MACOS, WEB, WINDOWS
- Metrics:** 1.2K likes, 1286 lines, 140 pub points, 99% popularity
- Readme:** Link
- Changelog:** Link
- Example:** Link
- Installing:** Link
- Versions:** Link
- Scores:** Link
- Description:** A versatile mapping package for Flutter. Simple and easy to learn, yet completely customizable and configurable, it's the best choice for mapping in your Flutter app.
- Latest Version:** v5.0.0
- Checks & Tests:** success
- points:** 140/140
- Metadata:** A versatile mapping package for Flutter that's
- Repository (GitHub):** Link
- View/report issues:** Link
- Contributing:** Link

flutter_map

The documentation site for `flutter_map` includes:

- Navigation:** flutter_map Docs v5, GitHub, pub.dev, discord.gg, Live Web Demo, Search
- Table of Contents:** flutter_map, How Does It Work?, GETTING STARTED (Installation, Quickstart, Examples, Migrating To v5), USAGE (Base Widget, Options, Layers, Controller, Event Handling, Full API Reference), LAYERS (Title Layer).
- Content:** flutter_map (Mapping package for Flutter, based off of 'leaflet.js'. Simple and easy to learn, yet completely customizable and configurable, it's the best choice for mapping in your Flutter app.), Demonstration (Setting up an interactive and compliant map is simpler than making your lunch-time coffee! It can be accomplished in just under 30 lines and a minute or two to install.), Feature Highlights (This code snippet demonstrates **everything** you need for a simple map (in just over 20 lines!), but of course, FM is capable of much more than just this, and you could find yourself lost in the many options available and possibilities opened!).
- Actions:** Export as PDF, Copy link, Edit on GitHub, Get Help.

Installing

The screenshot shows the official documentation for the flutter_map package. The left sidebar has sections like 'flutter_map', 'How Does It Work?', 'GETTING STARTED', 'Installation' (which is selected), 'Quickstart', 'Examples', 'Migrating To v5', 'USAGE', 'Base Widget', 'Options', 'Layers', 'Controller', 'Event Handling', and 'Full API Reference'. The main content area is titled 'Install' and contains two methods: 'From pub.dev' and 'From github.com'. The 'From pub.dev' method includes a command-line import command: `flutter pub add flutter_map` and `flutter pub add latlong2`. The 'From github.com' method includes a note about committing from Git.

Manually installation by adding keys in pubspec.yaml

➤ Not passing any version cause it takes last version

```
pubspec.yaml
version: 1.0.0+1
environment:
  sdk: '>2.19.2 <3.0.0'
dependencies:
  flutter:
    sdk: flutter
  flutter_svg: ^2.0.5
  flutter_map:
  latlong2:
```

The screenshot shows the official [flutter_map](#) documentation page for the **Tile Layer**. The left sidebar lists various layers: Base Widget, Options, Layers, Controller, Event Handling, Full API Reference, LAYERs, and Tile Layer (which is currently selected). The main content area is titled "Tile Layer" and describes it as a "CARTO/XYZ/Slippy Map Only" layer. It explains that the basis of any map is a `TileLayer`, which displays square raster images in a continuous grid, sourced from the Internet or a local file system. A note states that `flutter_map` supports [WMS Usage](#), but most map tiles are accessed through the CARTO/XYZ/Slippy Map standard, where the mapping library (`flutter_map`) fills in XYZ placeholders in a URL. Below this is a code snippet for the `TileLayer` class from the `flutter_map.plugin_api` library:

```
TileLayer(
  urlTemplate: 'https://tile.openstreetmap.org/{z}/{x}/{y}.png',
  userAgentPackageName: 'dev.fleaflet.flutter_map.example',
  // tileProvider: NetworkTileProvider(),
),
), // Tilelayer
], // FlutterMap
); // Scaffold
}
}
```

A warning message at the bottom right of the main content area reads: "⚠ You must comply to your tile server's ToS. Failure to do so may result in you being banned from their services. The OpenStreetMap Tile Server (as used above) can be found [here](#). Other servers may...".

➤ nearby_page.dart

The screenshot shows a Dart code editor with the file `nearby_page.dart` open. The code defines a `NearbyPage` widget that extends `StatelessWidget`. It imports several packages including `flutter/material`, `flutter_full_course/components/toolbar`, `flutter_full_course/config/app_strings`, `flutter_map/plugin_api`, and `latlong2/latlong`. The `build` method creates a `Scaffold` with a `Toolbar` and a `FlutterMap` as the body. The `FlutterMap` has `options: MapOptions(center: LatLng(51.509364, -0.128928), zoom: 10)` and a single child, a `TileLayer`. The `urlTemplate` for the `TileLayer` is set to `'https://tile.openstreetmap.org/{z}/{x}/{y}.png'`, and the `userAgentPackageName` is set to `'dev.fleaflet.flutter_map.example'`. There is a comment indicating that the `tileProvider` can be replaced by `NetworkTileProvider()`.

The screenshot shows a Dart code editor with the file `app_strings.dart` open. It contains static const definitions for various strings used in the app. These include editProfile, firstName, lastName, phoneNumber, location, birthday, male, female, other, gender, and nearby. The `nearby` string is highlighted in yellow.

```
static const editProfile = 'Edit Profile';
static const firstName = 'First name';
static const lastName = 'Last name';
static const phoneNumber = 'Phone number';
static const location = 'Location';
static const birthday = 'Birthday';
static const male = 'Male';
static const female = 'Female';
static const other = 'Other';

static const gender = 'Gender';
static const nearby = 'Nearby';
}
```

Updating main.dart via app_routes.dart

```
lib > config > app_routes.dart > AppRoutes > pages
import 'package:flutter_full_course/pages/edit_profile_page.dart';
import 'package:flutter_full_course/pages/home_page.dart';
import 'package:flutter_full_course/pages/login_page.dart';
import 'package:flutter_full_course/pages/main_page.dart';
import 'package:flutter_full_course/pages/nearby_page.dart';

class AppRoutes {
  static final pages = {
    login: (context) => LoginPage(),
    home: (context) => HomePage(),
    main: (context) => MainPage(),
    editProfile: (context) => EditProfilePage(),
    nearby: (context) => NearbyPage(),
  };

  static const login = '/';
  static const home = '/home';
  static const main = '/main';
  static const editProfile = '/edit_profile';
  static const nearby = '/nearby';
}
```

Output



Updating to avoid being blocked

The screenshot shows the official flutter_map documentation page. The left sidebar has sections like 'Base Widget', 'Options', 'Layers', 'Controller', 'Event Handling', and 'Full API Reference'. Under 'LAYERS', 'Tile Layer' is selected. The main content area is titled 'Tile Providers' and contains a note: 'It is possible to use more than one tile layer, and can be used with transparency-opacity. The children list works like the children of a Stack: last is on top.' Below this is a section titled 'userAgentPackageName' with a note: 'Always specify the userAgentPackageName argument to avoid being blocked by your tile server. It should be passed the application's package name, such as 'com.example.app'. This is important to avoid blocking by tile servers due to high-levels of unidentified traffic. If no value is passed, it defaults to 'unknown'. This is passed through to the NetworkTileProvider (if in use) in a suitably formatted string, where it forms the 'User-Agent' header, overriding any custom user agent specified in the HTTP client.'

Can add any name eg:

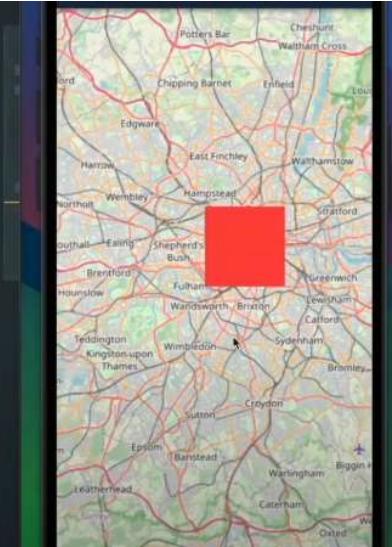
```
lib> pages > nearby_page.dart > NearbyPage > build
import 'package:flutter/material.dart';
import 'package:flutter_full_course/components/toolbar.dart';
import 'package:flutter_full_course/config/app_strings.dart';
import 'package:flutter_map/plugin_api.dart';
import 'package:latlong2/latlong.dart';

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

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: Toolbar(title: AppStrings.nearby),
      body: FlutterMap(
        options: MapOptions(center: LatLng(51.509364, -0.128928), zoom: 10),
        children: [
          TileLayer(
            urlTemplate: 'https://tile.openstreetmap.org/{z}/{x}/{y}.png',
            userAgentPackageName: 'dev.ces.flutter',
            // tileProvider: NetworkTileProvider(),
          ), // TileLayer
        ],
      ), // FlutterMap
    ); // Scaffold
  }
}
```

Adding Marker by using MarkerLayer

```
Widget build(BuildContext context) {
  return Scaffold(
    appBar: Toolbar(title: AppStrings.nearby),
    body: FlutterMap(
      options: MapOptions(center: LatLng(51.509364, -0.128928), zoom: 10),
      children: [
        TileLayer(
          urlTemplate: 'https://tile.openstreetmap.org/{z}/{x}/{y}.png',
          userAgentPackageName: 'dev.ces.flutter',
          // tileProvider: NetworkTileProvider(),
        ), // TileLayer
        MarkerLayer(
          markers: [
            Marker(
              point: LatLng(51.509364, -0.128928),
              height: 100,
              width: 100,
              builder: (context) {
                return Container(
                  color: Colors.red,
                ); // Container
              }
            ),
          ],
        ),
      ],
    ),
  );
}
```



• Showing username and location on map

```
nearby_page.dart >NearbyPage > build
height: 50,
point: LatLng(51.509364, -0.128928),
builder: (context) {
  return Column(
    children: [
      Container(
        padding: const EdgeInsets.symmetric(
          horizontal: 8, vertical: 4), // EdgeInsets.symmetric
        decoration: BoxDecoration(
          color: Colors.white,
          borderRadius:
            BorderRadius.all(Radius.circular(14))), // BoxDecoration
        child: Text(
          "Username",
          style: TextStyle(color: Colors.black),
        ), // Text
      ), // Container
      SvgPicture.asset(
        AppIcons.icLocation,
        colorFilter:
          ColorFilter.mode(Colors.black, BlendMode.srcIn),
      ) // SvgPicture.asset
    ],
  ); // Column
},
```



- Network request Api in flutter

The screenshot shows the pub.dev page for the `http` package. At the top, it says "Published 18 days ago · ⚡ dart.dev Dart 3 compatible". Below that are tabs for `SDK`, `DART`, `FLUTTER`, `PLATFORM`, `ANDROID`, `IOS`, `LINUX`, `MACOS`, `WEB`, and `WINDOWS`. To the right, there's a "6.3K" badge with a chart icon. Further down, there are tabs for `Readme`, `Changelog`, `Example`, `Installing`, `Versions`, and `Scores`. On the far right, there are statistics: `6365` likes, `140` pub points, and `100%` popularity. Below these are buttons for "pub v1.0.0" and "publisher dart.dev".

OR

The screenshot shows the pub.dev page for the `dio` package. It has a similar layout to the `http` page, with "Published 3 days ago · ⚡ flutter.cn Dart 3 compatible" at the top. It includes tabs for `SDK`, `DART`, `FLUTTER`, `PLATFORM`, `ANDROID`, `IOS`, `LINUX`, `MACOS`, `WEB`, and `WINDOWS`. A "5.6K" badge is present. Below the tabs are `Readme`, `Changelog`, `Example`, `Installing`, `Versions`, and `Scores` tabs. The main content area shows the `dio` package details, including its version `v5.2.0+1` and developer `dev.v5.2.0+1`. It also lists supported languages: English | 简体中文. A brief description follows: "A powerful HTTP client for Dart/Flutter, which supports global configuration, interceptors, FormData, request cancellation, file uploading/downloading, timeout, and custom adapters etc.".

The screenshot shows the `pubspec.yaml` file for a project. It includes entries for `login_page.dart` and `pubspec.yaml`. The `dependencies:` section is highlighted, showing the following code:

```
# To automatically upgrade your package dependencies to the latest major version, consider running 'flutter pub upgrade --major-versions'. All dependencies can be manually updated by changing the version number to the latest version available on pub.dev. To see which dependencies have new versions available, run 'flutter pub outdated'.
dependencies:
  flutter:
    sdk: flutter
  flutter_svg: ^2.0.5
  flutter_map:
  latlong2:
  http: ^1.0.0
```

Line numbers 25 through 36 are visible on the left.

Existing server

The screenshot shows the Postman interface with the following details:

- Workspace:** My Workspace
- Collection:** Flutter Course
- Request:** POST /login
- Body:** JSON (selected)

```
1 "username": "m1",
2 "password": "123456"
```
- Response:** Status: 200 OK, Time: 453 ms, Size: 637 B
- Body Content:**

```
1 {
2     "message": "success9",
3     "data": [
4         {
5             "id": 24,
6             "username": "m1",
7             "firstname": "mahdi",
8             "lastname": "mirzadeh",
9             "mobile": null,
10            "birthday": null,
11            "location": {
12                "lat": 53.72764593342592,
13                "lng": -104.66723521271976,
14                "name": "Canada"
15            },
16            "gender": null,
17            "visibleGender": false,
18            "token": "eyJhbGciOiJIUzI1NiInRScC1oIkoXVCJ9.eyJ2VymFtZSI6Im0xTiwiaWQ1OjI0LCJpYXQiOjE2ODQ0MTI5NDV9.DtfEkQcCVJLj2t1UQ2QapBURbhuiGsa...
```

Backend Server created using flutter dart

The screenshot shows the VS Code interface with the following details:

- File:** pubspec.yaml — flutter_course_server
- Content:**

```
name: server
description: An new Dart Frog application
version: 1.0.0+1
publish_to: none

environment:
  sdk: '>=2.19.2 <3.0.0'

dependencies:
  dart_frog: ^0.3.6
  dart_frog_web_socket: ^0.1.0-dev.3
```
- Terminal Output:**

```
ptexes/OS/opt/homebrew/Cellar/icu4c/72.1/lib/libcui18n.70.dylib' (no such file), '/opt/homebrew/Cellar/icu4c/72.1/1/libcui18n.70.dylib' (no such file), '/usr/local/lib/libcui18n.70.dylib' (no such file), '/usr/lib/libcui18n.70.dylib' (no such file, not in dyld cache)
  + flutter_course_server export PATH="$HOME/.pub-cache/bin"
  + Flutter_course_server dart_frog dev
  ✓ Running on http://localhost:8080 (0.1s)
The Dart VM service is listening on http://127.0.0.1:8181/XIlxRrLyk_0=/
The Dart DevTools debugger and profiler is available at: http://127.0.0.1:8181/XIlxRrLyk_0=/devtools?uri=ws://127.0.0.1:8181/XIlxRrLyk_0=/ws
[hotreload] Hot reload is enabled.
  {userId: 24, username: m1, password: $2b$10$ylfVp5XRhh81KvL9vSsI..gvK2cf506Iq8thMuAWveeTQSmlKuwny, firstname: mahdi, lastname: mirzadeh, mobile: null, birthday: null, location: {"lat": 53.72764593342592, "lng": -104.66723521271976, "name": "Canada"}, gender: null, visibleGender: null}
```

Applying Api



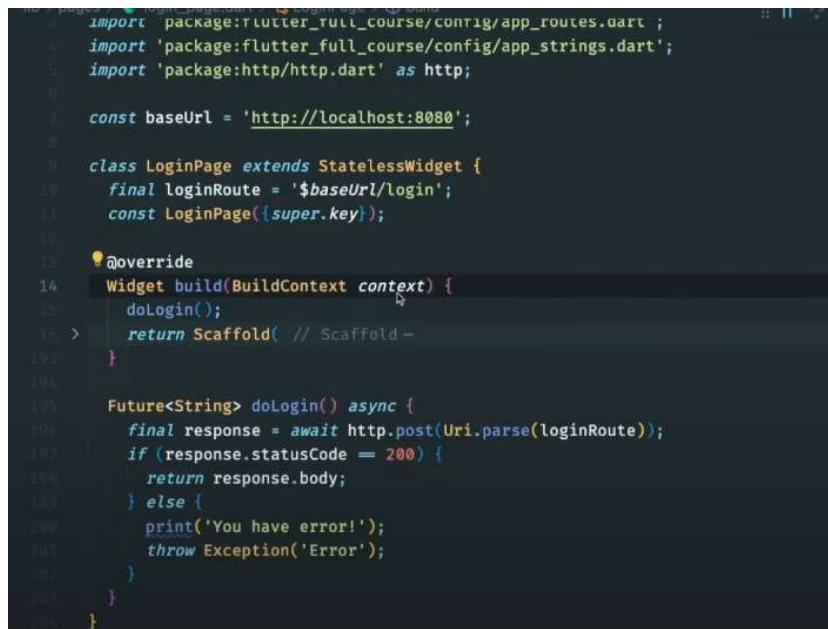
```
login_page.dart • pubspec.yaml
lib > pages > login_page.dart > LoginPage
import 'package:flutter/material.dart';
import 'package:flutter_full_course/config/app_icons.dart';
import 'package:flutter_full_course/config/app_routes.dart';
import 'package:flutter_full_course/config/app_strings.dart';

const baseUrl = 'http://localhost:8080';

class LoginPage extends StatelessWidget {
  final loginRoute = '$baseUrl/login';
  const LoginPage({super.key});

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: SingleChildScrollView(
```

Structure



```
import 'package:flutter_full_course/config/app_routes.dart';
import 'package:flutter_full_course/config/app_strings.dart';
import 'package:http/http.dart' as http;

const baseUrl = 'http://localhost:8080';

class LoginPage extends StatelessWidget {
  final loginRoute = '$baseUrl/login';
  const LoginPage({super.key});

  @override
  Widget build(BuildContext context) {
    doLogin();
    return Scaffold( // Scaffold -
  }

  Future<String> doLogin() async {
    final response = await http.post(Uri.parse(loginRoute));
    if (response.statusCode == 200) {
      return response.body;
    } else {
      print('You have error!');
      throw Exception('Error');
    }
  }
}
```

Applying

a) Using TextEditingController()

```
lib > pages > login_page.dart > LoginPage > LoginPage
1 import 'package:flutter/material.dart';
2 import 'package:flutter_full_course/config/app_icons.dart';
3 import 'package:flutter_full_course/config/app_routes.dart';
4 import 'package:flutter_full_course/config/app_strings.dart';
5 import 'package:http/http.dart' as http;
6
7 const baseUrl = 'http://localhost:8080';
8
9 class LoginPage extends StatelessWidget {
10   final loginRoute = '$baseUrl/login';
11   final usernameController = TextEditingController();
12   final passwordController = TextEditingController();
13   LoginPage({super.key});
14
15   @override
16   Widget build(BuildContext context) {
17
18     return Scaffold(
19       body: SingleChildScrollView(
20         TextField(
21           controller: usernameController,
22           decoration: InputDecoration(
23             hintText: AppStrings.username,
24             border: OutlineInputBorder(
25               borderRadius: BorderRadius.all(Radius.circular(12))), // OutlineInputBorder
26               filled: true,
27               fillColor: Colors.white.withOpacity(0.5),
28             ), // InputDecoration
29           ), // TextField
30           SizedBox(
31             height: 16,
32           ), // SizedBox
33           TextField(
34             controller: passwordController,
35             decoration: InputDecoration(
36               hintText: AppStrings.password,
37               border: OutlineInputBorder(
38                 borderRadius: BorderRadius.all(Radius.circular(12))), // OutlineInputBorder
39                 filled: true,
40                 fillColor: Colors.white.withOpacity(0.5),
41               ), // InputDecoration
42           ),
43           Future<String> doLogin() async {
44             final username = usernameController.text;
45             final password = passwordController.text;
46             final body = {
47               'username':username,
48               'password':password,
49             };
50             final response = await http.post(Uri.parse(loginRoute),body: jsonEncode(body));
51             if (response.statusCode == 200) {
52               print(response.body);
53               return response.body;
54             } else {
55               print('You have error!');
56               throw Exception('Error');
57             }
58           }
59         );
60       )
61     );
62   }
63 }
```

Output

A screenshot of a Flutter application running in an IDE. On the left, the code editor shows a snippet of Dart code:

```
217     print('You have error!');
218     throw Exception('Error');
219   }
220 }
221
```

The main window displays a mobile application interface. At the top, there is a search bar with the placeholder "Search". Below it is a login form with fields for "username" containing "mi" and "password" containing "123456". A "Forgot password?" link is visible. A yellow "Login" button is centered below the fields. To the right of the login form is a numeric keyboard.

At the bottom of the screen, there is a navigation bar with icons for back, forward, and other system functions. The status bar at the very top shows signal strength, battery level, and the time.

b) Using Variable

```
0 > pages > login_page.dart > LoginPage > password
  1 import 'dart:convert';
  2
  3 import 'package:flutter/material.dart';
  4 import 'package:flutter_full_course/config/app_icons.dart';
  5 import 'package:flutter_full_course/config/app_routes.dart';
  6 import 'package:flutter_full_course/config/app_strings.dart';
  7 import 'package:http/http.dart' as http;
  8
  9 const baseUrl = 'http://localhost:8080';
 10
 11 class LoginPage extends StatelessWidget {
 12   final loginRoute = '$baseUrl/login';
 13   var username = '';
 14   var password = '';
 15   LoginPage({super.key});
 16
 17   @override
 18   Widget build(BuildContext context) {
 19     return Scaffold(
 20       body: SingleChildScrollView(
 21         child: SizedBox(
 22           height: MediaQuery.of(context).size.height,
 23           child: Padding(
 24             padding: EdgeInsets.all(24),
 25             filled: true,
 26             fillColor: Colors.white.withOpacity(0.5),
 27           ), // InputDecoration
 28         ), // TextField
 29         SizedBox(
 30           height: 16,
 31         ), // SizedBox
 32         TextField(
 33           onChanged: (value) {
 34             username = value;
 35           },
 36           decoration: InputDecoration(
 37             hintText: AppStrings.password,
 38             border: OutlineInputBorder(
 39               borderRadius: BorderRadius.all(Radius.circular(12)),
 40               filled: true,
 41               fillColor: Colors.white.withOpacity(0.5),
 42             ), // InputDecoration
 43           ),
 44         );
 45       );
 46     Future<String> doLogin() async {
 47       final body = {
 48         'username': username,
 49         'password': password,
 50       };
 51       final response = await http.post(
 52         Uri.parse(loginRoute),
 53         body: jsonEncode(body),
 54       );
 55       if (response.statusCode == 200) {
 56         print(response.body);
 57         return response.body;
 58       } else {
 59         print(response.body);
 60         print('You have error!');
 61         throw Exception('Error');
 62       }
 63     }
 64   }
 65 }
```

Output

Same

- Creating models for our apis

user.dart model

using

```
Future<User> doLogin() async {
  final body = {
    'username': username,
    'password': password,
  };
  final response = await http.post(
    Uri.parse(loginRoute),
    body: jsonEncode(body),
  );
  if (response.statusCode == 200) {
    print(response.body);
    final json = jsonDecode(response.body);

    final response = await http.post(
      Uri.parse(loginRoute),
      body: jsonEncode(body),
    );
    if (response.statusCode == 200) {
      print(response.body);
      final json = jsonDecode(response.body);
      final user = User.fromJson(json['data']);
      return user;
    } else {
      print(response.body);
      print('You have error!');
      throw Exception('Error');
    }
  }
}
```

Using

```
SizedBox(
  height: 48,
  width: double.infinity,
  child: ElevatedButton(
    onPressed: () async {
      final user = await doLogin();
      Navigator.of(context)
        .pushReplacementNamed(AppRoutes.main);
    },
    style: ElevatedButton.styleFrom(
      backgroundColor: Colors.amber,
      foregroundColor: Colors.black,
    ),
    child: Text(AppStrings.login),
  ), // ElevatedButton
), // SizedBox
Spacer(),
Text(
```

Output: logout

The screenshot shows a development environment with two main panes. On the left is the code editor with files `login_page.dart`, `user.dart`, and `pubspec.yaml`. The `login_page.dart` file contains Dart code for handling user login. On the right is a preview of the Flutter application running on an iPhone. The app's title is "5minuteflutter". It displays two users: "User number 0" and "User number 1", each with a profile picture and a quote. The quote for User 0 is: "The sun is a daily reminder that we too can rise from the darkness, that we too can shine our own light ☀️". The quote for User 1 is partially visible.

```
lib> pages > login_page.dart > LoginPage > doLogin
  final response = await http.post(
    Uri.parse(loginRoute),
    body: jsonEncode(body),
  );
  if (response.statusCode == 200) {
    print(response.body);
    final json = jsonDecode(response.body);
    final user = User.fromJson(json['data']);
    return user;
  } else {
    print(response.body);
    print('You have error!');
    throw Exception('Error');
  }
}
PROBLEMS 165 OUTPUT DEBUG CONSOLE ... Filter (e.g. text, [exclude])
Restarted application in 1,597ms.
Flutter: {"message": "success!", "data": {"id": 24, "username": "m1", "firstname": "mahdi", "lastname": "mirzadeh", "mobile": null, "birthday": null, "location": {"lat": 53.72764593342592, "lng": -104.66723521271976, "name": "Canada"}, "gender": null, "visibleGender": false, "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9eyJlc2VybmFtZSI6Im0xIiwiaWQiOjI0LCJpYXQiOjE2ODY0MjI5MjF9.Hk8m5uK1z4bbBQXjQ3ZBSTMkNAwJt0h8bT80p5sUKB0"}}
```

STATE MANAGEMENT

a) Passing data while navigation (eg: props)

```
SizedBox(  
    height: 48,  
    width: double.infinity,  
    child: ElevatedButton(  
        onPressed: () async {  
            final user = await doLogin();  
            Navigator.of(context).push(PageRouteBuilder(  
                pageBuilder: (context, animation, secondaryAnimation) {  
                    return MainPage(user: user);  
                },  
            )); // PageRouteBuilder  
        },  
    ),
```

And

```
lib > pages > main_page.dart > MainPage > MainPage  
import 'package:flutter_full_course/pages/home_page.dart';  
import 'package:flutter_full_course/pages/profile_page.dart';  
import 'package:flutter_full_course/styles/app_colors.dart';  
import 'package:flutter_svg/svg.dart';  
  
class MainPage extends StatefulWidget {  
    final User user;  
    const MainPage({super.key, required this.user});
```

Then now we want to pass it to profile so

```
lib > pages > main_page.dart > _MainPageState > pages  
    currentIndex = value;  
    }  
},  
, // MyBottomNavigation  
); // Scaffold  
}  
  
final pages = [  
    HomePage(),  
    Center(  
        child: Text('Favorite'),  
    ), // Center  
    Center(  
        child: Text('Add Post'),  
    ), // Center  
    Center(  
        child: Text('Messages'),  
    ), // Center  
    ProfilePage(user: widget.user),  
];
```

Since in above we cannot access widget so converting pages obj to function

```
lib > pages > main_page.dart > _MainPageState > build
  class _MainPageState extends StatelessWidget {
    Menus currentIndex = Menus.home;
    @override
    Widget build(BuildContext context) {
      return Scaffold(
        extendBody: true,
        body: pages()[currentIndex.index],
        bottomNavigationBar: MyBottomNavigation(
          currentIndex: currentIndex,
          onTap: (value) {
            setState(() {
              currentIndex = value;
            });
          },
        ), // MyBottomNavigation
      ); // Scaffold
    }

    pages() => [
      HomePage(),
      Center(
        child: Text('Favorite'),
      ), // Center
      Center(
        child: Text('Add Post'),
      ),
    ];
  }
}
```

using

```
lib > pages > profile_page.dart > ProfilePage
  import 'package:flutter_full_course/config/app_routes.dart';
  import 'package:flutter_full_course/config/app_strings.dart';
  import 'package:flutter_full_course/model/user.dart';
  import 'package:flutter_full_course/styles/app_text.dart';

  enum ProfileMenu { edit, logout }

  class ProfilePage extends StatelessWidget {
    final User user;
    const ProfilePage({super.key, required this.user});
    @override
    Widget build(BuildContext context) {
      return Scaffold(
        appBar: Toolbar(
          title: AppStrings.profile,
          actions: [
            PopupMenuButton<ProfileMenu>(
              onSelected: (value) {
                switch (value) {
                  case ProfileMenu.edit:
                    Navigator.of(context).pushNamed(AppRoutes.editProfile);
                    break;
                }
              },
            ),
          ],
        ),
      );
    }
  }
}
```



```
        body: Column(  
            children: [  
                UserAvatar(  
                    size: 90,  
                ), // UserAvatar  
                SizedBox(  
                    height: 24,  
                ), // SizedBox  
                Text(  
                    '${user.firstname} ${user.lastname}',  
                    style: AppText.header2,  
                ), // Text  
                SizedBox(  
                    height: 12,
```

Updating the app_route.dart

```
  :kage:flutter_full_course/model/user.dart',
  :kage:flutter_full_course/pages/edit_profile_page.dart',
  :kage:flutter_full_course/pages/home_page.dart';
  :kage:flutter_full_course/pages/login_page.dart';
  :kage:flutter_full_course/pages/main_page.dart';
  :kage:flutter_full_course/pages/nearby_page.dart';

  routes {
final pages = {
  (context) => LoginPage(),
  context) => HomePage(),
  context) => MainPage(user: User(1, 'firstname', 'lastn',
  file: (context) => EditProfilePage(),
  (context) => NearbyPage(),

  const login = '/';
  const home = '/home';
  const main = '/main';
  const editProfile = '/edit_profile';
  const nearby = '/nearby';
```

Output



b) Main Provider class using inherited (eg: like contextApi)

- Since passing variables from one class to different like (props) is difficult so doing it simply (eg: Context API in react native)
- Inherited widget

```
lib > user_provider.dart > UserProvider > UserProvider
import 'package:flutter/material.dart';
import 'package:flutter_full_course/model/user.dart';

class UserProvider extends InheritedWidget {
    final Widget child;
    final User user;
    const UserProvider({
        required this.user,
        super.key,
        required this.child,
    }) : super(child: child);

    @override
    bool updateShouldNotify(covariant UserProvider oldWidget) {
        // TODO: implement updateShouldNotify
        throw UnimplementedError();
    }
}
```

Using in main.dart

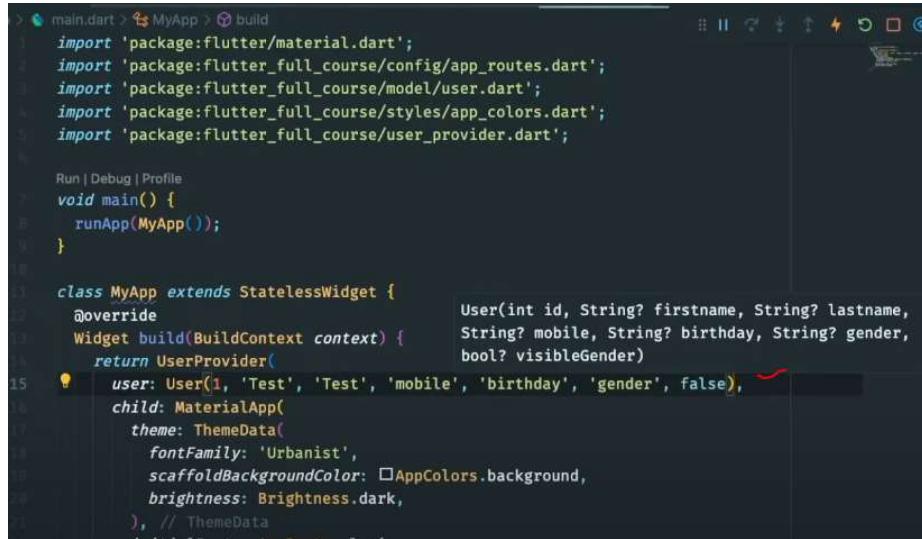
```
lib > main.dart > MyApp > build
import 'package:flutter/material.dart';
import 'package:flutter_full_course/config/app_routes.dart';
import 'package:flutter_full_course/styles/app_colors.dart';
import 'package:flutter_full_course/user_provider.dart';

Run | Debug | Profile
void main() {
    runApp(MyApp());
}

class MyApp extends StatelessWidget {
    @override
    Widget build(BuildContext context) {
        return UserProvider(
            child: MaterialApp(
                theme: ThemeData(
                    fontFamily: 'Urbanist',
                    scaffoldBackgroundColor: AppColors.background,
                    brightness: Brightness.dark,
                ),
                initialRoute: AppRoutes.login,
                routes: AppRoutes.pages,
            ), // MaterialApp
        ); // UserProvider
    }
}
```

Using:

For example we pass user

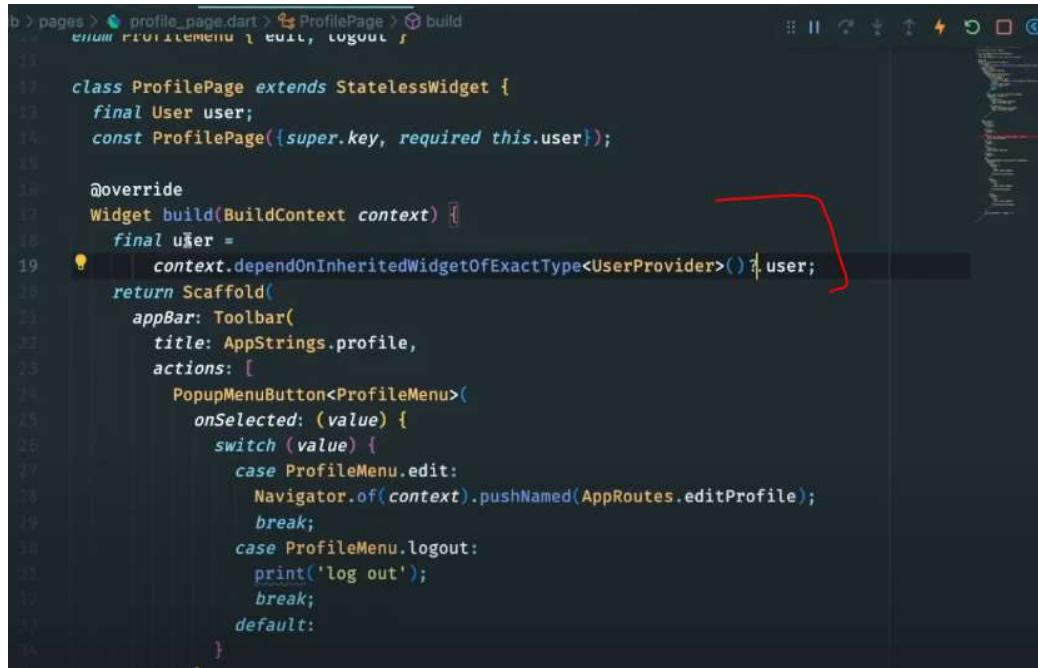


```
> main.dart > MyApp > build
import 'package:flutter/material.dart';
import 'package:flutter_full_course/config/app_routes.dart';
import 'package:flutter_full_course/model/user.dart';
import 'package:flutter_full_course/styles/app_colors.dart';
import 'package:flutter_full_course/user_provider.dart';

Run | Debug | Profile
void main() {
  runApp(MyApp());
}

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return UserProvider(
      user: User(1, 'Test', 'Test', 'mobile', 'birthday', 'gender', false),
      child: MaterialApp(
        theme: ThemeData(
          fontFamily: 'Urbanist',
          scaffoldBackgroundColor: AppColors.background,
          brightness: Brightness.dark,
        ),
        // ThemeData
      ),
    );
  }
}
```

Accessing user variable from parent in child



```
b > pages > profile_page.dart > build
enum ProfileMenu { edit, logout }

class ProfilePage extends StatelessWidget {
  final User user;
  const ProfilePage({super.key, required this.user});

  @override
  Widget build(BuildContext context) {
    final user =
        context.dependOnInheritedWidgetOfExactType<UserProvider>()?.user;
    return Scaffold(
      appBar: Toolbar(
        title: AppStrings.profile,
        actions: [
          PopupMenuButton<ProfileMenu>(
            onSelected: (value) {
              switch (value) {
                case ProfileMenu.edit:
                  Navigator.of(context).pushNamed(AppRoutes.editProfile);
                  break;
                case ProfileMenu.logout:
                  print('log out');
                  break;
                default:
              }
            }
          )
        ],
      ),
    );
  }
}
```

Using variable “user”

```
    body: Column(  
      children: [  
        UserAvatar(  
          size: 90,  
        ), // UserAvatar  
        SizedBox(  
          height: 24,  
        ), // SizedBox  
        Text(  
          '${user?.id} ${user?.firstname} ${user?.lastname}',  
          style: AppText.header2,  
        ), // Text  
        SizedBox(  
          height: 12,  
        ), // SizedBox
```

Output



- Updating user_provider.dart to make an static variable to easily access data in child widgets

```

> user_provider.dart > UserProvider > of
import 'package:flutter/material.dart';
import 'package:flutter_full_course/model/user.dart';

class UserProvider extends InheritedWidget {
  final Widget child;
  final User user;
  const UserProvider({
    required this.user,
    super.key,
    required this.child,
  }) : super(child: child);

  static User? of(BuildContext context) {
    return context.dependOnInheritedWidgetOfExactType<UserProvider>()?.user;
  }

  @override
  bool updateShouldNotify(covariant UserProvider oldWidget) {
    // TODO: implement updateShouldNotify
    throw UnimplementedError();
  }
}

```

Accessing user in child class

```

> pages > profile_page.dart > ProfilePage > build
import 'package:flutter_full_course/user_provider.dart';

enum ProfileMenu { edit, logout }

class ProfilePage extends StatelessWidget {
  final User user;
  const ProfilePage({super.key, required this.user});

  @override
  Widget build(BuildContext context) {
    final user = UserProvider.of(context); // Line 18
    return Scaffold(
      appBar: Toolbar(
        title: AppStrings.profile,
        actions: [
          PopupMenuButton<ProfileMenu>(
            ...
        ),
      ],
    );
  }
}

```

Output: same as previous output

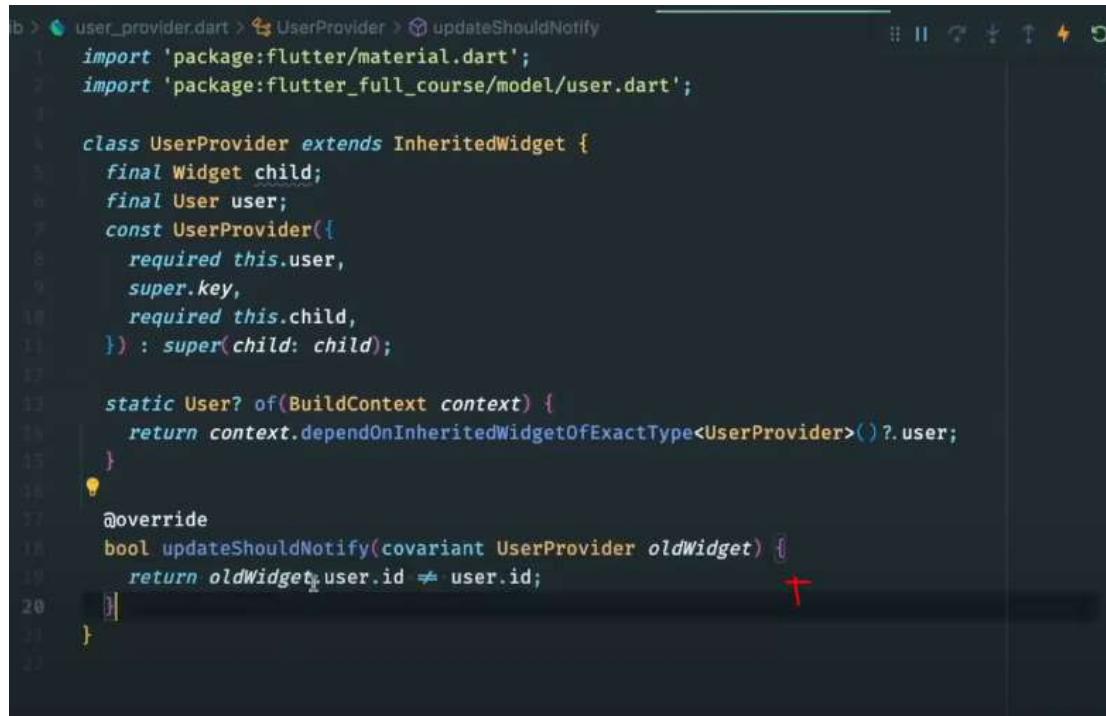
The above code is also used by flutter eg

```

@override
Widget build(BuildContext context) {
  return Scaffold(
    body: SingleChildScrollView(
      child: SizedBox(
        height: MediaQuery.of(context).size.height, // Line 85
        child: Padding(
          padding: EdgeInsets.all(24),
          child: Column(
            children: [
              Spacer(),
              Text(

```

Fixing the user_provider.dart



The screenshot shows a code editor with the file `user_provider.dart` open. The code defines a `UserProvider` class that extends `InheritedWidget`. It has fields for `child` and `user`, and a constructor that takes `user` and `child`. A static method `of` returns the provider from the context. An `updateShouldNotify` method is overridden to check if the `user.id` has changed. The code editor interface includes tabs for other files like `main.dart` and `MaterialApp`, and various toolbars and status indicators.

```
lib > user_provider.dart > UserProvider > updateShouldNotify
1 import 'package:flutter/material.dart';
2 import 'package:flutter_full_course/model/user.dart';
3
4 class UserProvider extends InheritedWidget {
5   final Widget child;
6   final User user;
7   const UserProvider({
8     required this.user,
9     super.key,
10    required this.child,
11  }) : super(child: child);
12
13 static User? of(BuildContext context) {
14   return context.dependOnInheritedWidgetOfExactType<UserProvider>()?.user;
15 }
16
17 @override
18 bool updateShouldNotify(covariant UserProvider oldWidget) {
19   return oldWidget.user.id != user.id;
20 }
21 }
```

- Handling updating user from profile

```
1 import 'package:flutter/material.dart';
2 import 'package:flutter_full_course/model/user.dart';
3
4 class UserProvider extends InheritedWidget {
5   final Widget child;
6   final UserService userService;
7   const UserProvider({
8     required this.userService,
9     super.key,
10    required this.child,
11  }) : super(child: child);
12
13   static UserService? of(BuildContext context) {
14     return context
15       .dependOnInheritedWidgetOfExactType<UserProvider>()
16       ?.userService;
17   }
18
19   @override
20   bool updateShouldNotify(covariant UserProvider oldWidget) {
21     return oldWidget.userService.user?.id != userService.user?.id;
22   }
23 }
24
25 class UserService {
26   User? _user;
27   User? get user => _user;
28   updateUser(User user) {
29     _user = user;
30   }
31 }
```

Updated

From:

```
1 class MyApp extends StatelessWidget {
2   @override
3   Widget build(BuildContext context) {
4     return UserProvider(
5       user: User(1, 'Test', 'Test', 'mobile', 'birthday', 'gender', false),
6       child: MaterialApp(
7         theme: ThemeData(
8           fontFamily: 'Urbanist',
9           scaffoldBackgroundColor: AppColors.background,
10          brightness: Brightness.dark,
11        ), // ThemeData
12        initialRoute: AppRoutes.login,
13        routes: AppRoutes.pages,
14      ), // MaterialApp
15    ); // UserProvider
16  }
17 }
```

To:

```
class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return UserProvider(
      user: UserService(),
      child: MaterialApp(
        theme: ThemeData(
          fontFamily: 'Urbanist',
          scaffoldBackgroundColor: AppColors.background,
          brightness: Brightness.dark,
        ), // ThemeData
        initialRoute: AppRoutes.login,
        routes: AppRoutes.pages,
      ), // MaterialApp
    ); // UserProvider
  }
}
```

Accessing

```
lib > pages > profile_page.dart > ProfilePage > build
1 import 'package:flutter/material.dart';
2 import 'package:flutter_full_course/components/toolbar.dart';
3 import 'package:flutter_full_course/components/user_avatar.dart';
4 import 'package:flutter_full_course/config/app_routes.dart';
5 import 'package:flutter_full_course/config/app_strings.dart';
6 import 'package:flutter_full_course/model/user.dart';
7 import 'package:flutter_full_course/styles/app_text.dart';
8 import 'package:flutter_full_course/user_provider.dart';

9 enum ProfileMenu { edit, logout }

10 class ProfilePage extends StatelessWidget {
11   final User user;
12   const ProfilePage({super.key, required this.user});

13   @override
14   Widget build(BuildContext context) {
15     final user = UserProvider.of(context)?.user;
16     return Scaffold(
17       appBar: Toolbar(
18         title: AppStrings.profile,
19         actions: [
20       ],
21     ),
22   );
23 }
```

➤ Removing the props being passed

From:

```
SizedBox(  
    height: 48,  
    width: double.infinity,  
    child: ElevatedButton(  
        onPressed: () async {  
            final user = await doLogin();  
            Navigator.of(context).push(PageRouteBuilder(  
                pageBuilder: (context, animation, secondaryAnimation) {  
                    return MainPage(  
                        user: user,  
                    ); // MainPage  
                },  
            )); // PageRouteBuilder
```

To

```
SizedBox(  
    height: 48,  
    width: double.infinity,  
    child: ElevatedButton(  
        onPressed: () async {  
            final user = await doLogin();  
            Navigator.of(context)  
                .pushReplacementNamed(AppRoutes.main);  
        },  
        style: ElevatedButton.styleFrom(  
            backgroundColor: Colors.amber,  
            foregroundColor: Colors.black,  
        ),  
        child: Text(AppStrings.login),  
    ), // ElevatedButton  
, // SizedBox
```

And removing

```
class AppRoutes {  
    static final pages = {  
        login: (context) => LoginPage(),  
        home: (context) => HomePage(),  
        main: (context) => MainPage(  
            user: User(  
                id, 'firstname', 'lastname', 'mobile', 'birthday', 'gender', false)), // User  
        editProfile: (context) => EditProfilePage(),  
        nearby: (context) => NearbyPage(),  
    };  
  
    static const login = '/';  
    static const home = '/home';  
    static const main = '/main';  
    static const editProfile = '/edit_profile';  
    static const nearby = '/nearby';  
}
```

```
class MainPage extends StatefulWidget {  
    final User user;  
    const MainPage({super.key, required this.user});  
  
    @override
```

```
pages() => [
  HomePage(),
  Center(
    child: Text('Favorite'),
  ), // Center
  Center(
    child: Text('Add Post'),
  ), // Center
  Center(
    child: Text('Messages'),
  ), // Center
  ProfilePage(
    user: widget.user,
  );
];
```

```
class ProfilePage extends StatelessWidget {
  final User user;
  const ProfilePage({super.key, required this.user});
```

From:

```
pages() => [
  HomePage(),
  Center(
    child: Text('Favorite'),
  ), // Center
  Center(
    child: Text('Add Post'),
  ), // Center
  Center(
    child: Text('Messages'),
  ), // Center
  ProfilePage()
];
```

To:

```
final pages = [
  HomePage(),
  Center(
    child: Text('Favorite'),
  ), // Center
  Center(
    child: Text('Add Post'),
  ), // Center
  Center(
    child: Text('Messages'),
  ), // Center
  ProfilePage()
];
```

Removing

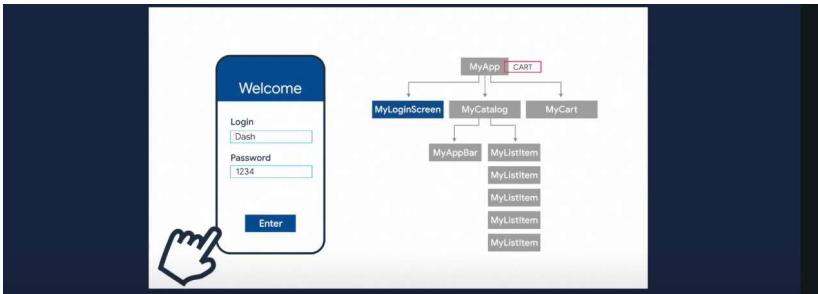
```
class _ MainPageState extends State<MainPage> {
  Menus currentIndex = Menus.home;
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      extendBody: true,
      body: pages[currentIndex.index],
      bottomNavigationBar: MyBottomNavigation(
        currentIndex: currentIndex,
        onTap: (value) {
          setState(() {
            currentIndex = value;
          });
        },
    ),
```

➤ Updating user while login

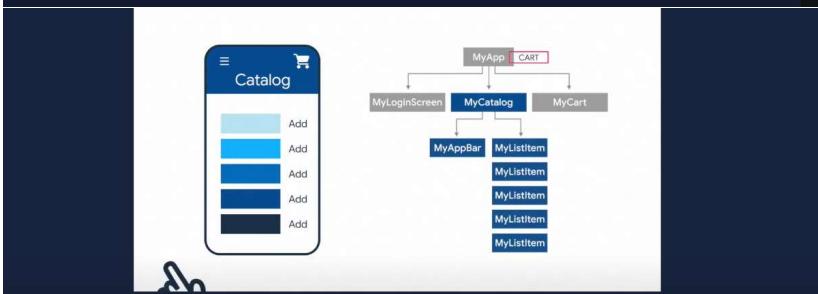
```
SizedBox(
  height: 48,
  width: double.infinity,
  child: ElevatedButton(
    onPressed: () async {
      final user = await doLogin();
      UserProvider.of(context)?.updateUser(user);
      Navigator.of(context)
        .pushReplacementNamed(AppRoutes.main);
    },
    style: ElevatedButton.styleFrom(
      backgroundColor: Colors.amber,
      foregroundColor: Colors.black,
    ),
)
```

Output: showing user who is logged in

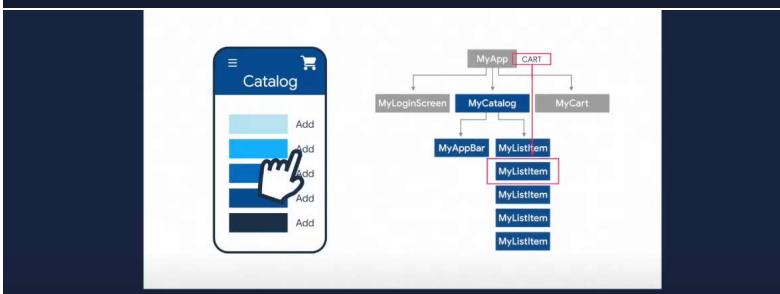
```
SizedBox(
  height: 48,
  width: double.infinity,
  child: ElevatedButton(
    onPressed: () async {
      final user = await doLogin();
      UserProvider.of(context)?.updateUser(user);
      Navigator.of(context)
        .pushReplacementNamed(AppRoutes.main);
    },
    style: ElevatedButton.styleFrom(
      backgroundColor: Colors.amber,
      foregroundColor: Colors.black,
    ),
)
```



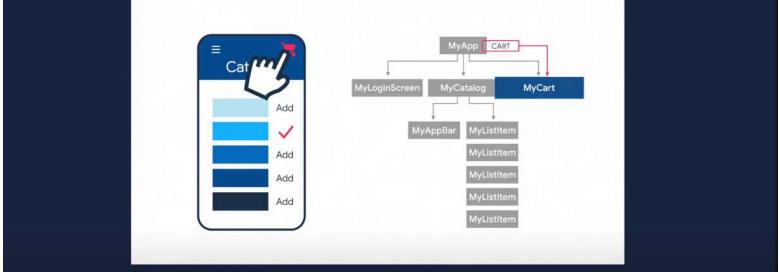
docs.flutter.dev/data-and-backend/state-mgmt



docs.flutter.dev/data-and-backend/state-mgmt

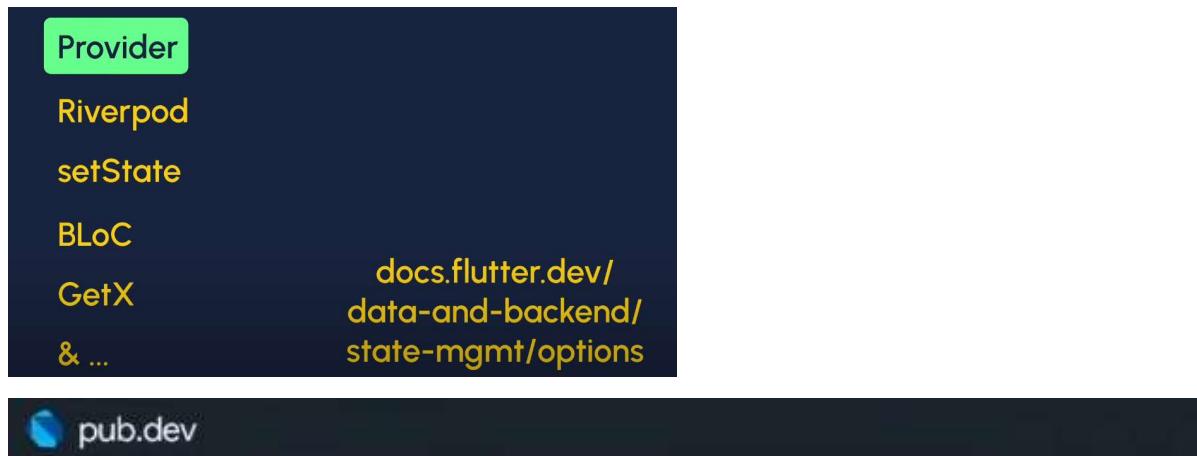


docs.flutter.dev/data-and-backend/state-mgmt



docs.flutter.dev/data-and-backend/state-mgmt

c) Like Redux Store



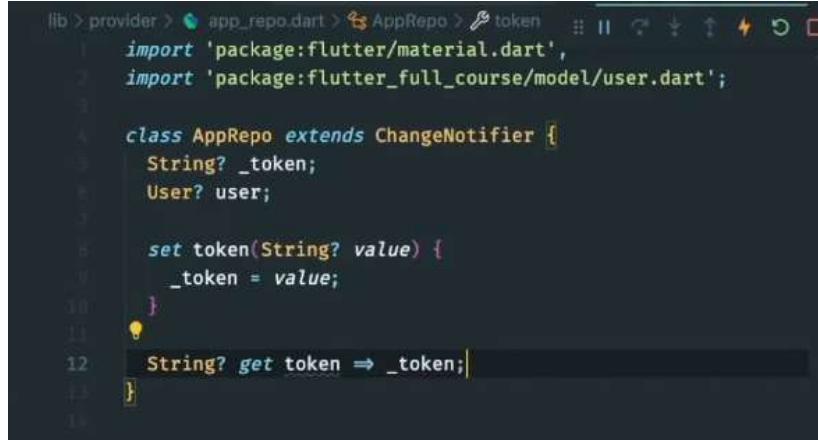
This screenshot shows the detailed page for provider 6.0.5. At the top, it says "provider 6.0.5" with a copy icon. Below that is a "Published 6 months ago · dash-overflow.net (dart 3 compatible)" link. A navigation bar includes "SDK", "FLUTTER", "PLATFORM", "ANDROID", "IDS", "LINUX", "MACOS", "WEB", and "WINDOWS". A tab bar at the bottom has "Readme" (which is underlined), "Changelog", "Example", "Installing", "Versions", and "Scores". Below the tabs are language links: English | French | Português | 简体中文 | Español | 한국어 | ಬ್ರಹ್ಮಾ | 日本語. There are also build status badges for "Build: passing", "codecov: 99%", and "chat: 220 online". A large circular "Flutter Favorite" logo is centered. Below the logo is a description: "A wrapper around InheritedWidget to make them easier to use and more reusable."

This screenshot shows the "dependencies:" section of a Dart file. It lists several packages: flutter, flutter_svg (~2.0.5), flutter_map, latlong2, http (~1.0.0), and provider (~6.0.5). The "provider" dependency is highlighted with a blue background.

```
dependencies:  
  flutter:  
    sdk: flutter  
  flutter_svg: ^2.0.5  
  flutter_map:  
  latlong2:  
    http: ^1.0.0  
  provider: ^6.0.5
```

Deleted user_provider.dart file

Create app_repo.dart



```
lib>provider> app_repo.dart > AppRepo > token
import 'package:flutter/material.dart';
import 'package:flutter_full_course/model/user.dart';

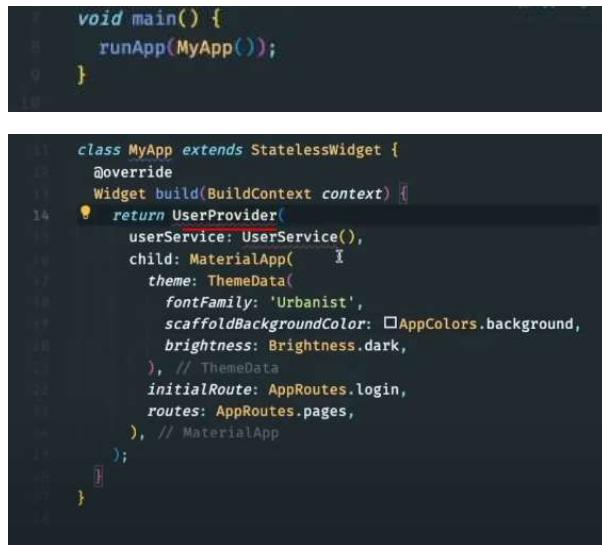
class AppRepo extends ChangeNotifier {
  String? _token;
  User? user;

  set token(String? value) {
    _token = value;
  }

  String? get token => _token;
}
```

Update

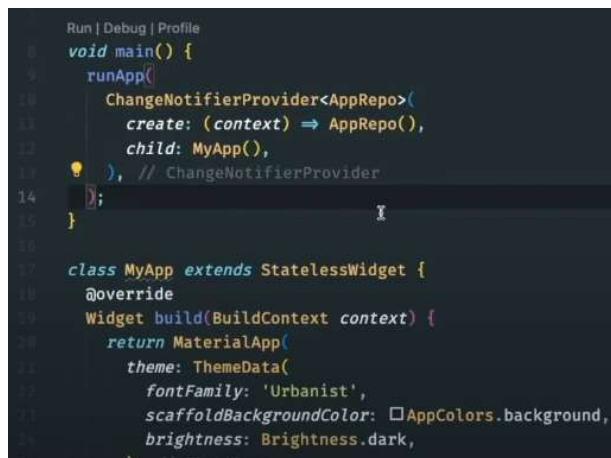
This:



```
void main() {
  runApp(MyApp());
}

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return UserProvider(
      userService: UserService(),
      child: MaterialApp(
        theme: ThemeData(
          fontFamily: 'Urbanist',
          scaffoldBackgroundColor: AppColors.background,
          brightness: Brightness.dark,
        ), // ThemeData
        initialRoute: AppRoutes.login,
        routes: AppRoutes.pages,
      ), // MaterialApp
    );
  }
}
```

To:



```
Run | Debug | Profile
void main() {
  runApp(
    ChangeNotifierProvider<AppRepo>(
      create: (context) => AppRepo(),
      child: MyApp(),
    ), // ChangeNotifierProvider
  );
}

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      theme: ThemeData(
        fontFamily: 'Urbanist',
        scaffoldBackgroundColor: AppColors.background,
        brightness: Brightness.dark,
      ),
    );
  }
}
```

- Creating app_config.dart inside config

```
lib > config > app_config.dart > AppConfig
class AppConfig {
  static const baseUrl = 'http://localhost:8080';
}
```

- Creating login_response.dart (for api response not model)

```
lib > data > response > login_response.dart > LoginResponse
import 'package:flutter_full_course/model/user.dart';

class LoginResponse {
  final User user;
  final String token;

  LoginResponse(this.user, this.token);
  factory LoginResponse.fromJson(Map<String, dynamic> json)
    LoginResponse(User.fromJson(json), json['token']);
}
```

- Creating login_service.dart (to manage api request)

```
lib > data > service > login_service.dart > LoginService
import 'dart:convert';

import 'package:flutter_full_course/config/app_config.dart';
import 'package:flutter_full_course/data/response/login_response.dart';
import 'package:flutter_full_course/model/user.dart';
import 'package:http/http.dart' as http;

class LoginService {
  final String username;
  final String password;

  LoginService(this.username, this.password);
  Future<LoginResponse> call() async {
    final result = await http.post(Uri.parse('${AppConfig.baseUrl}'),
      body: jsonEncode({
        'username': username,
        'password': password,
      }));
    return LoginResponse.fromJson(jsonDecode(result.body)['data']);
  }
}
```

- Creating login_provider.dart

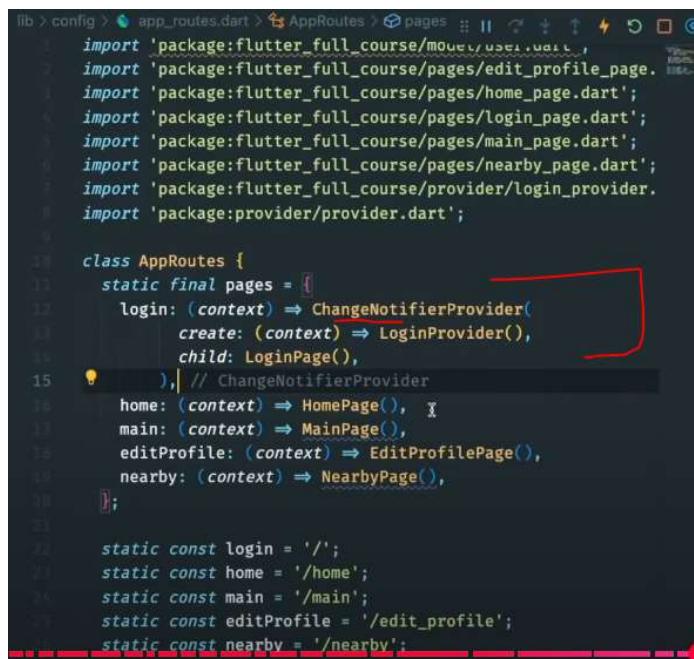


```
lib > provider > login_provider.dart > LoginProvider > lib
import 'package:flutter/material.dart';
import 'package:flutter_full_course/data/response/login_resp
import 'package:flutter_full_course/data/service/login_servi

class LoginProvider extends ChangeNotifier {
    var username = '';
    var password = '';

    Future<LoginResponse> login() async {
        return LoginService(username, password).call();
    }
}
```

Updating app_routes.js



```
lib > config > app_routes.dart > AppRoutes > lib
import 'package:flutter_full_course/move/user.dart';
import 'package:flutter_full_course/pages/edit_profile_page.dart';
import 'package:flutter_full_course/pages/home_page.dart';
import 'package:flutter_full_course/pages/login_page.dart';
import 'package:flutter_full_course/pages/main_page.dart';
import 'package:flutter_full_course/pages/nearby_page.dart';
import 'package:flutter_full_course/provider/login_provider.dart';
import 'package:provider/provider.dart';

class AppRoutes {
    static final pages = [
        login: (context) => ChangeNotifierProvider(
            create: (context) => LoginProvider(),
            child: LoginPage(),
        ), // ChangeNotifierProvider
        home: (context) => HomePage(),
        main: (context) => MainPage(),
        editProfile: (context) => EditProfilePage(),
        nearby: (context) => NearbyPage(),
    ];
}

static const login = '/';
static const home = '/home';
static const main = '/main';
static const editProfile = '/edit_profile';
static const nearby = '/nearby';
```

Updating onchanged

```
Text(
  AppStrings.loginToContinue,
  style: TextStyle(
    color: Colors.white,
  ), // Text
  Spacer(),
  TextField(
    onChanged: (value) {
      Provider.of<LoginProvider>(context).username = value;
    },
    decoration: InputDecoration(
      hintText: AppStrings.username,
      border: OutlineInputBorder(
        borderRadius: BorderRadius.all(Radius.circular(12))
      ),
      filled: true,
      fillColor: Colors.white.withOpacity(0.5),
    ), // InputDecoration
  ), // TextField
  SizedBox(
    height: 16,
  ), // SizedBox
  TextField(
    onChanged: (value) {
      Provider.of<LoginProvider>(context).password = value;
    },
    decoration: InputDecoration(
      hintText: AppStrings.password,
      border: OutlineInputBorder(
        borderRadius: BorderRadius.all(Radius.circular(12))
      ),
      filled: true,
      fillColor: Colors.white.withOpacity(0.5),
    ), // InputDecoration
  ), // TextField
), // Column
), // Container
```

Do this for above (listen:false)

```
onChanged: (value) {
  Provider.of<LoginProvider>(context, listen: false).us
}, // TextField
decoration: InputDecoration(
  hintText: AppStrings.username,
  border: OutlineInputBorder(
    borderRadius: BorderRadius.all(Radius.circular(12))
  ),
  filled: true,
  fillColor: Colors.white.withOpacity(0.5),
), // InputDecoration
), // Column
), // Container
```

And

From:

```
SizedBox(  
    height: 48,  
    width: double.infinity,  
    child: ElevatedButton(  
        onPressed: () async {  
            final user = await doLogin();  
            UserProvider.of(context)?.updateUser(u  
                Navigator.of(context)  
                    .pushReplacementNamed(AppRoutes.ma  
            },  
            style: ElevatedButton.styleFrom(  
                backgroundColor: Colors.amber,  
                foregroundColor: Colors.black,  
            ),  
            child: Text(AppStrings.login),  
        ), // ElevatedButton
```

To: (login and save user and token in AppRepo)

```
    child: ElevatedButton(
      onPressed: () {
        Provider.of<LoginProvider>(context).login().then((value) {
          Provider.of<AppRepo>(context).user = value.user;
          Provider.of<AppRepo>(context).token = value.token;
          Navigator.of(context)
            .pushReplacementNamed(AppRoutes.main);
        });
      },
      style: ElevatedButton.styleFrom(
        backgroundColor: Colors.amber,
        foregroundColor: Colors.black,
```

Do> listen:false

```
onPressed: () {
    Provider.of<LoginProvider>(context, listen: false)
        .login()
        .then((value) {
            Provider.of<AppRepo>(context, listen: false).use
                value.user;
            Provider.of<AppRepo>(context, listen: false).tok
                value.token;
            Navigator.of(context)
                .pushReplacementNamed(AppRoutes.main);
        });
},
style: ElevatedButton.styleFrom(
    backgroundColor: Colors.amber,
```

- Updating profile page



```
enum ProfileMenu { edit, logout }

class ProfilePage extends StatelessWidget {
  const ProfilePage({
    super.key,
  });

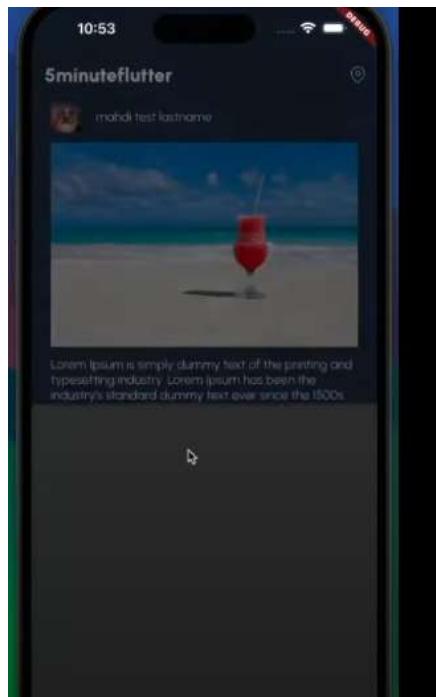
  @override
  Widget build(BuildContext context) {
    final user = Provider.of<AppRepo>(context).user; ✓
    return Scaffold(
      appBar: Toolbar(
        title: AppStrings.profile,
```

- For further about api with and without model see starting pages

- Adding Model to add post
- `showModalBottomSheet(...)`

```
class _MainPageState extends State<MainPage> {
    Menus currentIndex = Menus.home;
    @override
    Widget build(BuildContext context) {
        return Scaffold(
            extendBody: true,
            body: pages[currentIndex.index],
            bottomNavigationBar: MyBottomNavigation(
                currentIndex: currentIndex,
                onTap: (value) {
                    if (value == Menus.add) {
                        showModalBottomSheet(
                            context: context,
                            builder: (context) {
                                return Container();
                            },
                        );
                    }
                    setState(() {
                        currentIndex = value;
                    });
                },
            ),
        );
    }
}
```

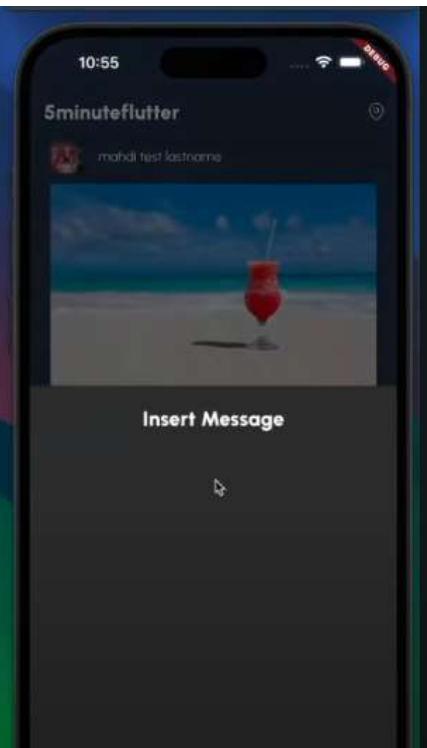
Output



- Creating new_post_modal.dart

```
class _MainPageState extends State<MainPage> {
  Menus currentIndex = Menus.home;
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      extendBody: true,
      body: pages[currentIndex.index],
      bottomNavigationBar: MyBottomNavigation(
        currentIndex: currentIndex,
        onTap: (value) {
          if (value == Menus.add) {
            showModalBottomSheet(
              context: context,
              builder: (context) {
                return NewPostModal();
              },
            );
          }
        },
      ),
      setState(() {
        currentIndex = value;
      });
    );
  }
}
```

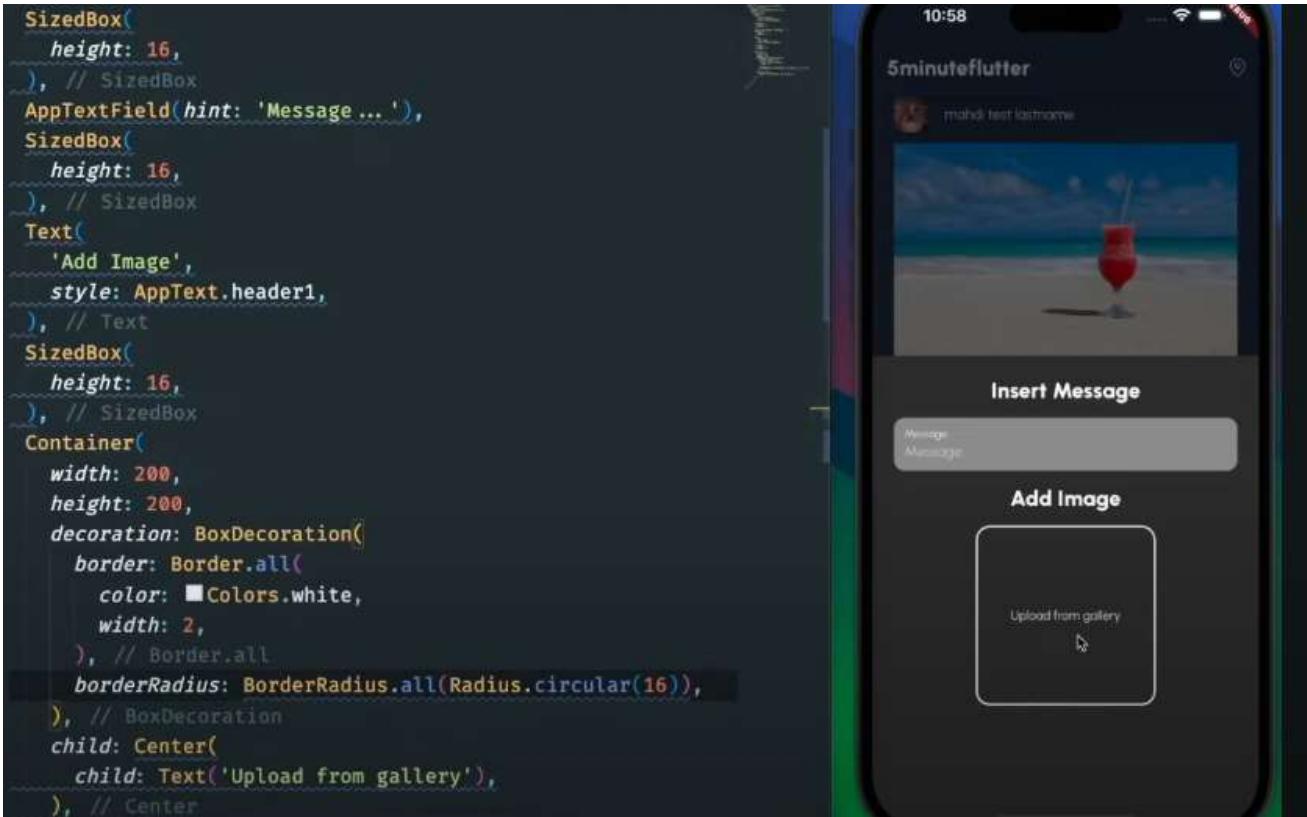
Creating



The image shows a Flutter application running on an iPhone. A modal dialog is displayed over the main screen. The modal has a dark background and contains a single line of text: "Insert Message". Above the modal, the main screen shows a beach scene with a red cocktail glass on the sand.

```
lib > components > new_post_modal.dart > NewPostModal
1 import 'package:flutter/material.dart',
2 import 'package:flutter_full_course/styles/app_text.dart';
3
4 class NewPostModal extends StatelessWidget {
5   const NewPostModal({super.key});
6
7   @override
8   Widget build(BuildContext context) {
9     return Container(
10       padding: const EdgeInsets.all(24),
11       child: Column(
12         children: [
13           Text(
14             'Insert Message',
15             style: AppText.header1,
16           ), // Text
17           ], // Column
18       ); // Container
19     }
20 }
```

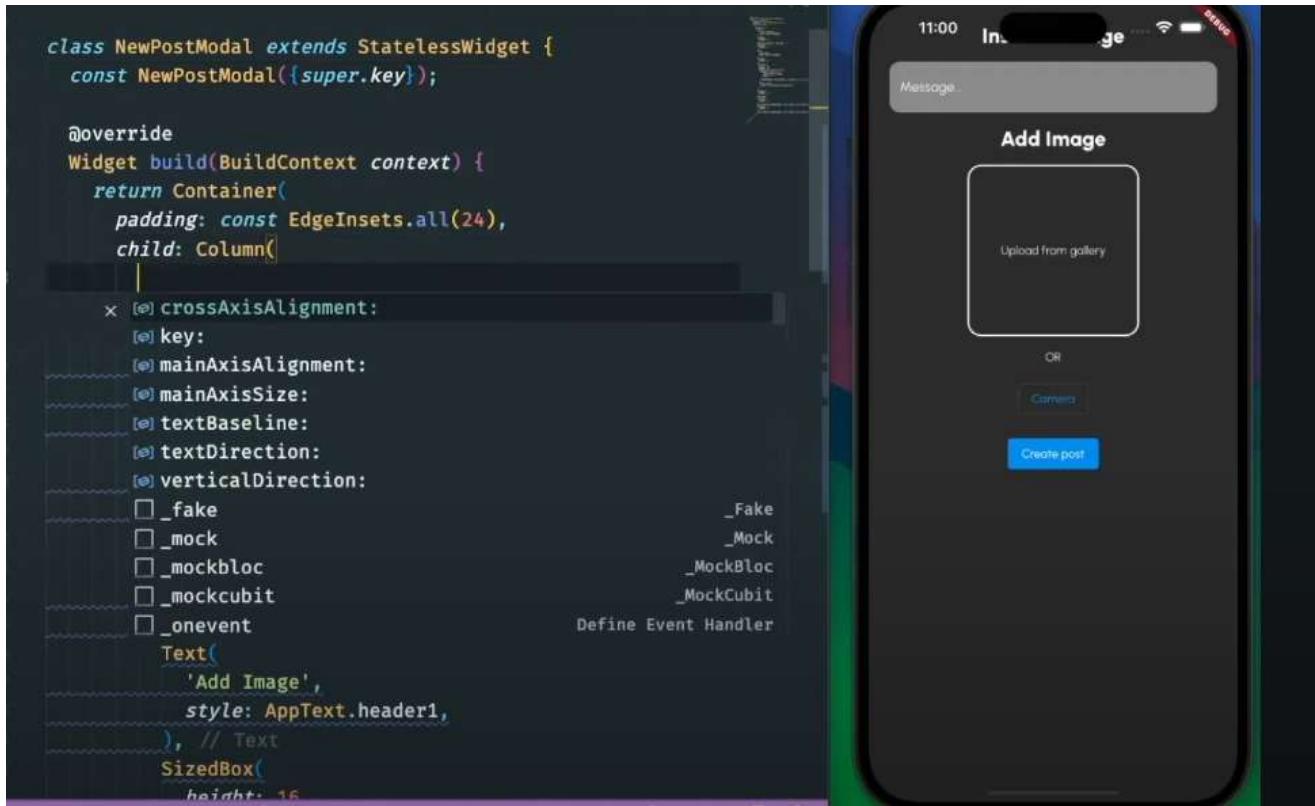
Updating



- **isScrollControlled:true**

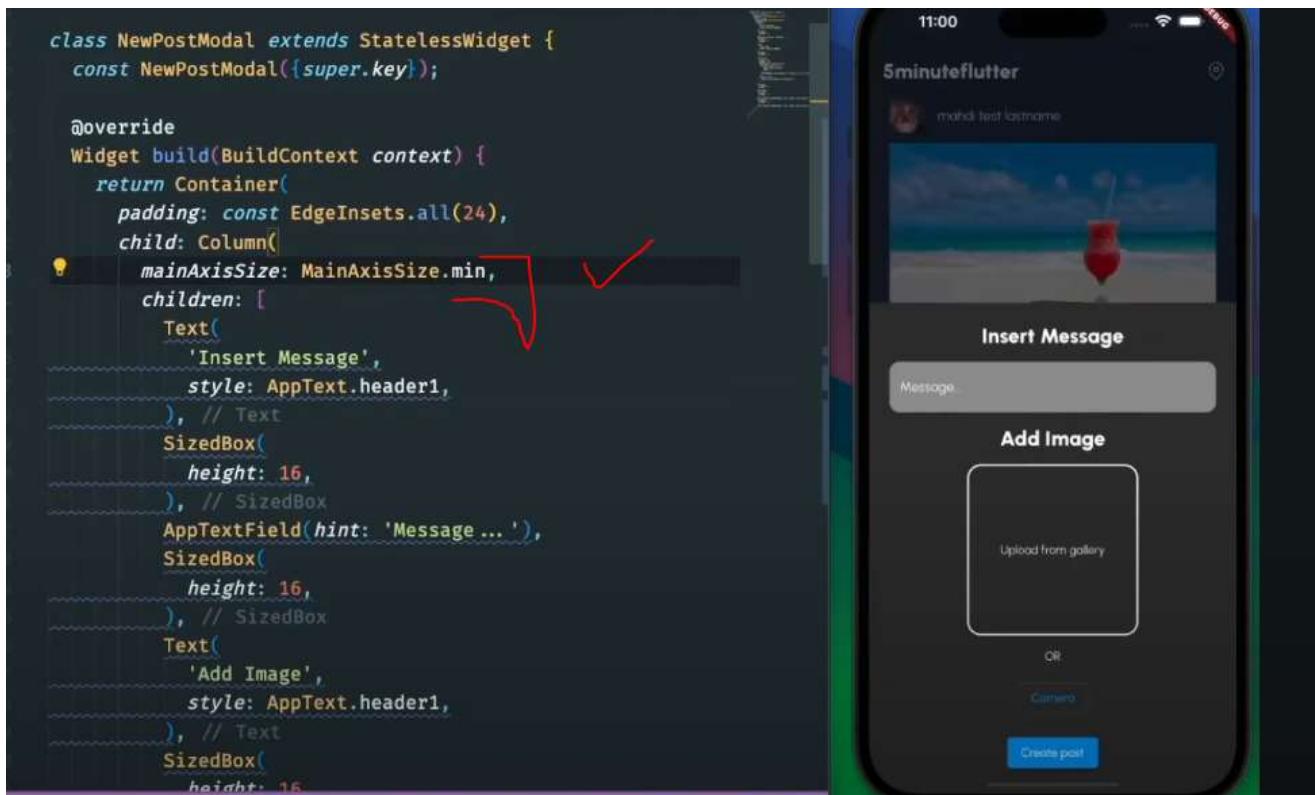
```
onTap: (value) {
  if (value == Menus.add) {
    showModalBottomSheet(
      context: context,
      isScrollControlled: true,
      builder: (context) {
        return NewPostModal();
      },
    );
    return;
  }
  setState(() {
    currentIndex = value;
  });
}
```

But it became fullscreen

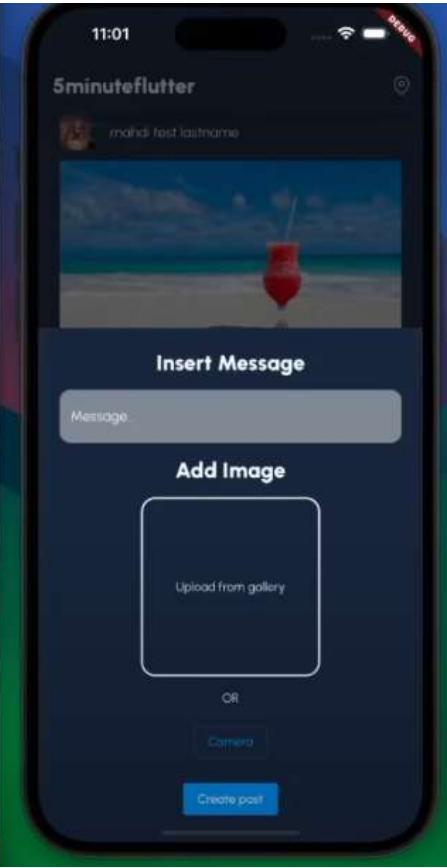


To take only content space

- `mainAxisSize:MainAxisSize.min`



Color and radius



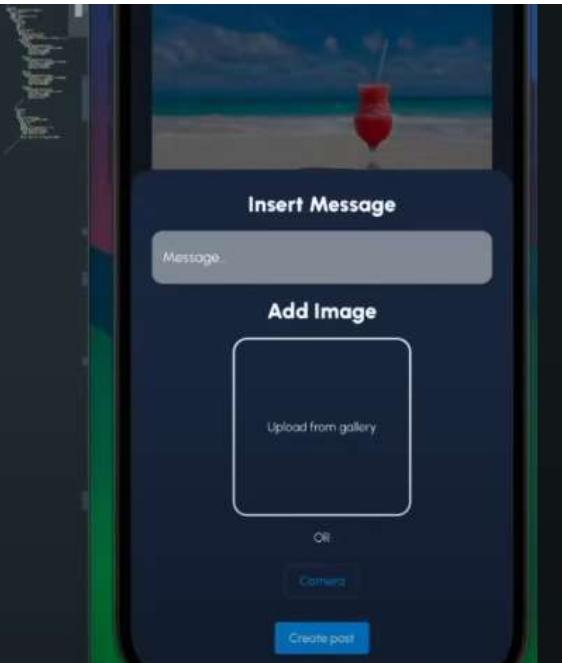
A screenshot of a Flutter application running on an iPhone 14 Pro Max simulator. The app shows a modal dialog titled "Insert Message". The modal has a dark blue background and rounded corners. Inside the modal, there is a text input field labeled "Message.", a button labeled "Add Image", and a section for uploading images from the gallery or camera. A red callout bubble is visible at the top right of the screen.

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

  @override
  Widget build(BuildContext context) {
    return Container(
      decoration: BoxDecoration(
        color: AppColors.background,
        borderRadius: BorderRadius.only(
          topLeft: Radius.circular(24),
          topRight: Radius.circular(24),
        ), // BorderRadius.only // BoxDecoration
        padding: const EdgeInsets.all(24),
        child: Column(
          mainAxisSize: MainAxisSize.min,
          children: [
            Text(
              'Insert Message',
              style: AppText.header1,
            ), // Text
            SizedBox(
              height: 16,
            ), // SizedBox
            AppTextField(hint: 'Message...'),
            SizedBox(
              height: 16,
            ), // SizedBox
            Text(

```

Giving transparent color to modal to see radius

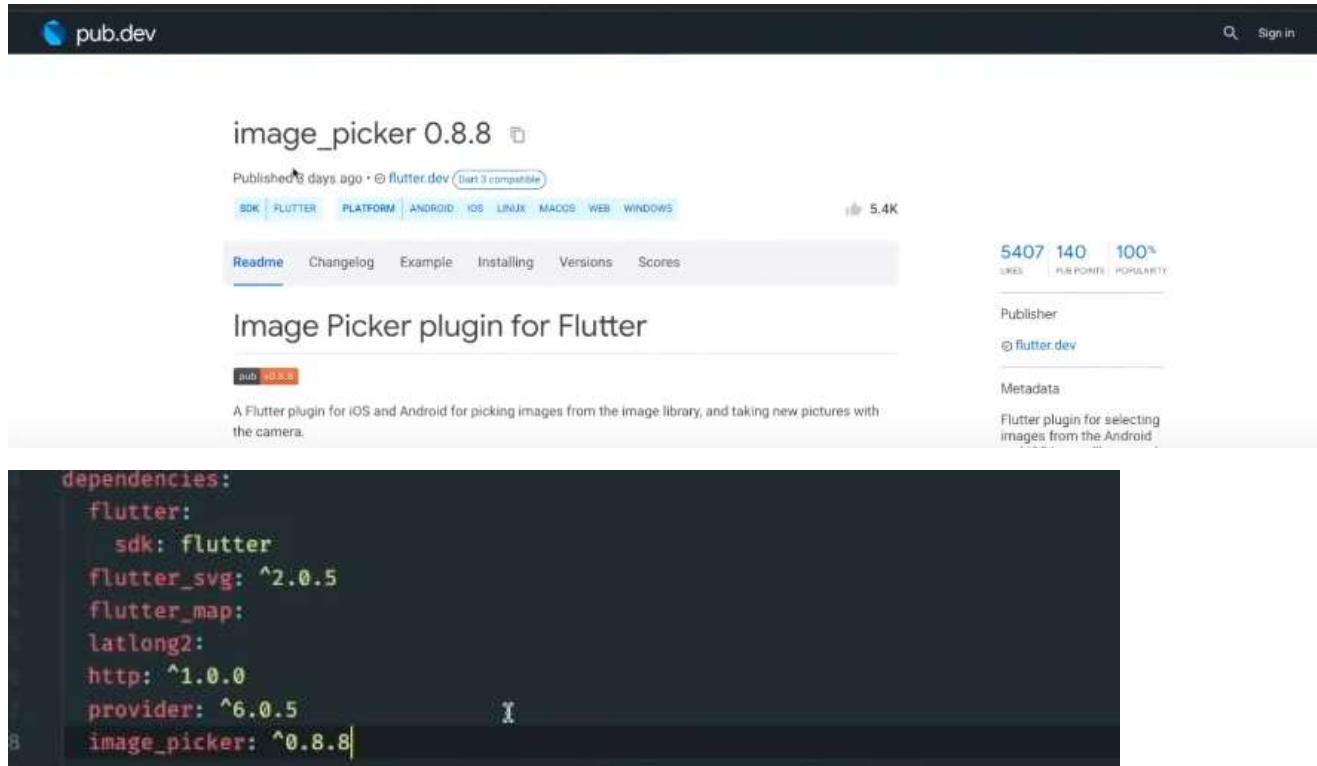


A screenshot of the same Flutter application, but the modal dialog now has a transparent background. This allows the rounded corners of the modal to be visible against the dark blue background of the main screen. The modal content and layout remain the same as in the previous screenshot.

```
class _ MainPageState extends State<MainPage> {
  Menus currentIndex = Menus.home;
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      extendBody: true,
      body: pages[currentIndex.index],
      bottomNavigationBar: MyBottomNavigation(
        currentIndex: currentIndex,
        onTap: (value) {
          if (value == Menus.add) {
            showModalBottomSheet(
              context: context,
              isScrollControlled: true,
              backgroundColor: Colors.transparent,
              builder: (context) {
                return NewPostModal();
              },
            );
          }
        },
        setState(() {

```

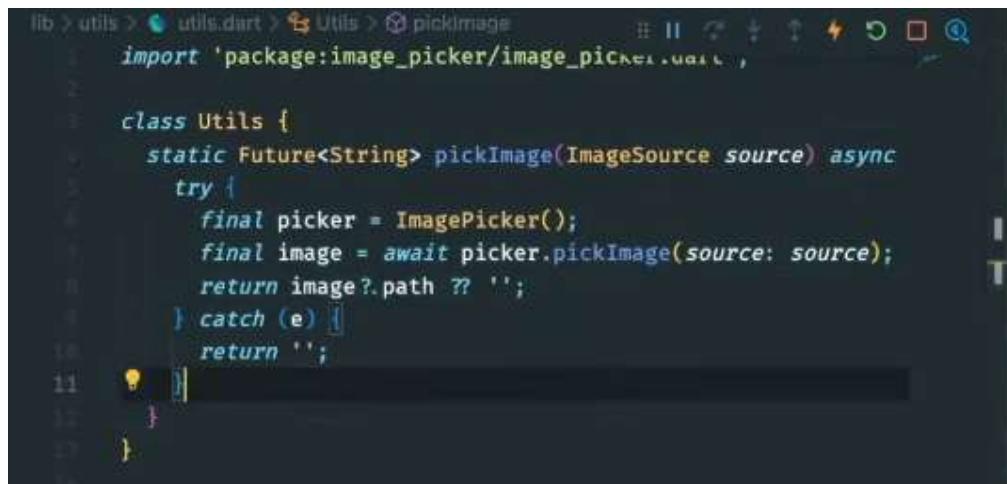
- **Implementing upload from gallery and camera**



The screenshot shows the pub.dev page for the `image_picker` package version 0.8.8. The page includes the package name, a download count of 5.4K, and links to Readme, Changelog, Example, Installing, Versions, Scores, and Metadata. It also shows the publisher as flutter.dev and a brief description of the plugin.

```
dependencies:  
  flutter:  
    sdk: flutter  
  flutter_svg: ^2.0.5  
  flutter_map:  
  latlong2:  
  http: ^1.0.0  
  provider: ^6.0.5  
  image_picker: ^0.8.8|
```

utils



```
lib > utils > utils.dart > Utils > pickImage  
import 'package:image_picker/image_picker.dart',  
  
class Utils {  
  static Future<String> pickImage(ImageSource source) async  
  try {  
    final picker = ImagePicker();  
    final image = await picker.pickImage(source: source);  
    return image?.path ?? '';  
  } catch (e) {  
    return '';  
  }  
}
```

```
lib> provider > post_provider.dart > PostProvider > 
import 'package:image_picker/image_picker.dart';

class PostProvider extends ChangeNotifier {
  final List<Post> list = [];
  String message = '';

  getPost() async {
    list.clear();
    list.addAll(await GetPostService().call());
    notifyListeners();
  }

  Future<void> createPost(String token) async {
    await CreatePostService(message, null, token).call();
    message = '';
    getPost();
  }

  pickImage(ImageSource source) {
    try {
      final path = Utils.pickImage(source);
      print(path);
    } catch (e) {
      print(e);
    }
  }
}
```

Restart debugging

Update setting

iOS

Starting with version 0.8.1 the iOS implementation uses PHPicker to pick (multiple) images on iOS 14 or higher. As a result of implementing PHPicker it becomes impossible to pick HEIC Images on the iOS simulator in iOS 14+. This is a known issue. Please test this on a real device, or test with non-HEIC images until Apple solves this issue. [63426347 - Apple known issue](#)

Add the following keys to your `Info.plist` file, located in `<project root>/ios/Runner/Info.plist`:

- `NSPhotoLibraryUsageDescription` - describe why your app needs permission for the photo library. This is called *Privacy - Photo Library Usage Description* in the visual editor.
 - This permission will not be requested if you always pass `false` for `requestFullMetadata`, but App Store policy requires including the plist entry.
- `NSCameraUsageDescription` - describe why your app needs access to the camera. This is called *Privacy - Camera Usage Description* in the visual editor.
- `NSMicrophoneUsageDescription` - describe why your app needs access to the microphone, if you intend to record videos. This is called *Privacy - Microphone Usage Description* in the visual editor.

Android

Starting with version 0.8.1 the Android implementation support to pick (multiple) images on Android 4.3 or higher.

No configuration required - the plugin should work out of the box. It is however highly recommended to prepare for Android killing the application when low on memory. How to prepare for this is discussed in the [Handling MainActivity destruction on Android](#) section.

It is no longer required to add `android:requestLegacyExternalStorage="true"` as an attribute to the `<application>` tag in `AndroidManifest.xml`, as `image_picker` has been updated to make use of scoped storage.

Handling MainActivity destruction

When under high memory pressure the Android system may kill the `MainActivity` of the application using the `MediaStore.ACTION_IMAGE_CAPTURE` intent. This means that while the intent is executing the source code

Info.plist

```
<true/>
  <key>NSPhotoLibraryUsageDescription</key>
  <string>Flutter course app need access to your gallery to
  <key>NSCameraUsageDescription</key>
  <string>Flutter course app need access to your camera to
</dict>
</plist>
```

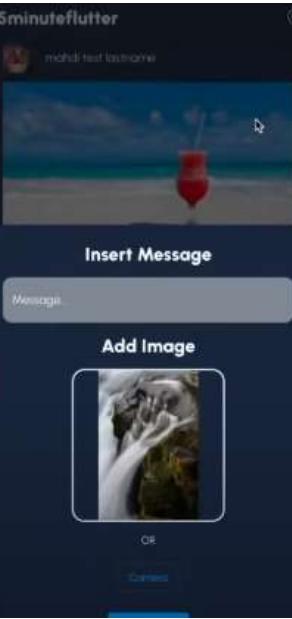
Using

```
.read<PostProvider>().pickImage(ImageSource.gallery);
```

```
SizedBox(
    height: 16,
),
GestureDetector(
    onTap: (){
        pickImage(ImageS
),
    child: Container(
        width: 200,
        height: 200,
        decoration: BoxDecoration(
            border: Border.all(
                color: Colors.white,
```

Showing image when path is not empty

```
Consumer<PostProvider>(
  builder: (context, value, child) => GestureDetector(
    onTap: () {
      context.read<PostProvider>().pickImage(Image
    },
    child: Container(
      width: 200,
      height: 200,
      decoration: BoxDecoration(
        border: Border.all(
          color: Colors.white,
          width: 2,
        ), // Border.all
        borderRadius: BorderRadius.all(Radius.circ
      ), // BoxDecoration
      child: value.imagePath == null
        ? Center(
            child: Text('Upload from gallery'),
          ) // Center
        : Image.file(File(value.imagePath!)),
    ), // Container
  ), // GestureDetector
), // Consumer
SizedBox()
```



• Image cropper flutter

The screenshot shows the pub.dev page for the `image_cropper` package. At the top, it says "image_cropper 5.0.0". Below that, it says "Published 4 days ago · Ⓜ hunghd.dev Dart 3 compatible". There are tabs for "SDK", "FLUTTER" (which is selected), "PLATFORM", "ANDROID", "IOS", and "WEB". To the right, it shows "1.6K" and "1692 LIKES 135 PUB POINTS 99% POPULARITY". Below the tabs, there are links for "Readme", "Changelog", "Example", "Installing", "Versions", and "Scores". On the right, there are sections for "Publisher" (@hunghd.dev) and "Metadata". The main content area shows the package's dependencies:

```
dependencies:
  flutter:
    sdk: flutter
  flutter_svg: ^2.0.5
  flutter_map:
  latlong2:
  http: ^1.0.0
  provider: ^6.0.5
  image_picker: ^0.8.8
  image_cropper: ^5.0.0
```

Util

The screenshot shows a Dart code editor with a file named `utils.dart`. The code defines a class `Utils` with two static methods: `pickImage` and `cropImage`.

```
lib > utils > Utils.dart > Utils > cropImage
import 'package:image_cropper/image_cropper.dart';
import 'package:image_picker/image_picker.dart';

class Utils {
  static Future<String> pickImage(ImageSource source) async {
    try {
      final picker = ImagePicker();
      final image = await picker.pickImage(source: source);
      return image?.path ?? '';
    } catch (e) {
      print(e);
      return '';
    }
  }

  static Future<CroppedFile?> cropImage(String path) async {
    final result = await ImageCropper().cropImage(sourcePath:
    15.     return result;
  }
}
```

Post_provider

```
lib > provider > post_provider.dart > PostProvider > 
  String? imagePath;

 getPost() async {
    list.clear();
    list.addAll(await GetPostService().call());
    notifyListeners();
  }

  Future<void> createPost(String token) async {
    await CreatePostService(message, null, token).call();
    message = '';
    getPost();
  }

  pickImage(ImageSource source) async {
    try {
      final path = await Utils.pickImage(source);
      await Utils.cropImage(path);
      // imagePath = path;
      // notifyListeners();
    } catch (e) {
      print(e);
    }
  }
}
```

After choosing image



➤ Fixing to display only aspect ratio

```
lib> utils > -utils.dart > Utils > cropImage
  final image = await picker.pickImage();
  return image?.path ?? '';
} catch (e) {
  print(e);
  return '';
}

static Future<CroppedFile?> cropImage(String path) async {
  final result = await ImageCropper().cropImage(
    sourcePath: path,
    aspectRatioPresets: [
      CropAspectRatioPreset.square,
    ],
    aspectRatio: CropAspectRatio(ratioX: 1, ratioY: 1),
  );
  return result;
}
```

Output



➤ Saving image path to the cropped image

```
lib > provider > post_provider.dart > PostProvider > PostProvider.dart
String? imagePath;

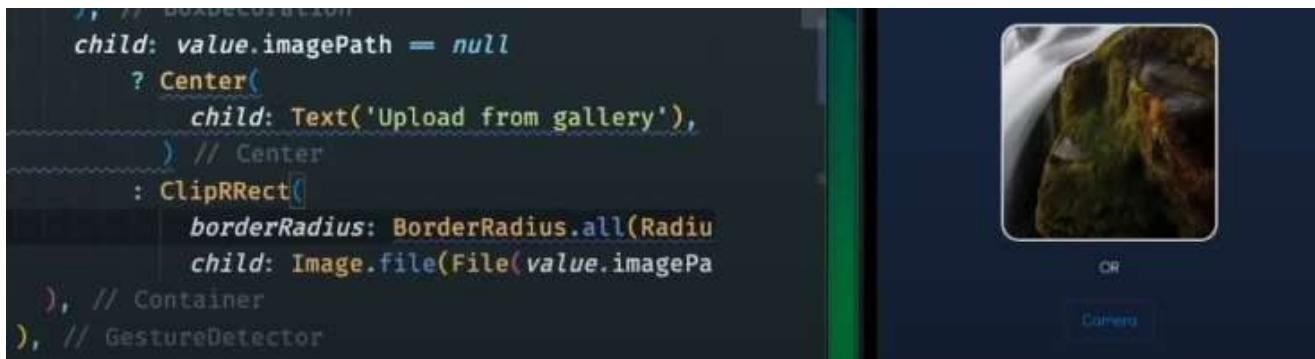
getPost() async {
    list.clear();
    list.addAll(await GetPostService().call());
    notifyListeners();
}

Future<void> createPost(String token) async {
    await CreatePostService(message, null, token).call();
    message = '';
    getPost();
}

pickImage(ImageSource source) async {
    try {
        final path = await Utils.pickImage(source);
        final croppedImage = await Utils.cropImage(path);

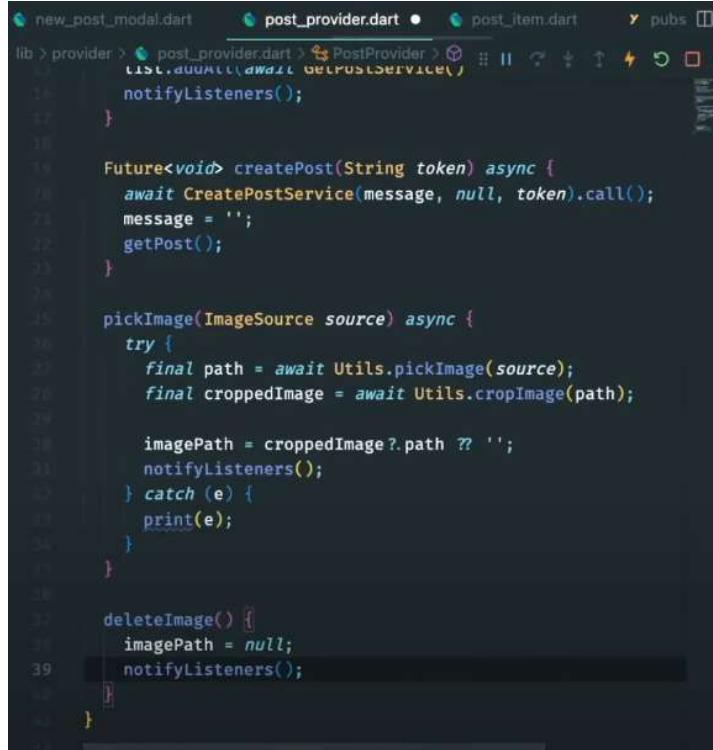
        imagePath = croppedImage?.path ?? '';
        notifyListeners();
    } catch (e) {
        print(e);
    }
}
```

clipRect: for Giving radius



```
child: value.imagePath == null
    ? Center(
        child: Text('Upload from gallery'),
    ) // Center
    : ClipRRect(
        borderRadius: BorderRadius.all(Radius.circular(10)),
        child: Image.file(File(value.imagePath)),
    ), // Container
), // GestureDetector
```

Removing image after adding



```
new_post_modal.dart post_provider.dart post_item.dart pubspec.yaml
lib > provider > post_provider.dart > PostProvider > PostProvider()
  LIST_OF_DART_FILES getPostService();
    notifyListeners();
}

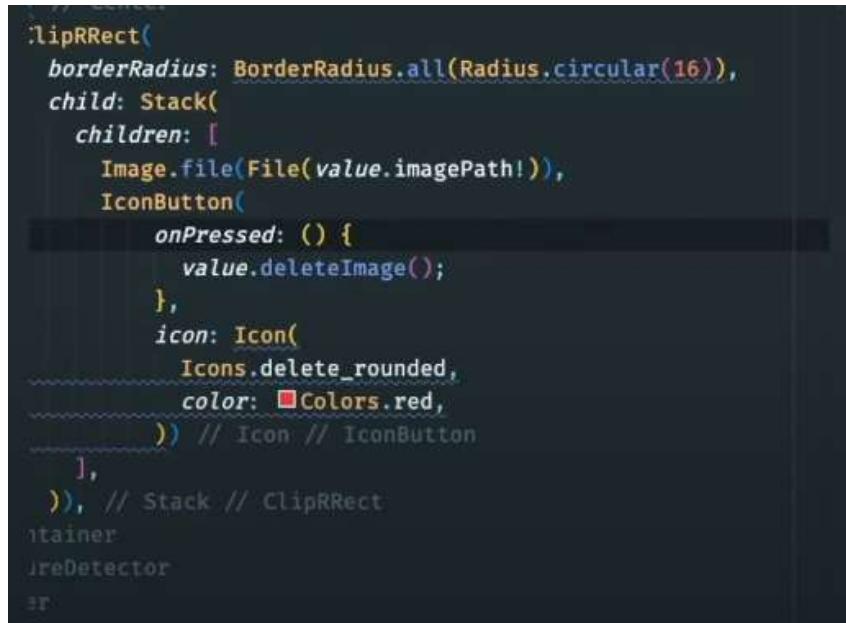
Future<void> createPost(String token) async {
  await CreatePostService(message, null, token).call();
  message = '';
  getPost();
}

pickImage(ImageSource source) async {
  try {
    final path = await Utils.pickImage(source);
    final croppedImage = await Utils.cropImage(path);

    imagePath = croppedImage?.path ?? '';
    notifyListeners();
  } catch (e) {
    print(e);
  }
}

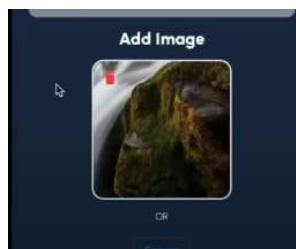
deleteImage() {
  imagePath = null;
  notifyListeners();
}
}
```

Using Stack, IconButton and value.deleteImage()



```
ClipRRect(
  borderRadius: BorderRadius.all(Radius.circular(16)),
  child: Stack(
    children: [
      Image.file(File(value.imagePath!)),
      IconButton(
        onPressed: () {
          value.deleteImage();
        },
        icon: Icon(
          Icons.delete_rounded,
          color: Colors.red,
        ) // Icon // IconButton
      ],
    ),
  ), // Stack // ClipRRect
),
GestureDetector
)
```

Output



Uploading file to server

```
Future<Map<String, dynamic>> upload(
  String apiUrl, String fieldName, String path,
  {String? token}) async {
try {
  final client = http.MultipartRequest('POST', _getV1Url(apiUrl));
  if (token != null) {
    client.headers.addAll({'Authorization': 'Bearer $token'});
  }
  client.files.add(await http.MultipartFile.fromPath(fieldName, path,
    contentType: parser.MediaType('image', 'jpeg')));
}

return _handleResponse(
  await http.Response.fromStream(await client.send()));
} catch (e) {
  throw Exception(e);
}
}
```

Upload_service.dart file

```
lib > data > service > upload_service.dart > UploadService
import 'package:flutter_full_course/data/service/base_service.dart';

class UploadService extends ServiceBase<String> {
  final String path;

  UploadService(this.path);

  @override
  Future<String> call() async {
    final result = await upload('upload', 'photo', path);
    return result['data'][0]['url'];
  }
}
```

Postman api

The screenshot shows a Postman request configuration and its resulting JSON response.

Request Configuration:

- Method: POST
- URL: {{baseUrl}}/upload
- Body tab selected
- File type: form-data
- Key: photo
- Value: drink-84533_640.jpg

Response:

- Status: 200 OK (44 ms, 31)
- Body (Pretty):

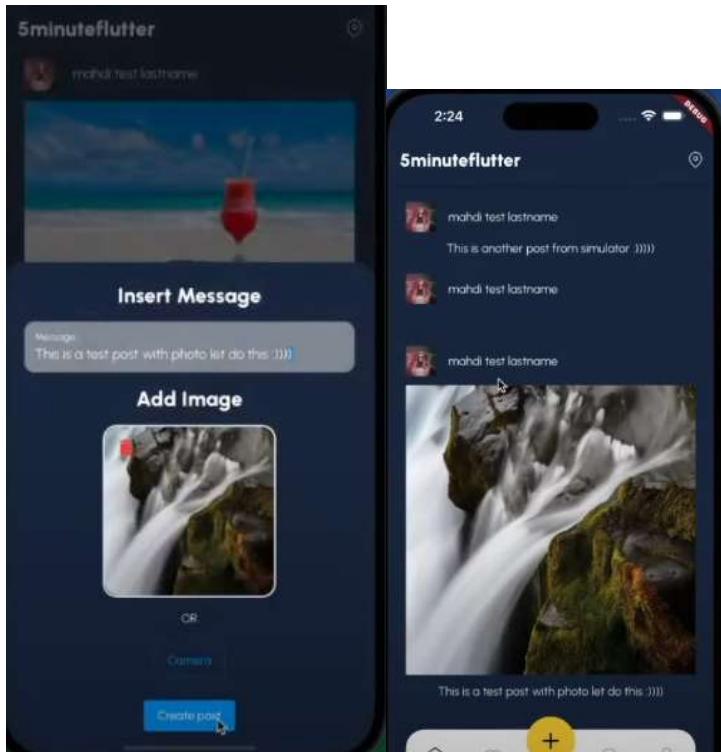
```
1:   "message": "success",
2:   "data": [
3:     {
4:       "url": "/images/drink-84533_640.jpg"
5:     }
6:   ]
7: }
```

Using it inside the api

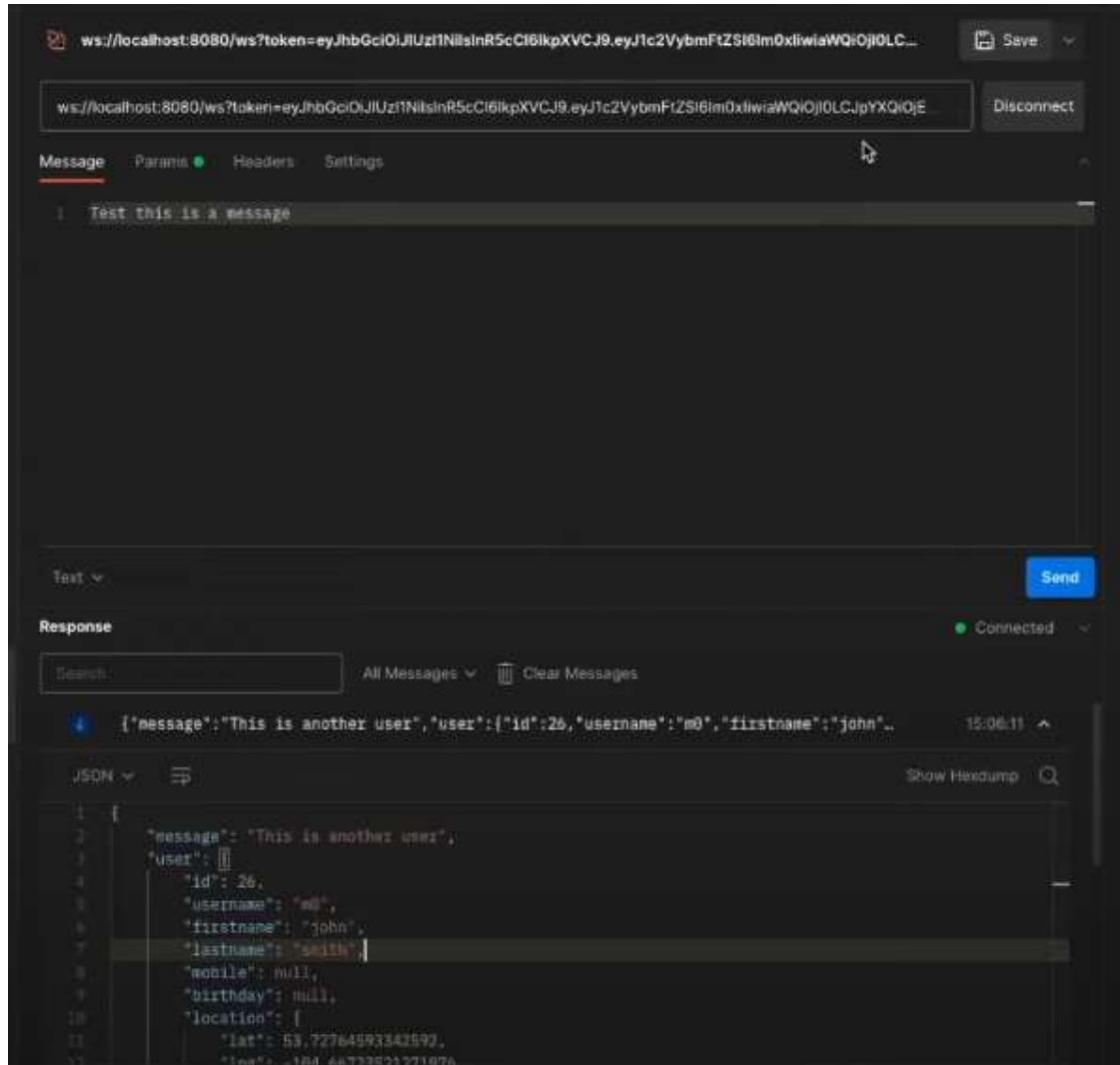
```
lib > provider > post_provider.dart > PostProvider > 
13
14     getPost() async {
15         list.clear();
16         list.addAll(await GetPostService().call());
17         notifyListeners();
18     }
19
20     Future<void> createPost(String token) async {
21         String? image;
22         if (imagePath != null) {
23             image = await upload();
24         }
25         await CreatePostService(message, image, token).call();
26         message = '';
27         imagePath = null;
28         getPost();
29     }
30
31     Future<String> upload() async {
32         return await UploadService(imagePath!).call();
33     }
34
35     pickImage(ImageSource source) async {
36         try {
37             final path = await Utils.pickImage(source);
38             final croppedImage = await Utils.cropImage(path);
39         }
40     }

```

Output working to upload



• Websockets



```
import 'package:web_socket_channel/web_socket_channel.dart';
import 'package:web_socket_channel/status.dart' as status;

main() async {
  final wsUrl = Uri.parse('ws://localhost:1234');
  var channel = WebSocketChannel.connect(wsUrl);

  channel.stream.listen((message) {
    channel.sink.add('received!');
    channel.sink.close(status.goingAway);
  });
}
```

```
dependencies:
  flutter:
    sdk: flutter
  flutter_svg: ^2.0.5
  flutter_map:
  latlong2:
  http: ^1.0.0
  provider: ^6.0.5
  image_picker: ^0.8.8
  image_cropper: ^5.0.0
  web_socket_channel: ^2.4.0|
```

```
lib> data> model> chat.dart> Chat> Chat.fromJson
import 'package:flutter_full_course/data/model/user.dart';

class Chat {
    final String message;
    final User user;

    Chat(this.message, this.user);

    factory Chat.fromJson(Map<String, dynamic> json) => Chat(
        json['message'],
        User.fromJson(
            json['user'],
        ), // User.fromJson
    ); // Chat
}
```

app_repo.dart

```
app_text_field.dart  post.dart  app_routes.dart  app_repo.dart

lib> provider> app_repo.dart> AppRepo
class AppRepo extends ChangeNotifier {
    String? _token;
    User? user;
    WebSocketChannel? _channel;
    List<Chat> chats = [];

    set token(String? value) {
        _token = value;
        _connectSocket();
    }

    String? get token => _token;

    _connectSocket() {
        final wsUrl = Uri.parse('ws://localhost:8080/ws?token=$_token');
        _channel = WebSocketChannel.connect(wsUrl);
        _channel?.stream.listen((message) {
            print(message);
            chats.add(Chat.fromJson(jsonDecode(message)));
            notifyListeners();
        });
    }
}
```

```
lib>pages>chat_page.dart>ChatPage>build import 'package:flutter/material.dart';
import 'package:flutter_full_course/components/chat_me_item.dart';
import 'package:flutter_full_course/components/toolbar.dart';
import 'package:flutter_full_course/provider/app_repo.dart';
import 'package:provider/provider.dart';

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

  @override
  Widget build(BuildContext context) {
    final provider = context.watch<AppRepo>();
    return Scaffold(
      appBar: Toolbar(title: 'Chat'),
      body: ListView.builder(
        itemBuilder: (context, index) {
          return ChatMeItem(chat: provider.chats[index]);
        },
        itemCount: provider.chats.length,
      ), // ListView.builder
    ); // Scaffold
  }
}
```

And

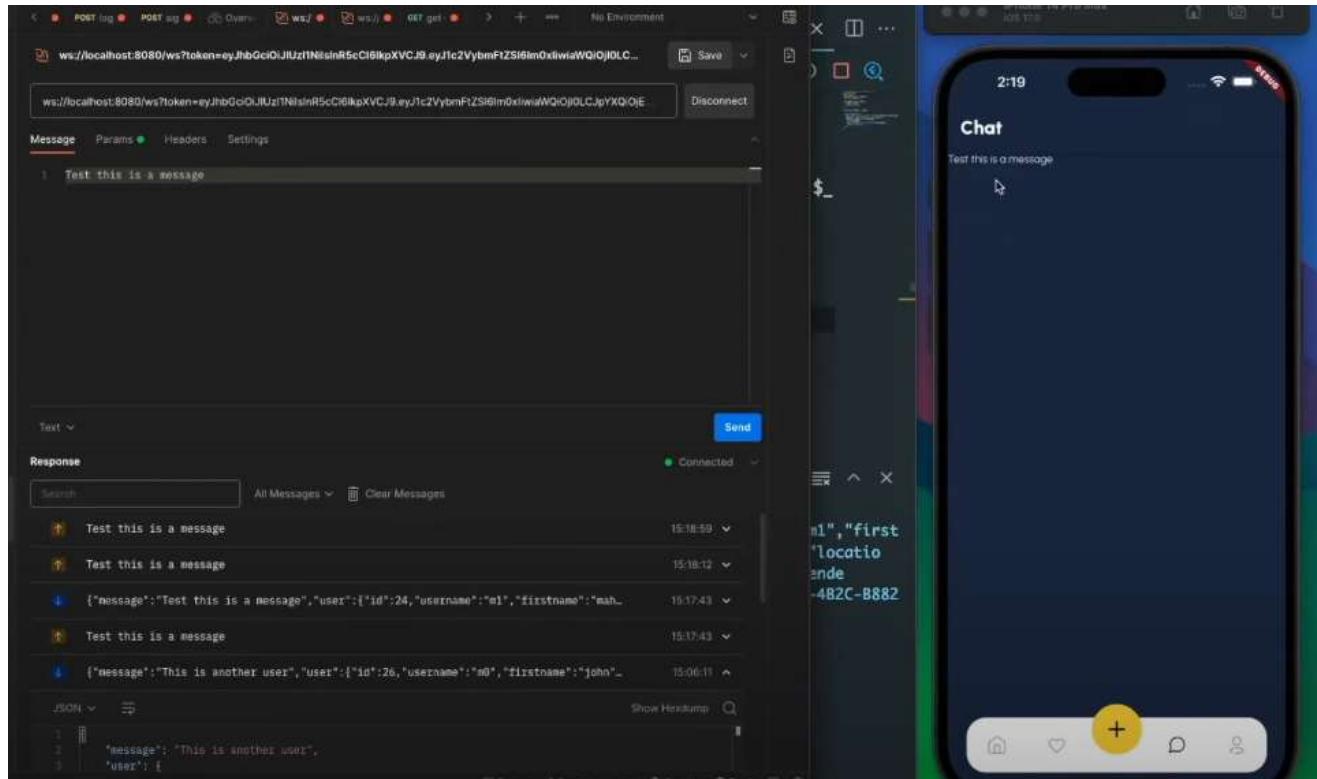
From

```
main_page.dart 2 × new_post_modal.dart post_provider.dart ... lib > pages > main_page.dart > _MainPageState > Scaffold > pages
43     );
44   }
45
46   final pages = [
47     HomePage(),
48     Center(
49       child: Text('Favorite'),
50     ),
51     Center(
52       child: Text('Add Post'),
53     ),
54     Center(
55       child: Text('Messages'),
56     ),
57     ProfilePage()
58   ];
59 }
```

To

```
main_page.dart — flutter_full_course
main_page.dart 2 × new_post_modal.dart post_provider.dart ...
lib > pages > main_page.dart > _MainPageState > Scaffold > pages
43     );
44   }
45
46   final pages = [
47     HomePage(),
48     Center(
49       child: Text('Favorite'),
50     ),
51     Center(
52       child: Text('Add Post'),
53     ),
54     ChatPage({Key? key}),
55     ProfilePage()
56   ];
57 }
```

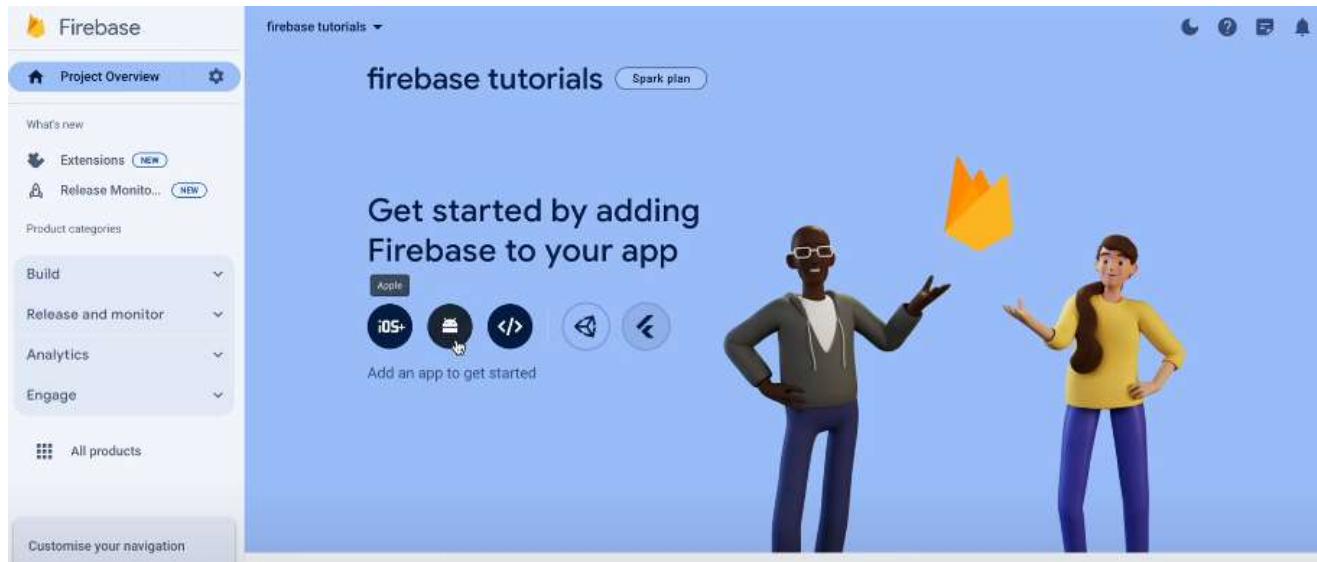
Output: from app_repo



```
PROBLEMS 193 OUTPUT DEBUG CONSOLE ... Filter (e.g. text, !excl...   
n": {"lat": 53.72764593342592, "lng": -104.66723521271976, "name": "Canada"}, "gender": "", "visibleGender": false, "avatar": "/images/image_cropper_C9A08305-DB60-4B2C-B882-83C745D31E4A-97175-00000596FA321C89.jpg"}  
[VERBOSE-2:dart_vm_initializer.cc(41)] Unhandled Exception: type 'String' is not a subtype of type 'Map<String, dynamic>'  
#0      AppRepo._connectSocket.<anonymous closure> app_repo.dart:2  
#1      _RootZone.runUnaryGuarded (dart:async/zone.dart:1594:10)  
#2      _BufferingStreamSubscription._sendData (dart:async/stream_impl.dart:339:11)  
#3      _BufferingStreamSubscription.add (dart:async/stream_impl.dart:273:7)
```

The remaining can be learned by chatgpt

- **Firebase and Flutter**



- **For android**

The screenshot shows the Android Studio interface with the project navigation bar on the left and the code editor on the right.

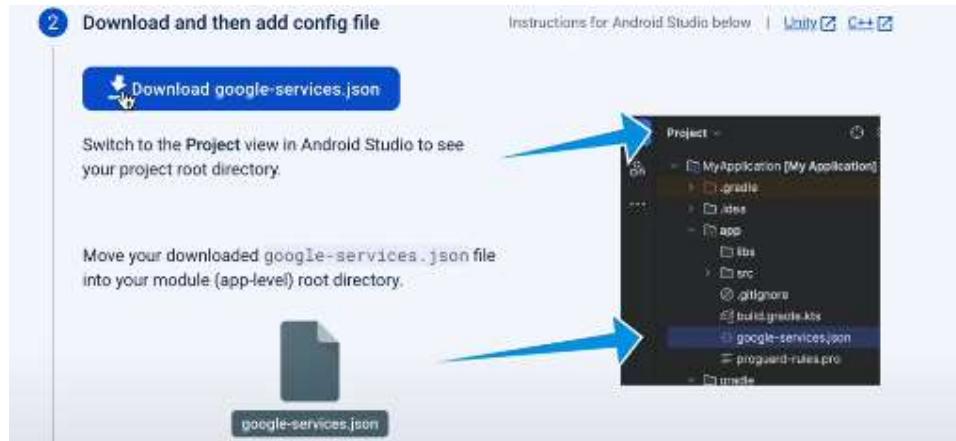
Project Structure:

- firebase_tutorials
- |- .dart_tool
- |- .idea
- |- android [firebase_tutorials_android]
 - |- .gradle
 - |- app
 - |- src
 - |- build.gradle
 - |- gradle
 - |- .gitignore
 - |- build.gradle
 - |- firebase_tutorials_android.iml
 - |- gradle.properties
 - |- gradlew
 - |- gradlew.bat
 - |- local.properties
 - |- settings.gradle
- |- build
- |- ios
- |- lib
- |- linux
- |- macos
- |- test
- |- web
- |- windows
 - |- .gitignore
 - |- .metadata
 - |- analysis_options.yaml
 - |- firebase_tutorials.iml
 - |- pubspec.lock
 - |- pubspec.yaml
 - |- README.md
- |- External Libraries
- |- Scratches and Consoles

x Add Firebase to your Android app

- 2 Download and then add config file
 - 3 Add Firebase SDK
 - 4 Next steps

Adding google-services.json



Adding required code



Added in setting.gradle

The screenshot shows the settings.gradle file in the Android Studio editor. It highlights the addition of the dependency "id 'com.google.gms.google-services' version '4.4.1' apply false" under the "plugins" section. The "Open for Editing in Android Studio" button is visible at the top right.

```
pluginManagement {
    def flutterSdkPath : Closure = {
        def properties = new Properties()
        file("local.properties").withInputStream { properties.load(it) }
        def flutterSdkPathString = properties.getProperty("flutter.sdk")
        assert flutterSdkPath != null, "Flutter SDK not set in local.properties"
        return flutterSdkPath
    }
    settings.ext.flutterSdkPath = flutterSdkPath()

    includeBuild('${settings.ext.flutterSdkPath}/packages/flutter_tools/
repositories {
    google()
    mavenCentral()
    gradlePluginPortal()
}

plugins {
    id "dev.flutter.flutter-plugin-loader" version "1.0.0"
    id "com.android.application" version "7.3.0" apply false
    id "org.jetbrains.kotlin.android" version "1.7.10" apply false
    id 'com.google.gms.google-services' version '4.4.1' apply false
}
include ":app"
```

2. Then, in your module (app-level) build.gradle file, add both the google-services plug-in and any Firebase SDKs that you want to use in your app:

Module (app-level) Gradle file (<project>/<app-module>/build.gradle):

```
plugins {
    id 'com.android.application'
    // Add the Google services Gradle plugin
    id 'com.google.gms.google-services'
    ...
}

dependencies {
    // Import the Firebase BoM
    implementation platform('com.google.firebase:firebase-bom:32.8.0')

    // TODO: Add the dependencies for Firebase products you want to use
    // When using the BoM, don't specify versions in Firebase dependencies
    implementation 'com.google.firebaseio:firebase-analytics'

    // Add the dependencies for any other desired Firebase products
    // https://firebase.google.com/docs/android/setup#available-libraries
}
```

By using the Firebase Android BoM, your app will always use compatible Firebase library versions. [Learn more](#)

3. After adding the plug-in and the desired SDKs, sync your Android project with the Gradle files.

Previous

Next

* Get more help at <https://help.gradle.org>.

BUILD FAILED in 1m 55s

Flutter Fix

```
[!] Your project requires a newer version of the Kotlin Gradle plugin.
Find the latest version on https://kotlinlang.org/docs/releases.html#release-details, then
update /Users/abubakar/Desktop/firebase_tutorials/android/build.gradle:
ext.kotlin_version = '<latest-version>'
```

Error: Gradle task assembleDebug failed with exit code 1

← → ⌂ ↻ kotlinlang.org/docs/releases.html#release-details

Gmail YouTube All apps Videos Alarm Bard AI Chat pub.dev Canva Analyst - Healer videos Alarm APIs Be Max APIs okapp-mobile-ui ...

Let's put the fun back into functional programming! Join here →

Kotlin v1.9.23

Home Get started Take Kotlin tour

Kotlin overview What's new in Kotlin Releases and roadmap

Kotlin releases

Kotlin roadmap

Basics Concepts Multipointer development

Release details

The following table lists details of the latest Kotlin releases.

You can also use preview versions of Kotlin.

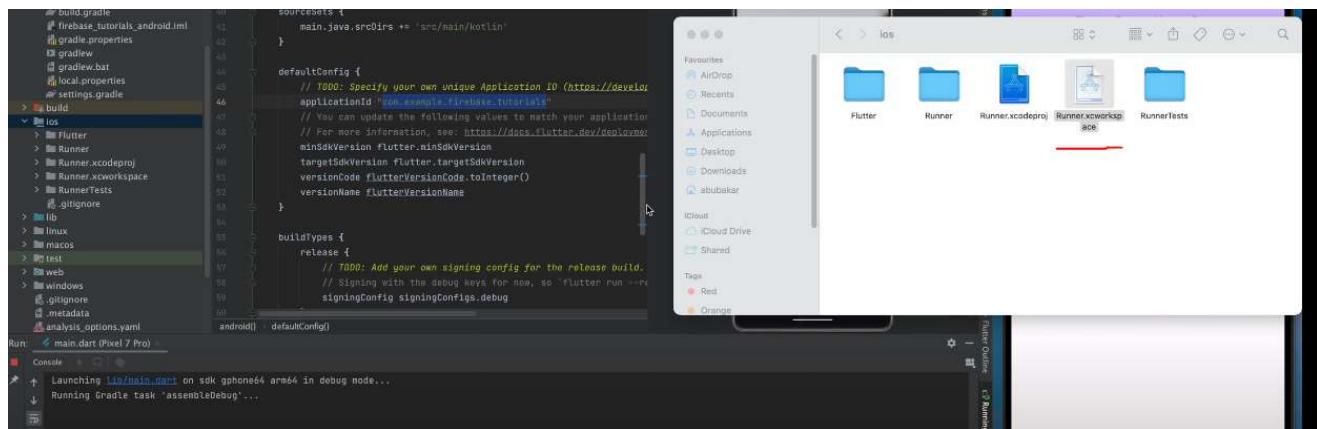
Build info	Build highlights
1.9.23 Released: March 7, 2024 Release on GitHub ↗	A bug fix release for Kotlin 1.9.20, 1.9.21, and 1.9.22. Learn more about Kotlin 1.9.20 in What's new in Kotlin 1.9.20.

plugins {
 id "dev.flutter.flutter-plugin-loader" version "1.0.0"
 id 'com.android.application' version "7.3.0" apply false
 id "org.jetbrains.kotlin.android" version "1.9.23" apply false
 id 'com.google.gms.google-services' version '4.4.1' apply false
}

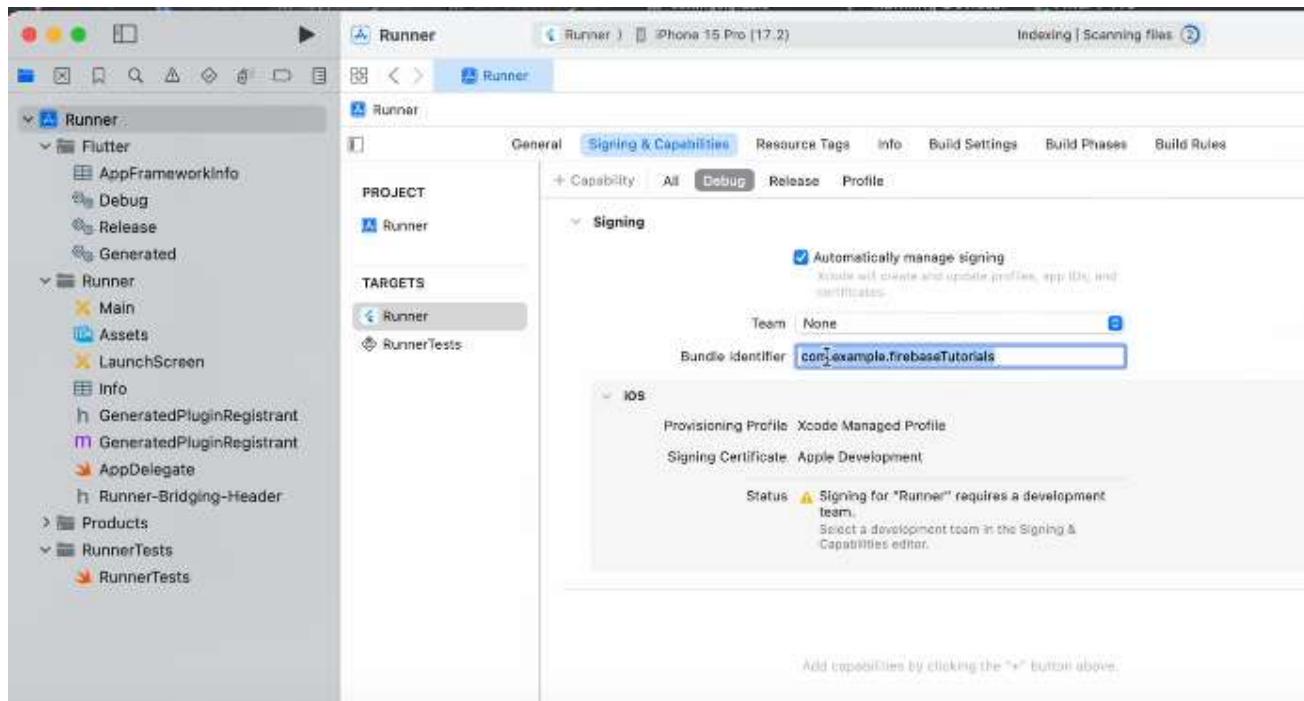
include ':app'

For IOS

Open file in xcode



Runner>signin>bundleId



× Add Firebase to your Apple app

1 Register app

Apple bundle ID [?](#)

com.example.firebaseio.tutorials

App nickname (optional) [?](#)

My Apple app

If specified, the app nickname will be used throughout the Firebase console to represent this app. Nicknames aren't visible to users.

App Store ID (optional) [?](#)

123456789

[Register app](#)

2 Download config file

Instructions for Xcode below | [Unity](#) [C++](#)

[Download GoogleService-Info.plist](#)

Move the GoogleService-Info.plist file that you just downloaded into the root of your Xcode project and add it to all targets.



[Next](#)

Add Firebase SDK

1 Add initialisation code

To connect Firebase when your app starts up, add the initialisation code below to your app's main entry point.

SwiftUI Swift Objective-C

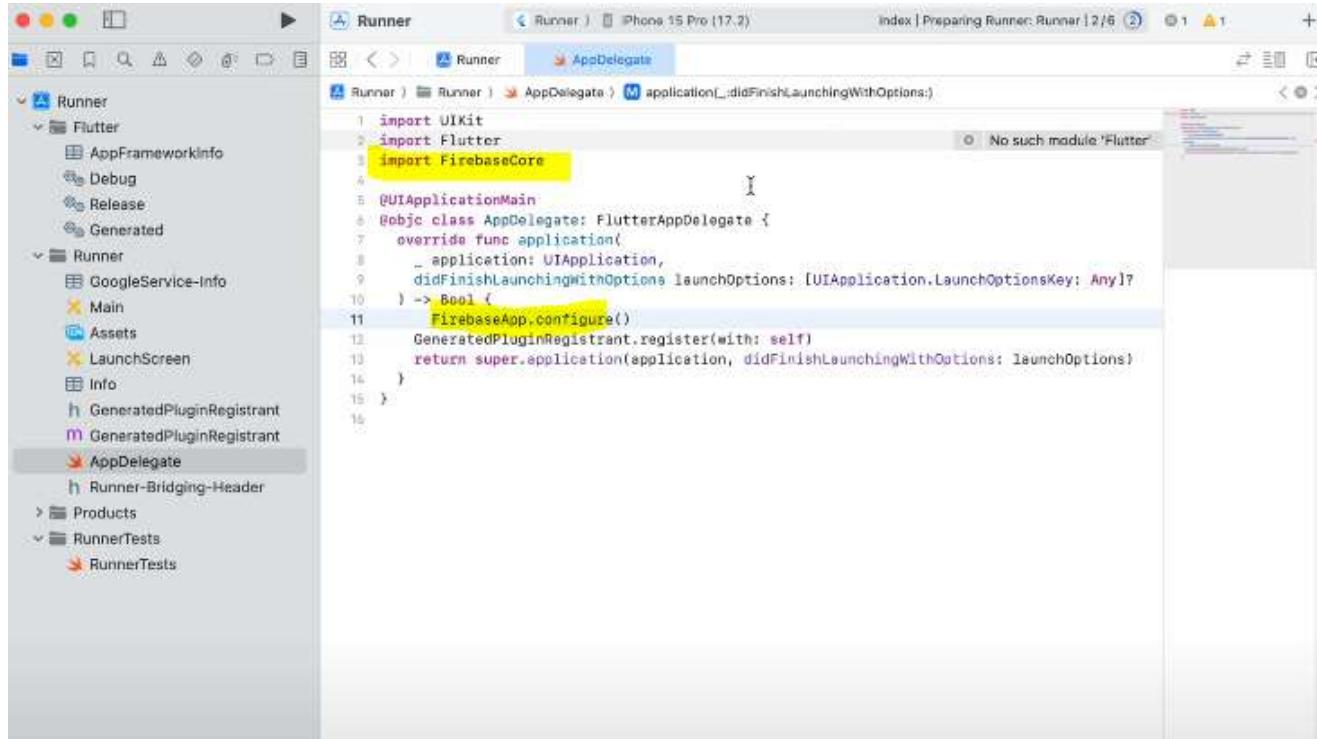
```
import UIKit
import FirebaseCore

UIApplicationMain
class AppDelegate: UIResponder, UIApplicationDelegate {

    var window: UIWindow?

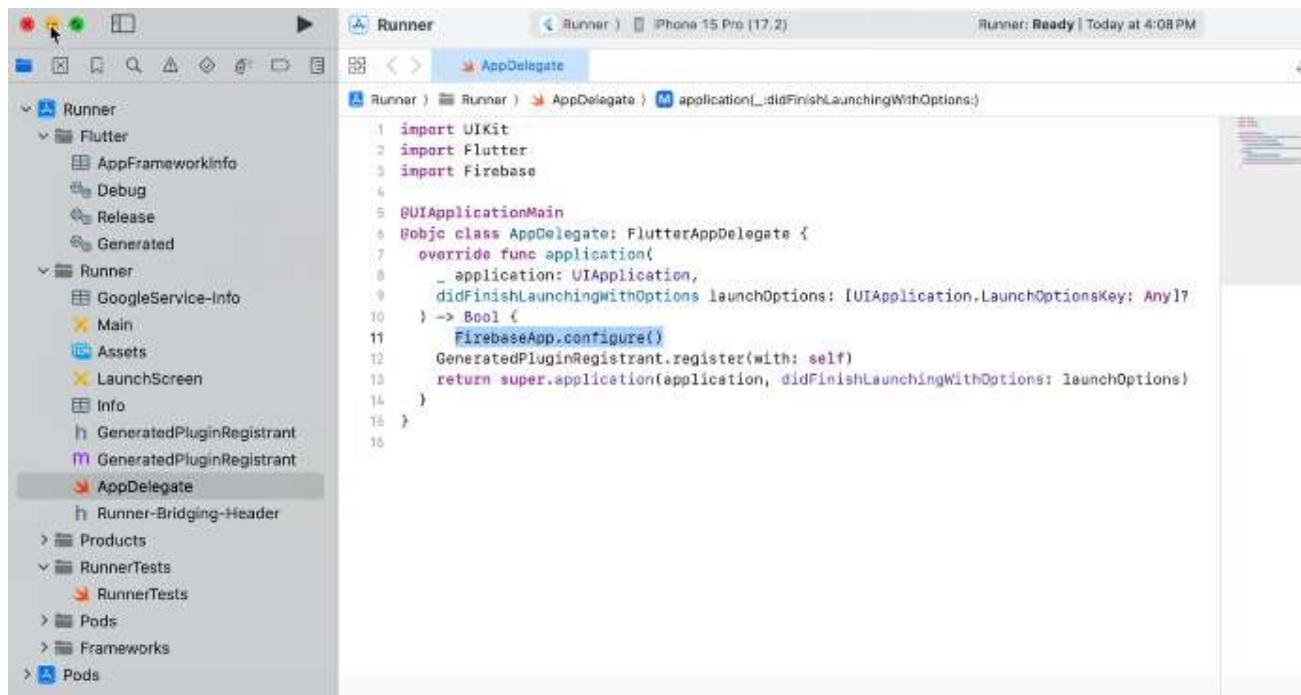
    func application(_ application: UIApplication,
                    didFinishLaunchingWithOptions launchOptions:
                        [UIApplicationLaunchOptionsKey: Any]?) -> Bool {
        FirebaseApp.configure()
        return true
    }
}
```

Xcode>appdelegate>



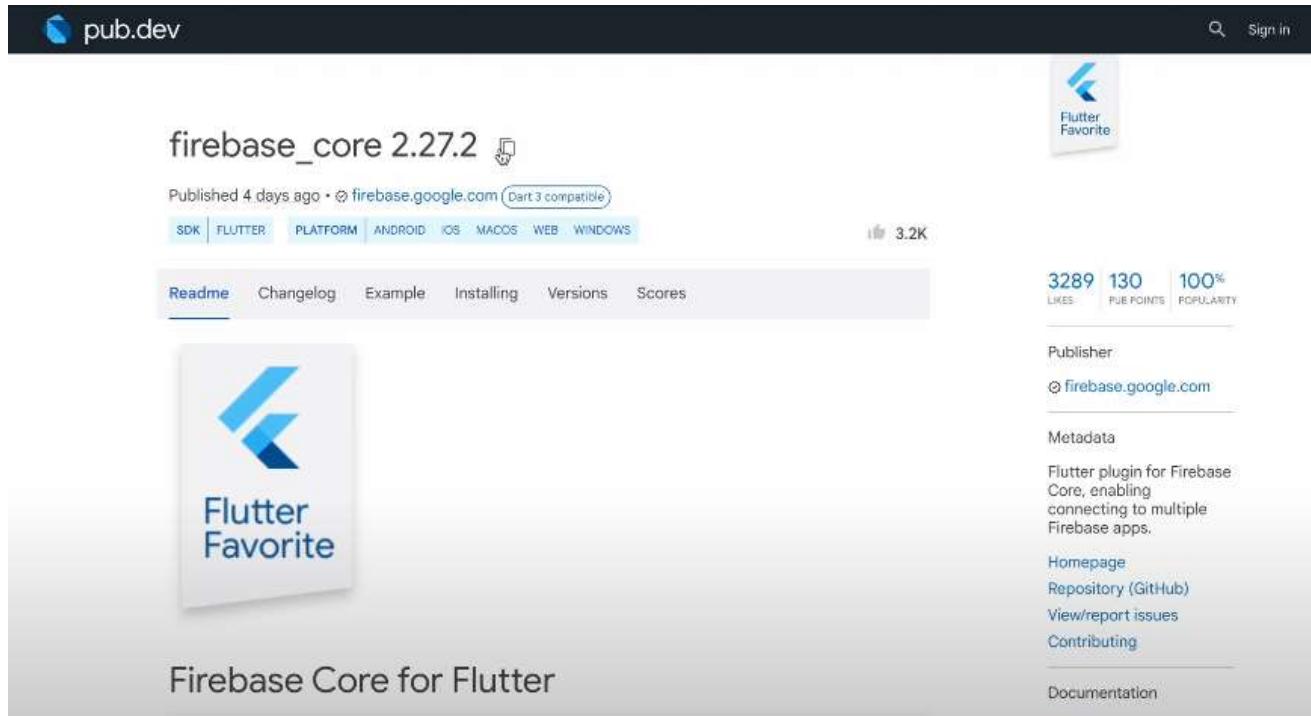
```
import UIKit
import Flutter
import FirebaseCore
@UIApplicationMain
class AppDelegate: FlutterAppDelegate {
    override func application(
        _ application: UIApplication,
        didFinishLaunchingWithOptions launchOptions: [UIApplication.LaunchOptionsKey: Any]?
    ) -> Bool {
        FirebaseApp.configure()
        GeneratedPluginRegistrant.register(with: self)
        return super.application(application, didFinishLaunchingWithOptions: launchOptions)
    }
}
```

Or



```
import UIKit
import Flutter
import Firebase
@UIApplicationMain
class AppDelegate: FlutterAppDelegate {
    override func application(
        _ application: UIApplication,
        didFinishLaunchingWithOptions launchOptions: [UIApplication.LaunchOptionsKey: Any]?
    ) -> Bool {
        FirebaseApp.configure()
        GeneratedPluginRegistrant.register(with: self)
        return super.application(application, didFinishLaunchingWithOptions: launchOptions)
    }
}
```

Installing



The screenshot shows the pub.dev website for the `firebase_core` package. The page header includes the pub.dev logo, a search bar, and a sign-in link. The main content area displays the package details:

- Name:** firebase_core 2.27.2
- Published:** 4 days ago • firebase.google.com (Dart 3 compatible)
- Platforms:** SDK, FLUTTER, PLATFORM, ANDROID, IOS, MACOS, WEB, WINDOWS
- Downloads:** 3.2K
- Likes:** 3289 | **PUB POINTS:** 130 | **POPULARITY:** 100%

Below the header, there are tabs for Readme, Changelog, Example, Installing, Versions, and Scores. A large image of the Flutter logo is displayed, along with the text "Flutter Favorite".

The **Installing** tab is selected, showing the following code snippet from the `pubspec.yaml` file:

```
flutter
  sdk: flutter

  # The following adds the CupertinoIcons font to your application.
  # Use with the CupertinoIcons class for iOS style icons.
  cupertino_icons: ^1.0.6
  firebase_core: ^2.27.2

dev_dependencies:
  flutter_test:
    sdk: flutter

  # The "flutter_lints" package below contains a set of recommended lints to
  # encourage good coding practices. The lint set provided by the package is
  # activated in the "analysis_options.yaml" file located at the root of your
  # package. See that file for information about deactivating specific lint
  # rules and activating additional ones.
  flutter_lints: ^3.0.0

  # For information on the generic Dart part of this file, see the
  # following page: https://dart.dev/tools/pub/pubspec
```

At the bottom of the code editor, it says "Document 1/1 dependencies: firebase_core: ^2.27.2".

example/lib/main.dart

```
// Copyright 2019 The Chromium Authors. All rights reserved.
// Use of this source code is governed by a BSD-style license that can be
// found in the LICENSE file.

import 'dart:async';
import 'package:flutter/foundation.dart';
import 'package:flutter/material.dart';
import 'package:firebase_core/firebase_core.dart';
import 'firebase_options.dart';

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

class MyApp extends StatelessWidget {
  const MyApp({Key? key}) : super(key: key);

  String get name => "foo";

  Future<void> initializeDefault() async {
    FirebaseApp app = await Firebase.initializeApp(
      options: DefaultFirebaseOptions.currentPlatform,
    );
    print('Initialized default app $app');
  }

  Future<void> initializeDefaultFromAndroidResource() async {
    if (defaultTargetPlatform != TargetPlatform.android || kIsWeb) {
      print('Not running on Android, skipping');
      return;
    }
    FirebaseApp app = await Firebase.initializeApp();
    print('Initialized default app $app from Android resource');
  }

  Future<void> initializeSecondary() async {

```

Main.dart *For manually info from google-services.json

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

void main() async {
  WidgetsFlutterBinding.ensureInitialized();

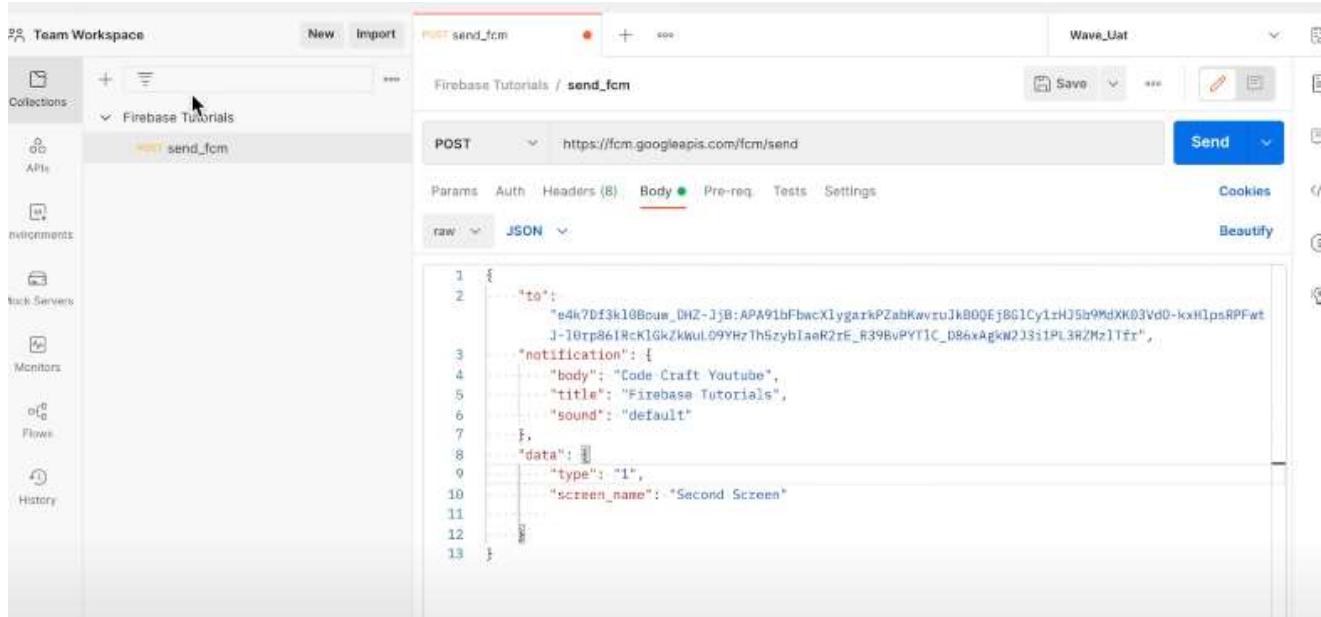
  if(Platform.isAndroid){
    await Firebase.initializeApp(
      options: const FirebaseOptions(
        apiKey: 'AIzaSyBPQJSS2noc-0g13H64MykngINHMG18auU',
        appId: '1:999951161:android:4da9e9d9920df852028888',
        messagingSenderId: '999951161',
        projectId: 'fir-tutorials-df86f',
      );
  }else{
    await Firebase.initializeApp();
  }

  runApp(const MyApp());
}

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

The screenshot shows the Firebase Project settings page. The top navigation bar includes the Firebase logo, a dropdown for 'firebase tutorials', and icons for notifications and account management. On the left, a sidebar lists 'Project Overview', 'Extensions (NEW)', 'Release Monito...', 'Product categories', 'Build', 'Release and monitor', 'Analytics', 'Engage', and 'All products'. Below this is a section for 'Customise your navigation' with a note about customising the console experience. The main content area is titled 'Project settings' and has tabs for General, Cloud Messaging (which is selected), Integration, Service accounts, Data privacy, and Users and permissions. Under the Cloud Messaging tab, there's a section for 'Firebase Cloud Messaging API (V1)' which is 'Enabled'. It includes a 'Sender ID' field containing '999095116161' and a 'Service account' link. A note says it's recommended for most use cases with a 'Learn more' link. Another section for 'Cloud Messaging API (Legacy)' is 'Disabled', with a note about migrating from legacy APIs by June 2024 and a 'Learn more' link. At the bottom, there's an 'Apple app configuration' section with tabs for 'Apple apps' (selected) and 'Firebase Cloud Messaging can use either an APNs authentication key or APNs certificate to connect'.

• Sending Notifications



Installation

```
# The following adds the Cupertino Icons font to your application.
# Use with the CupertinoIcons class for iOS style icons.
cupertino_icons: ^1.0.6
firebase_core: ^2.27.2
firebase_messaging: ^14.8.0
flutter_local_notifications: ^17.0.0

dev_dependencies:
  flutter_test:
    sdk: flutter

# The "flutter_lints" package below contains a set of recommended lints
# encourage good coding practices. The lint set provided by the package
# can be found at https://github.com/flutter/flutter_lints/blob/master/lint.yaml
```

Enable:multiDexEnabled

```
// You can update the following values to match your application needs.
// For more information, see: https://docs.flutter.dev/deployment/android
minSdkVersion flutter.minSdkVersion
targetSdkVersion flutter.targetSdkVersion
versionCode flutterVersionCode.toInt()
versionName flutterVersionName
multiDexEnabled true

buildTypes {
    release {
        // TODO: Add your own signing config for the release build.
        // Signing with the debug keys for now, so 'flutter run --release'
        android{ defaultConfig{
```

The screenshot shows the FlutterFire documentation page for messaging/notifications. The left sidebar lists various topics: Getting Started, Analytics, App Check, Authentication, Cloud Firestore, Cloud Firestore ODM, Cloud Functions, Cloud Messaging (which is selected), Overview, Usage, Apple Integration, Permissions, Notifications (selected), and Server Integration. The main content area is titled "iOS Configuration". It explains how to enable foreground notifications by calling the `setForegroundNotificationPresentationOptions` method. A code snippet is provided:

```
await FirebaseMessaging.instance.setForegroundNotificationPresentationOptions(
    alert: true, // Required to display a heads up notification
    badge: true,
    sound: true,
);
```

A "Copy" button is located next to the code. Below the code, a note says: "Set all values back to `false` to revert to the default functionality." To the right of the main content, there are several links: "Displaying notifications", "Via Firebase Console", "Via Admin SDKs", "Via REST", "Handling Interaction", "Advanced Usage", "Foreground Notifications", and "Background Fetch".

Make folder and file

The screenshot shows the code for the `firebase_messaging/example` package. The code is a Dart script that demonstrates how to handle messages. It includes comments explaining the steps to run the project and send messages. The code defines a `firebaseMessagingBackgroundHandler` function that handles background messages. It also creates an `AndroidNotificationChannel` for heads-up notifications.

```
/// 2. Run `meios bootstrap` in FlutterFire project.
/// 3. In your terminal, root to ./packages/firebase_messaging/firebase_messaging
/// 4. Run 'flutterfire configure' in the example/ directory to setup your app
/// 5. Open 'token_monitor.dart' and change 'apidKey' to yours.
/// 6. Run the app on an actual device for iOS, android is fine to run on an emulated device
/// 7. Use the following script to send a message to your device: scripts/sendMessage
///   you will need nodejs installed on your computer. Then the following:
///     a. Download a service account key (JSON file) from your Firebase console
///     b. Ensure your device/emulator is running, and run the FirebaseMessaging
///     c. Copy the token that is printed in the console and paste it here: ht
///     c. From your terminal, root to example/scripts directory & run 'npm install'
///     d. Run 'npm run send-message' in the example/scripts directory and you're good to go!
/// Note: Flutter API documentation for receiving messages: https://firebase.google.com/docs/flutter/messaging-background
/// Note: If you find your messages have stopped arriving, it is extremely likely that your app is too aggressive with their throttling policy.
///
/// To verify that your messages are being received, you ought to see a notification
/// Define a top-level named handler which background/terminated messages will
/// call. Be sure to annotate the handler with @pragma('vm:entry-point') abc
@pragma('vm:entry-point')
Future<void> _firebaseMessagingBackgroundHandler(RemoteMessage message) async {
  await Firebase.initializeApp(options: DefaultFirebaseOptions.currentPlatform);
  await setupFlutterNotifications();
  showFlutterNotification(message);
  // If you're going to use other Firebase services in the background, such as
  // make sure you call 'initializeApp' before using other Firebase services.
  print('Handling a background message ${message.messageId}');
}

/// Create a [AndroidNotificationChannel] for heads up notifications
late AndroidNotificationChannel channel;
```

```
← → ⌂ pub.dev/packages/firebase_messaging/example

if (notification != null && android != null && !kIsWeb) {
  flutterLocalNotificationsPlugin.show(
    notification.hashCode,
    notification.title,
    notification.body,
    notificationDetails(
      android: AndroidNotificationDetails(
        channelId,
        channel.name,
        channelDescription: channel.description,
        // TODO add a proper drawable resource to android, for now using
        // one that already exists in example app.
        icon: 'launch_background',
      ),
    ),
  );
}

// Initialize the [FlutterLocalNotificationsPlugin] package.
late FlutterLocalNotificationsPlugin flutterLocalNotificationsPlugin;

Future<void> main() async {
  WidgetsFlutterBinding.ensureInitialized();
  await Firebase.initializeApp(options: DefaultFirebaseOptions.currentPlatform);
  // Set the background messaging handler early on, as a named top-level function
  FirebaseMessaging.onBackgroundMessage(_firebaseMessagingBackgroundHandler);

  if (!kIsWeb) {
    await setupFlutterNotifications();
  }
  runApp(MessagingExampleApp());
}

// Entry point for the example application.
```

```
main.dart push_notification.dart pubspec.yaml

5
6
7   @pragma('vm:entry-point')
8   Future<void> _firebaseMessagingBackgroundHandler(RemoteMessage message) async {
9     if(Platform.isAndroid){
10       await Firebase.initializeApp(
11         options: const FirebaseOptions(
12           apiKey: 'AIzaSyBFQG3S2moc-Dg13H6HMykngINHMG1BauU',
13           appId: '1:999095116161:android:4da9e9d9920df852028880',
14           messagingSenderId: '999095116161',
15           projectId: 'fir-tutorials-df84f'),
16     );
17   }else{
18     await Firebase.initializeApp();
19   }
20   print('Handling a background message ${message.messageId}');

```

```
main.dart push_notification.dart pubspec.yaml

31   messagingSenderId: '999095116161',
32   projectId: 'fir-tutorials-df84f),
33 );
34 }else{
35   await Firebase.initializeApp();
36 }

38   FirebaseMessaging.onBackgroundMessage(_firebaseMessagingBackgroundHandler

39
40
41   runApp(const MyApp());
42 }
```



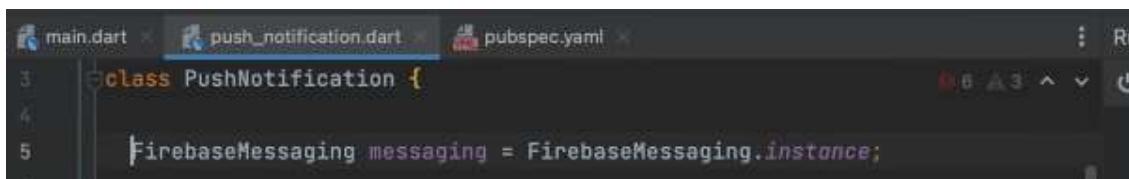
- Getting Started >
- Analytics >
- App Check >
- Authentication >
- Cloud Firestore >
- Cloud Firestore ODM >
- Cloud Functions >
- Cloud Messaging >
 - Overview
 - Usage
 - Apple Integration
 - Permissions
- Notifications
- Server Integration

FCM, via the `requestPermission` API:

```
FirebaseMessaging messaging = FirebaseMessaging.instance.copyWith  
NotificationSettings settings = await messaging.requestPermissions  
    alert: true,  
    announcement: false,  
    badge: true,  
    carPlay: false,  
    criticalAlert: false,  
    provisional: false,  
    sound: true,  
);  
  
if (settings.authorizationStatus == AuthorizationStatus.authorized) {  
    print('User granted permission');  
} else if (settings.authorizationStatus == AuthorizationStatus.provisional) {  
    print('User granted provisional permission');  
} else {  
    print('User declined or has not accepted permission');  
}
```

Requesting permission
Permission
Provisional authorization

On Apple based platforms, once a permission request has been handled by the user (authorized or denied), it is not possible to re-



A screenshot of an IDE showing a code editor with three tabs: main.dart, push_notification.dart, and pubspec.yaml. The push_notification.dart tab is active and contains the following code:

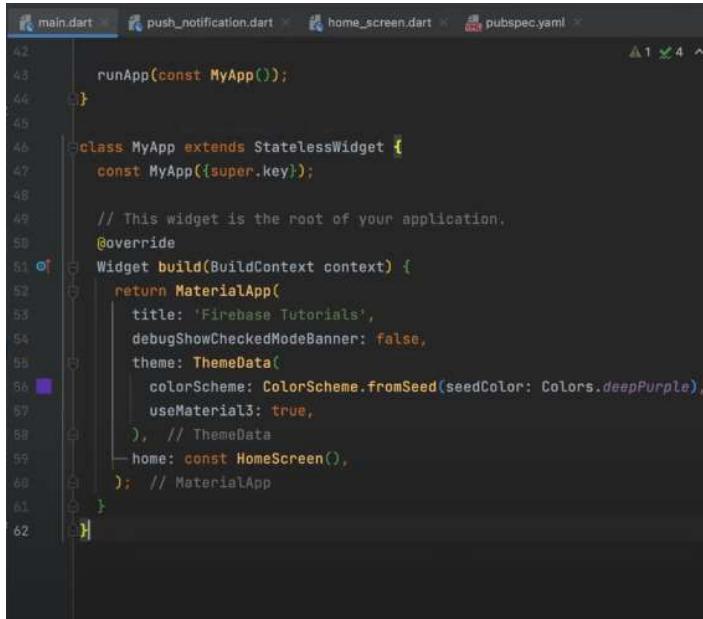
```
class PushNotification {  
  FirebaseMessaging messaging = FirebaseMessaging.instance;
```

```
void requestPermission ()async{
  NotificationSettings settings = await messaging.requestPermission(
    alert: true,
    announcement: false,
    badge: true,
    carPlay: false,
    criticalAlert: false,
    provisional: false,
    sound: true,
  );
}

if (settings.authorizationStatus == AuthorizationStatus.authorized) {
  print('User granted permission');
} else if (settings.authorizationStatus == AuthorizationStatus.provisional) {
  print('User granted provisional permission');
} else {
  print('User declined or has not accepted permission');
}
}
```

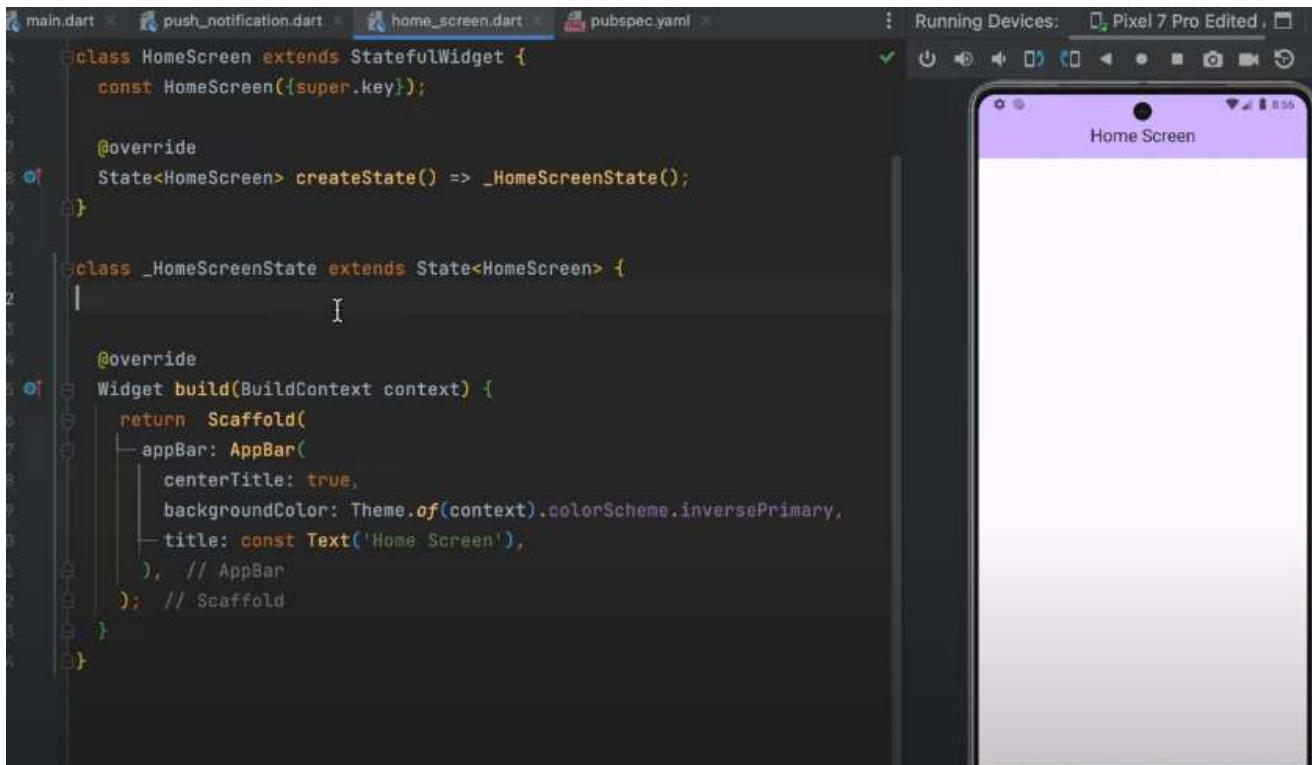
```
Future<String> getDeviceToken () async{
  String? deviceToken = await messaging.getToken();
  print('deviceToken=> $deviceToken');
  return deviceToken?? "";
}
```

Main.dart



```
42 runApp(const MyApp());
43 }
44
45 class MyApp extends StatelessWidget {
46   const MyApp({super.key});
47
48   // This widget is the root of your application.
49   @override
50   Widget build(BuildContext context) {
51     return MaterialApp(
52       title: 'Firebase Tutorials',
53       debugShowCheckedModeBanner: false,
54       theme: ThemeData(
55         colorScheme: ColorScheme.fromSeed(seedColor: Colors.deepPurple),
56         useMaterial3: true,
57       ), // ThemeData
58       home: const HomeScreen(),
59     ); // MaterialApp
60   }
61 }
62 }
```

home_screen.dart



The screenshot shows the Android Studio interface with the 'home_screen.dart' file open in the editor. The code defines a StatelessWidget named 'HomeScreen' with a stateful child '_HomeScreenState'. The build method returns a Scaffold with an inverse primary color scheme and a centered title 'Home Screen'. To the right of the editor, there is a preview window showing a smartphone screen with a purple header bar labeled 'Home Screen'.

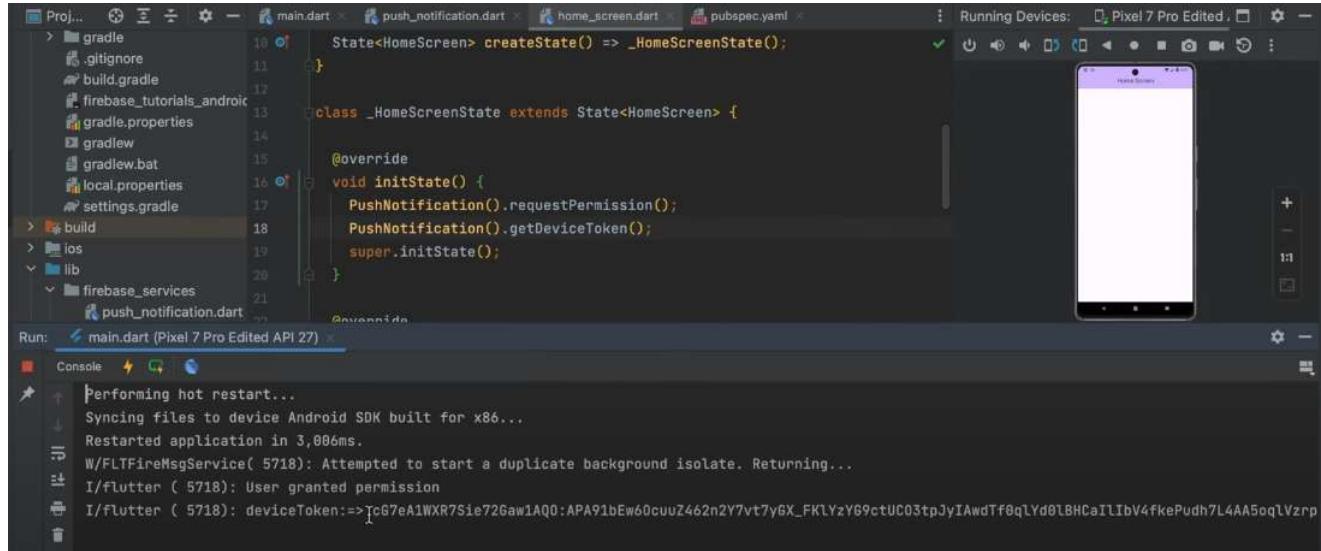
```
class HomeScreen extends StatelessWidget {
  const HomeScreen({super.key});

  @override
  State<HomeScreen> createState() => _HomeScreenState();
}

class _HomeScreenState extends State<HomeScreen> {
  ...

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        centerTitle: true,
        backgroundColor: Theme.of(context).colorScheme.inversePrimary,
        title: const Text('Home Screen'),
      ), // AppBar
    ); // Scaffold
  }
}
```

Updated



Creating channel

The screenshot shows the FlutterFire documentation for 'Notifications'. The sidebar on the left lists various services: Getting Started, Analytics, App Check, Authentication, Cloud Firestore, Cloud Firestore ODM, Cloud Functions, Cloud Messaging (with sub-links for Overview, Usage, Apple Integration, Permissions, Notifications, and Server Integration), and a collapsed section for Notifications. The main content area shows steps for creating a notification channel:

1. Add the `flutter_local_notifications` package to your local project.
2. Create a new `AndroidNotificationChannel` instance:

```
const AndroidNotificationChannel channel = AndroidNotificationChannel(  
    'high_importance_channel', // id  
    'High Importance Notifications', // title  
    'This channel is used for important notifications.', // description  
    importance: Importance.max,  
);
```
3. Create the channel on the device (if a channel with an id already exists, it will be updated):

```
final FlutterLocalNotificationsPlugin flutterLocalNotificationsPlugin =  
    FlutterLocalNotificationsPlugin();  
  
await flutterLocalNotificationsPlugin  
    .resolvePlatformSpecificImplementation<AndroidFlutterLocalNotificationsPlugin>().createNotificationChannel(channel);
```

On the right, there is a sidebar with links for 'Displaying notifications' (Via Firebase, Console, Via Admin SDKs, Via REST, Handling Interaction, Advanced Usage, Foreground Notifications) and a search bar.

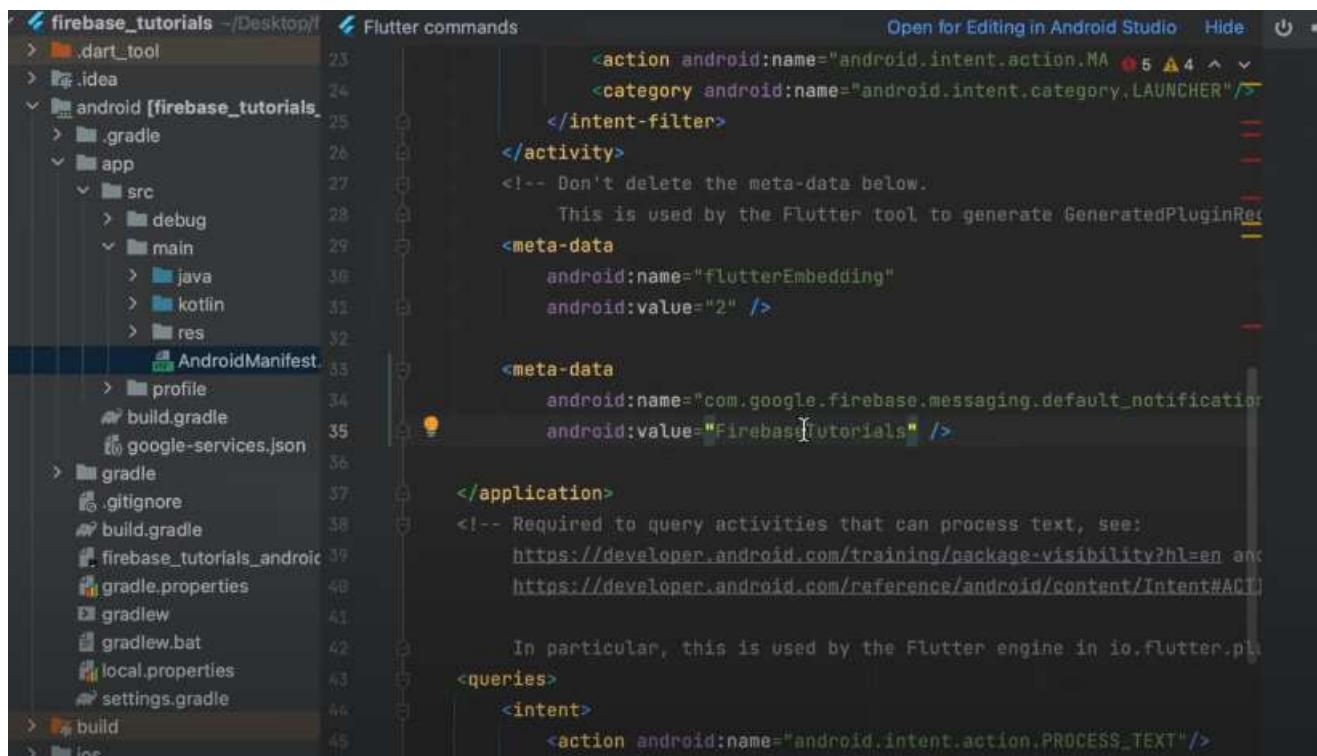
```
main.dart push_notification.dart AndroidManifest.xml home_screen.dart
24 // open app setting
25 print('User declined or has not accepted permission');
26 }
27 }
28
29
30
31 Future<String> getDeviceToken () async{
32     String? deviceToken = await messaging.getToken();
33     print('deviceToken:=> $deviceToken');
34     return deviceToken??"";
35 }
36
37
38
39     AndroidNotificationChannel channel = const AndroidNotificationChannel(
40         'FirebaseTutorials', // id
41         'FirebaseTutorials', // title
42         importance: Importance.high,
43         showBadge: true,
44         playSound: true,
45         enableVibration: true,
46     );
47
48 }
```

Once created, we can now update FCM to use our own channel rather than the default FCM one. To do this, open the

`android/app/src/main/AndroidManifest.xml` file for your FlutterProject project. Add the following `meta-data` schema within the `application` component:



View the [official documentation](#) to learn more.



The screenshot shows the Android Studio interface with the project 'firebase_tutorials' open. On the left, the project structure is displayed, including the .gradle, app, build, and gradle folders. The app folder contains src, debug, main, java, kotlin, and res subfolders, with the AndroidManifest.xml file selected. The right pane shows the content of the AndroidManifest.xml file. The code includes sections for intent filters, meta-data for flutterEmbedding and com.google.firebase.messaging.default_notification_channel, and application queries. A specific line of code is highlighted: `android:value="FirebaseTutorials" />`. The status bar at the top indicates 'Open for Editing in Android Studio'.

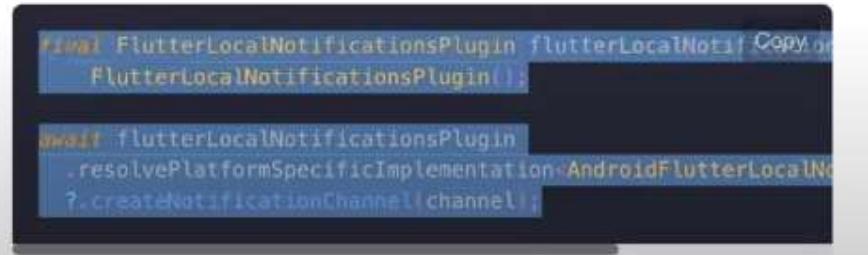
```
<action android:name="android.intent.action.MAIN" />
<category android:name="android.intent.category.LAUNCHER" />
</intent-filter>
</activity>
<!-- Don't delete the meta-data below.
This is used by the Flutter tool to generate GeneratedPluginRegistrant.java -->
<meta-data
    android:name="flutterEmbedding"
    android:value="2" />

<meta-data
    android:name="com.google.firebase.messaging.default_notification_channel"
    android:value="FirebaseTutorials" />

</application>
<!-- Required to query activities that can process text, see:
https://developer.android.com/training/package-visibility?hl=en
https://developer.android.com/reference/android/content/Intent#ACTION_PROCESS_TEXT
In particular, this is used by the Flutter engine in io.flutter.plugins.GeneratedPluginRegistrant -->
<queries>
    <intent>
        <action android:name="android.intent.action.PROCESS_TEXT" />

```

3. Create the channel on the device (if a channel with an id already exists, it will be updated):



```
final FlutterLocalNotificationsPlugin flutterLocalNotificationsPlugin = FlutterLocalNotificationsPlugin();
await flutterLocalNotificationsPlugin.resolvePlatformSpecificImplementation<AndroidFlutterLocalNotificationsPlugin>().createNotificationChannel(channel);
```

Once created, we can now update FCM to use our own channel rather than the default FCM one. To do this, open the

`android/app/src/main/AndroidManifest.xml` file for your

Adding icon

```
49 notificationsPlugin = FlutterLocalNotificationsPlugin();
50
51
52 .utterLocalNotificationsPlugin>()
53
54
55 :Android = const AndroidInitializationSettings('@mipmap/ic_launcher');
56
57 initializationSettings();
58
59 [android: initializationSettingsAndroid, iOS: initializationSettingsIos]]
60
```

The screenshot shows a code editor with several tabs at the top: main.dart, push_notification.dart (which is the active tab), AndroidManifest.xml, home_screen.dart, pubspec.yaml, and Run. The code in the editor is as follows:

```
final FlutterLocalNotificationsPlugin flutterLocalNotificationsPlugin =
  await flutterLocalNotificationsPlugin
    .resolvePlatformSpecificImplementation<AndroidFlutterLocalNotifications>()
    ?.createNotificationChannel(channel);

AndroidInitializationSettings initializationSettingsAndroid = const AndroidInitializationSettings('@mipmap/ic_launcher');

var initializationSettingsIos = const DarwinInitializationSettings();

var initializationSettings = InitializationSettings(android: initializationSettingsAndroid,
  iOS: initializationSettingsIos);

await flutterLocalNotificationsPlugin.initialize(initializationSettings,
  onDidReceiveNotificationResponse: (NotificationResponse response) {
  });
}
```

--- check from doc

• Splash Screen

The screenshot shows a code editor and a file browser for a Flutter application named "shoping_app".

main.dart:

```
import 'package:flutter/material.dart';
import 'package:shoping_app/screens/splash_screen.dart';

void main() {
  runApp(const MyApp());
}

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

  // This widget is the root of your application.
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Shoping App',
      debugShowCheckedModeBanner: false,
      theme: ThemeData(
        colorScheme: ColorScheme.fromSeed(seedColor: Colors.deepPurple),
        useMaterial3: true,
      ),
      home: const SplashScreen(),
    );
  }
}
```

pubspec.yaml:

```
# For information on the generic Dart part of this file, see the
# following page: https://dart.dev/tools/pub/pubspec

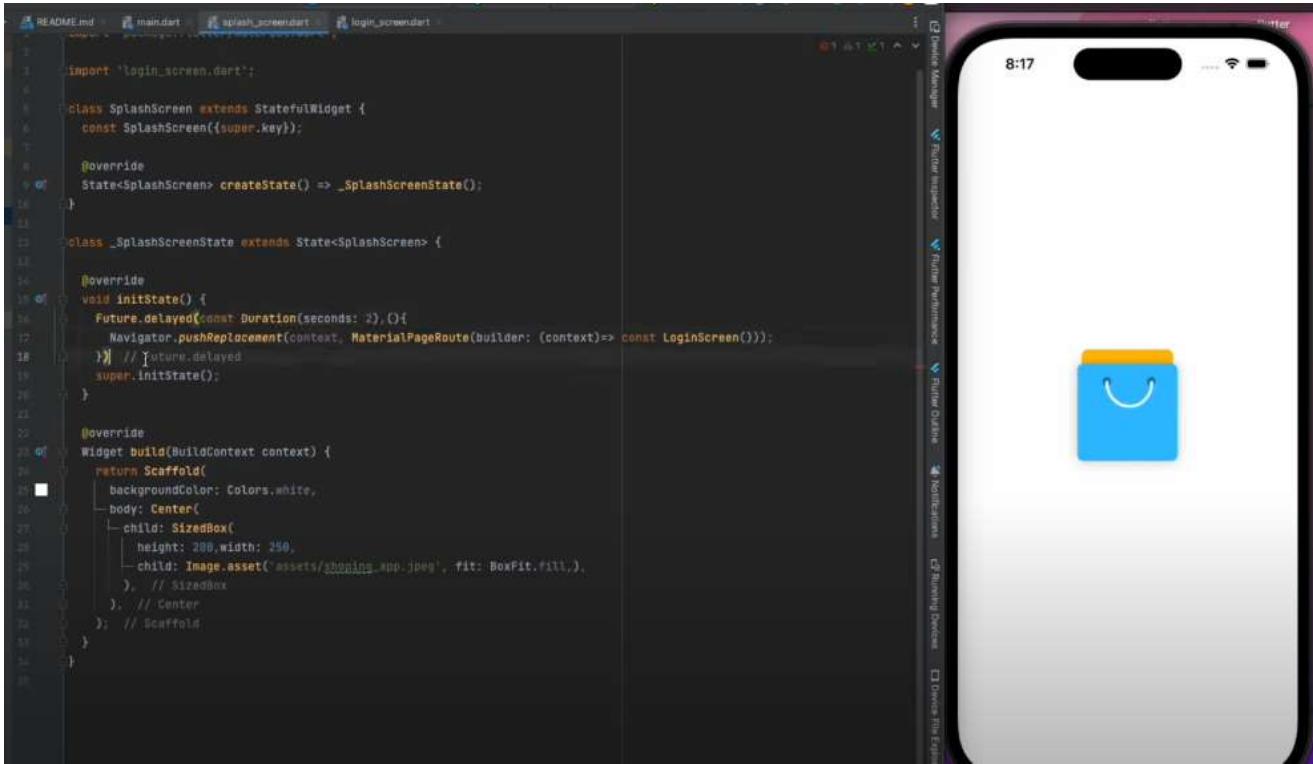
# The following section is specific to Flutter packages.
flutter:

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

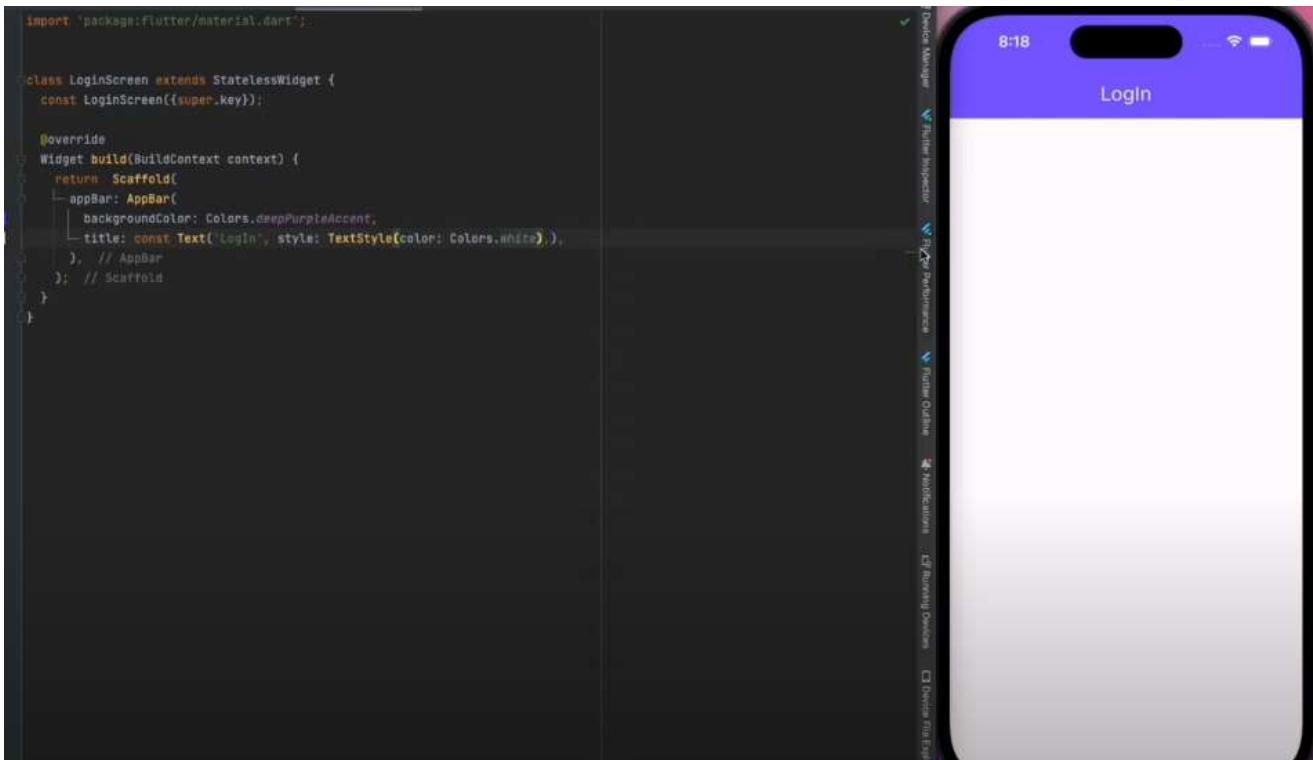
  # To add assets to your application, add an assets section, like this:
  assets:
    - assets/
```

The file browser on the left shows the project structure with files like README.md, main.dart, splash_screen.dart, and pubspec.yaml.

Splash screen



THEN pushReplacement to login screen



- Provider state management

Without state management: Our whole page is rendering just to change on variable and LOC could be 2000+ lines

```
3   int count = 0;
4
5   @override
6   Widget build(BuildContext context) {
7     print("Hello");
8     return Scaffold(
9       appBar: AppBar(
10         centerTitle: true,
11         backgroundColor: Colors.blueGrey,
12         title: Text("Counter", style: TextStyle(color: Colors.white)),
13       ), // AppBar
14
15       body: Center(child: Text(count.toString(), textScaleFactor: 5.0),
16
17
18       floatingActionButton: FloatingActionButton(
19         backgroundColor: Colors.blueGrey,
20         child: Icon(Icons.add),
21         onPressed: () {
22           count++;
23           print(count);
24           setState(() {
25
26           });
27         },
28       ), // FloatingActionButton
29     ); // Scaffold
30   }
31 }
```

Output

```
Flutter  Console   
Performing hot reload...
Syncing files to device sdk gphone64 x86 64...
I/flutter ( 6761): Hello
Reloaded 1 of 666 libraries in 6,929ms (compile: 512 ms, reload: 2120 ms, reassemble: 1745 ms).
D/EGL_emulation( 6761): app_time_stats: avg=1128.75ms min=8.48ms max=26520.28ms count=24
D/EGL_emulation( 6761): app_time_stats: avg=2939.31ms min=2939.31ms max=2939.31ms count=1
I/flutter ( 6761): 3
I/flutter ( 6761): Hello
```



The screenshot shows the pub.dev website for the provider 6.0.5 package. At the top, there's a navigation bar with the pub.dev logo, sign-in options, and a help dropdown. Below the header, the package title "provider 6.0.5" is displayed with a copy icon. It indicates the package was published 7 months ago by dash-overflow.net, is Dart 3 compatible, and has 8.6K likes, 140 pub points, and 100% popularity. The package is categorized under FLUTTER, PLATFORM, ANDROID, IOS, LINUX, MACOS, WEB, and WINDOWS. Below the main stats, there are links for Readme, Changelog, Example, Installing, Versions, and Scores. Language support is listed as English, French, Português, 简体中文, Español, 한국어, ภาษาไทย, and 日本語. The package description states it's a wrapper around InheritedWidget to make them easier to use and more reusable. It lists benefits such as simplified allocation/disposal of resources, lazy-loading, reduced boilerplate, and devtool friendliness. The publisher information shows dash-overflow.net with a GitHub link and a report issues button. A sidebar on the right contains metadata about the package.

provider.dart

A screenshot of a code editor showing the provider.dart file. The code defines a Counter class that implements ChangeNotifier. It has a count field initialized to 0. The increment() method increments the count and notifies listeners. The decrement() method decrements the count and notifies listeners. The code is part of a larger project structure with README.md, main.dart, and home_screen.dart files.

```
import 'package:flutter/cupertino.dart';
class Counter with ChangeNotifier{
  int count = 0;
  void increment(){
    count++;
    notifyListeners();
  }
  void decrement(){
    count--;
    notifyListeners();
  }
}
```

Wrapping in main.dart

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

import 'home_screen.dart';

void main() {
  runApp(MultiProvider(
    providers: [
      ChangeNotifierProvider(create: (_) => Counter()),
    ],
    child: const MyApp()),); // MultiProvider
}

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

  // This widget is the root of your application.
  @override
  Widget build(BuildContext context) {
```

Accessing count variable in Counter Widget using context.watch<ProviderName>.variable as reference

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

class Count extends StatelessWidget {
  const Count({Key? key}) : super(key: key);

  @override
  Widget build(BuildContext context) {
    return Text(context.watch<Counter>().count.toString(), textScaleFactor: 7,);
  }
}
```

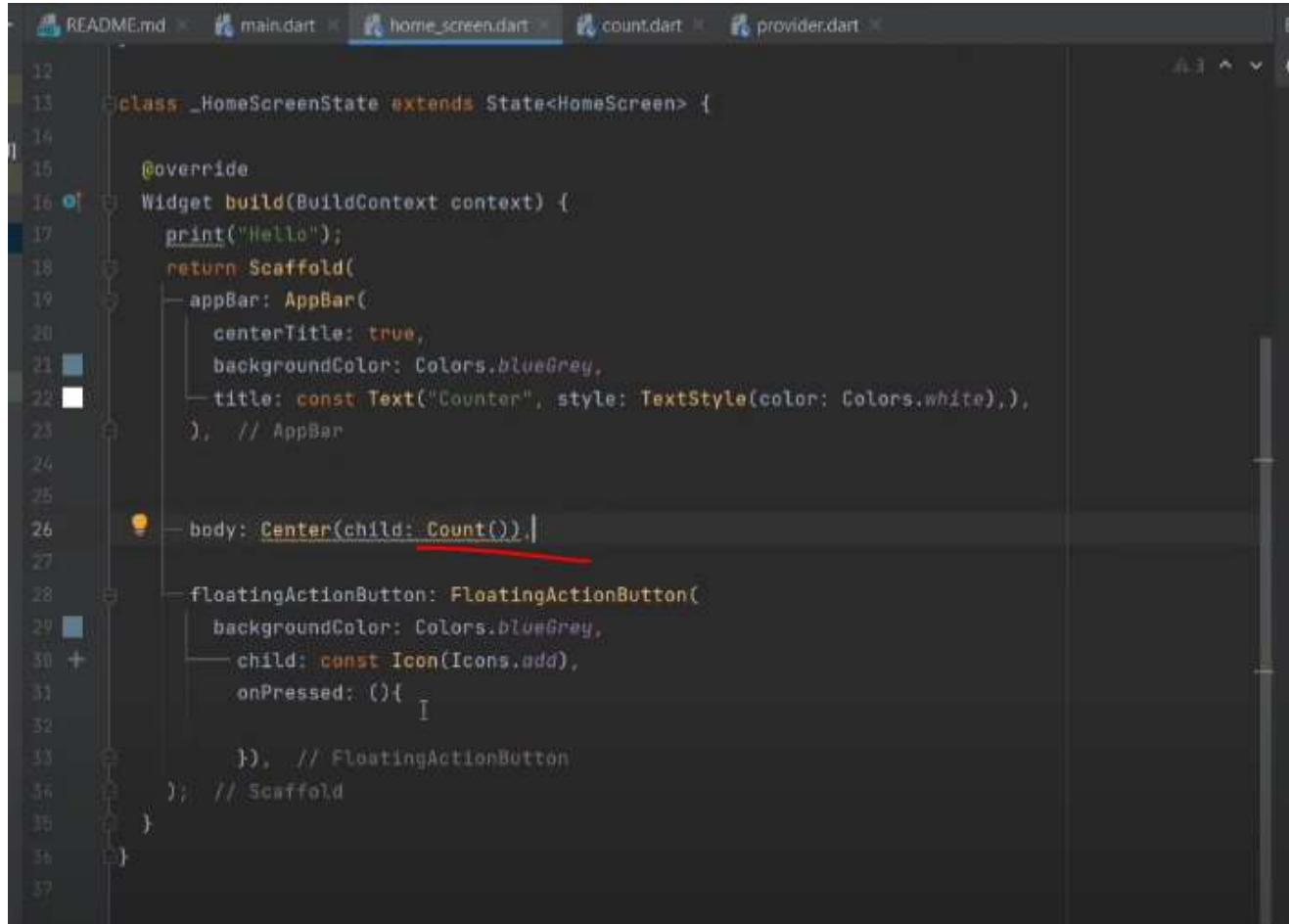
Or

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

class Count extends StatelessWidget {
  const Count({Key? key}) : super(key: key);

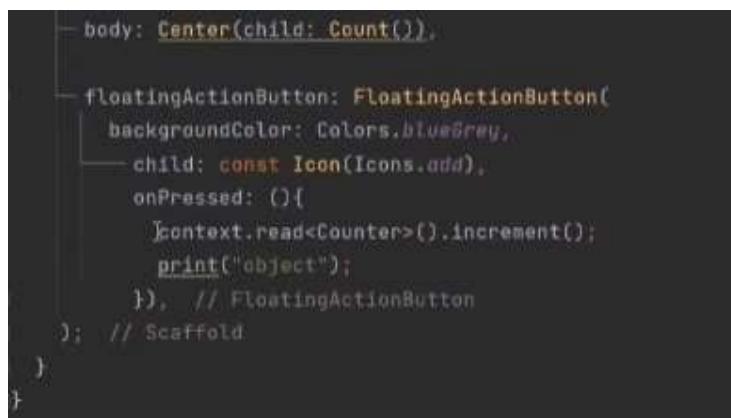
  @override
  Widget build(BuildContext context) {
    final myProvider = context.watch<Counter>();
    return Text(myProvider.count.toString(), textScaleFactor: 7,);
  }
}
```

Using Count widget in home screen



```
12
13     class _HomeScreenState extends State<HomeScreen> {
14
15     @override
16     Widget build(BuildContext context) {
17         print("Hello");
18         return Scaffold(
19             appBar: AppBar(
20                 centerTitle: true,
21                 backgroundColor: Colors.blueGrey,
22                 title: const Text("Counter", style: TextStyle(color: Colors.white),),
23             ), // AppBar
24
25
26             body: Center(child: Count());
27
28             floatingActionButton: FloatingActionButton(
29                 backgroundColor: Colors.blueGrey,
30                 child: const Icon(Icons.add),
31                 onPressed: () {
32                     print("object");
33                 }, // FloatingActionButton
34             ); // Scaffold
35         );
36     }
37 }
```

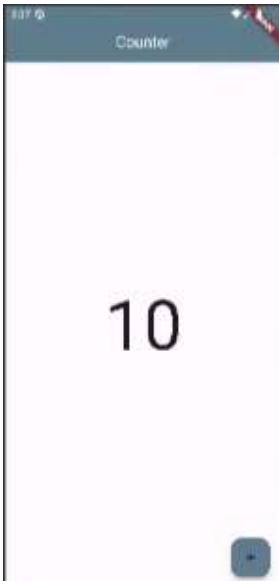
Updating value of count variable using
context.read<ProviderName>.function



```
body: Center(child: Count()),
floatingActionButton: FloatingActionButton(
    backgroundColor: Colors.blueGrey,
    child: const Icon(Icons.add),
    onPressed: () {
        context.read<Counter>().increment();
        print("object");
    }, // FloatingActionButton
); // Scaffold
}
```

Output: now we can see that “Hello” is not being printed again and again when we are increasing the value of count using provider

```
Console: I/flutter ( 7582): object  
I/flutter ( 7582): object  
I/flutter ( 7582): object  
I/flutter ( 7582): object
```



- We can also call apis using provider

Making Model of api response

The screenshot shows the Reqres API documentation at reqres.in. The page lists various API endpoints for users:

Method	Endpoint	Description
GET	/api/users?page=2	LIST USERS
GET	/api/users/:id	SINGLE USER
GET	/api/users/:id/	SINGLE USER NOT FOUND
GET	/api/users	LIST <RESOURCE>
GET	/api/users/:id/:field	SINGLE <RESOURCE>
GET	/api/users/:id/:field/	SINGLE <RESOURCE> NOT FOUND

On the right, a sample request and response are shown for the "/api/users?page=2" endpoint:

```
Request: /api/users?page=2
Response: 200
{
  "page": 2,
  "per_page": 6,
  "total": 12,
  "total_pages": 2,
  "data": [
    {
      "id": 7,
      "email": "michael.lawson@reqres.in",
      "first_name": "Michael",
      "last_name": "Lawson",
      "avatar": "https://reqres.in/img/michael.jpg"
    }
  ]
}
```

JSON to Dart

Paste your JSON in the textarea below, click convert and get your Dart classes for free.

JSON

```
34 },  
35 },  
36 {  
37   "id": 11,  
38   "email": "george.edwards@reqres.in",  
39   "first_name": "George",  
40   "last_name": "Edwards",  
41   "avatar": "https://reqres.in/img/users/11.jpg",  
42 },  
43 {  
44   "id": 12,  
45   "email": "rachel.howell@reqres.in",  
46   "first_name": "Rachel",  
47   "last_name": "Howell",  
48   "avatar": "https://reqres.in/img/users/12.jpg",  
49 },  
50 },  
51 "support": {  
52   "url": "https://reqres.in/#support-link",  
53   "text": "To keep ReqRes free, consider getting a Pro Plan."  
54 }  
55 }
```

Model

Generate Dart Use private fields

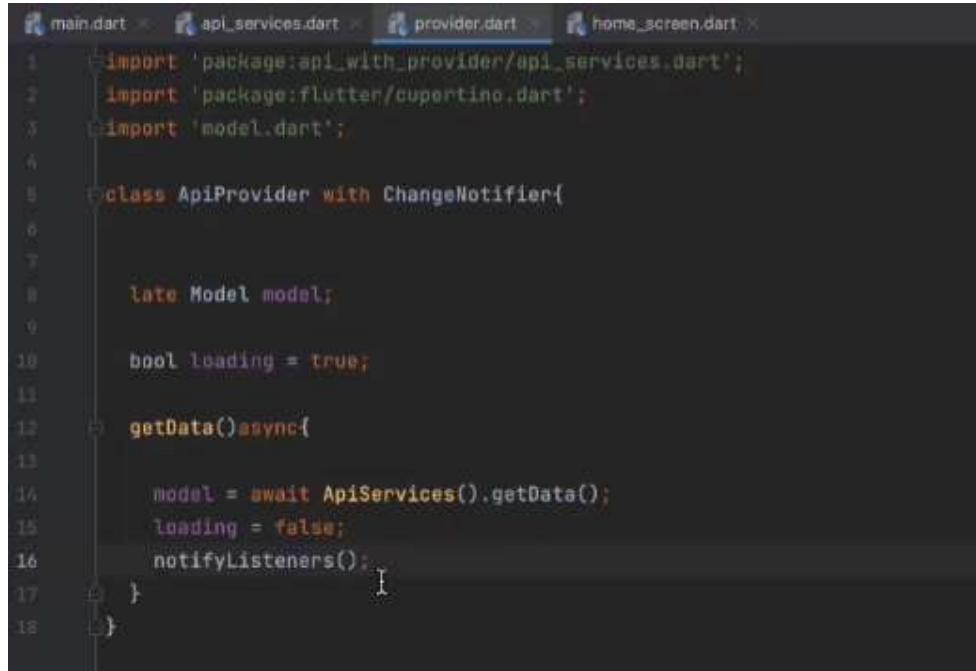
Copy Dart code to clipboard

```
class Model {  
  int? page;  
  int? perPage;  
  int? total;  
  int? totalPages;  
  List<Data>? data;  
  Support? support;  
  
  Model({  
    this.page,  
    this.perPage,  
    this.total,  
    this.totalPages,  
    this.data,  
    this.support});  
  
  Model.fromJson(Map<String, dynamic> json) {  
    page = json['page'];  
    perPage = json['per_page'];  
    total = json['total'];  
    totalPages = json['total_pages'];  
    if (json['data'] != null) {  
      data = <Data>[];  
      json['data'].forEach((v) {  
        data!.add(new Data.fromJson(v));  
      });  
    }  
  }  
}  
  
class Data {  
  int id;  
  String email;  
  String first_name;  
  String last_name;  
  String avatar;  
}  
  
class Support {  
  String url;  
  String text;  
}
```

Copy the above model to model.dart classname starts with CapitalLetter function with small

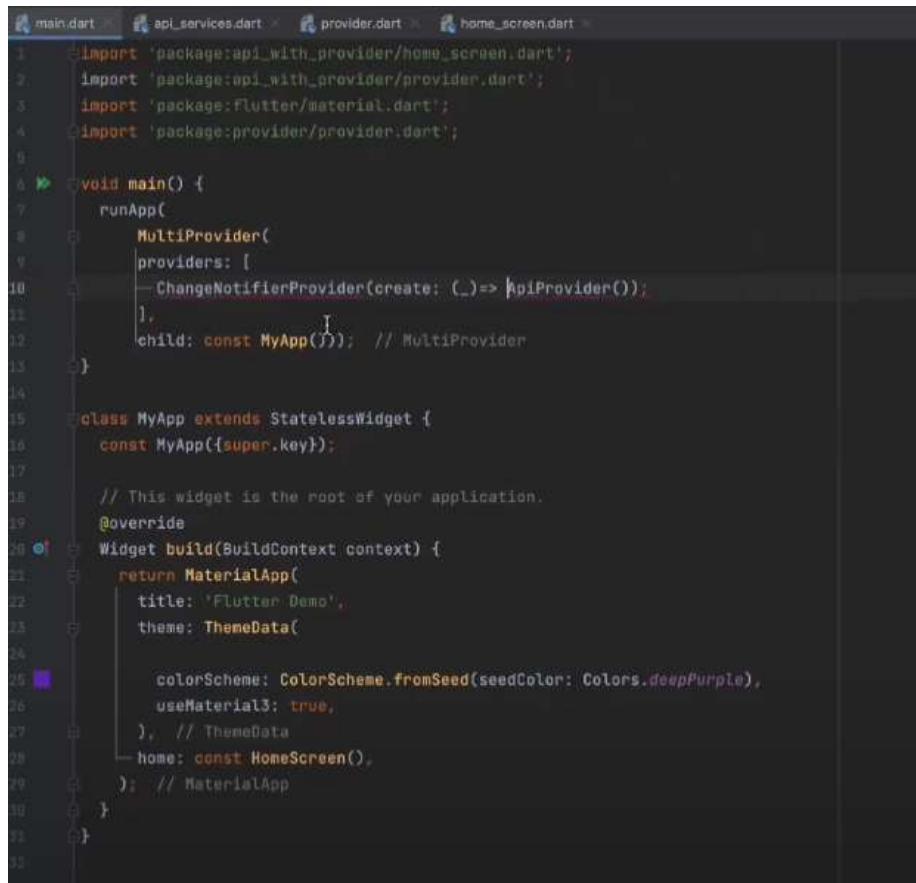
```
main.dart  api_services.dart  provider.dart  home_screen.dart  
1 import 'dart:convert';  
2  
3 import 'package:http/http.dart' as http;  
4  
5 import 'model.dart';  
6  
7 class ApiServices{  
8  
9   Future<Model> getData(){  
10  
11     late Model model;  
12  
13     final response = await http.get(Uri.parse("https://reqres.in/api/users?page=2"));  
14  
15  
16     if(response.statusCode == 200 || response.statusCode == 201){  
17  
18       model = Model.fromJson(json.decode(response.body));  
19     }else{  
20       print("Error");  
21     }  
22  
23     return model;  
24   }  
25 }  
26 }
```

Provider



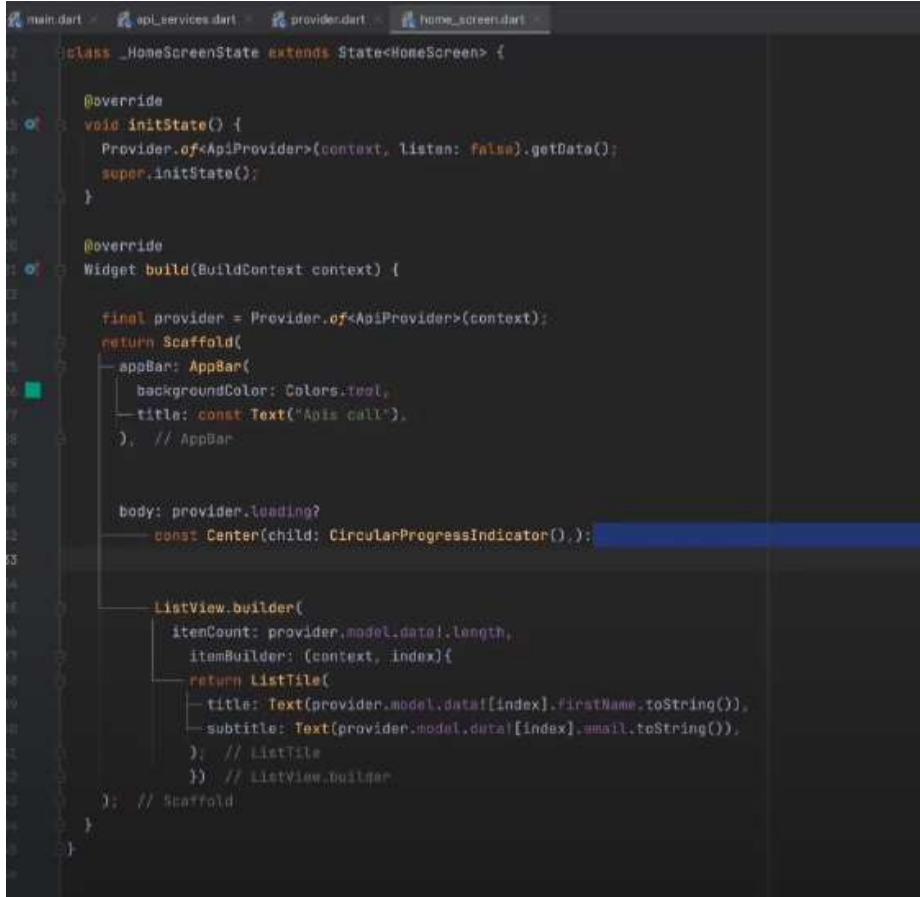
```
1 import 'package:api_with_provider/api_services.dart';
2 import 'package:flutter/cupertino.dart';
3 import 'model.dart';
4
5 class ApiProvider with ChangeNotifier{
6
7
8     late Model model;
9
10    bool loading = true;
11
12    getData() async{
13
14        model = await ApiServices().getData();
15        loading = false;
16        notifyListeners();
17    }
18}
```

Wrapping out main.dart app



```
1 import 'package:api_with_provider/home_screen.dart';
2 import 'package:api_with_provider/provider.dart';
3 import 'package:flutter/material.dart';
4 import 'package:provider/provider.dart';
5
6 void main() {
7     runApp(
8         MultiProvider(
9             providers: [
10                 ChangeNotifierProvider(create: (_) => ApiProvider());
11             ],
12             child: const MyApp()); // MultiProvider
13     );
14
15     class MyApp extends StatelessWidget {
16         const MyApp({super.key});
17
18         // This widget is the root of your application.
19         @override
20         Widget build(BuildContext context) {
21             return MaterialApp(
22                 title: 'Flutter Demo',
23                 theme: ThemeData(
24
25                     colorScheme: ColorScheme.fromSeed(seedColor: Colors.deepPurple),
26                     useMaterial3: true,
27                 ), // ThemeData
28                 home: const HomeScreen(),
29             ); // MaterialApp
30         }
31     }
32 }
```

Using ApiService provider



```
class _HomeScreenState extends State<HomeScreen> {

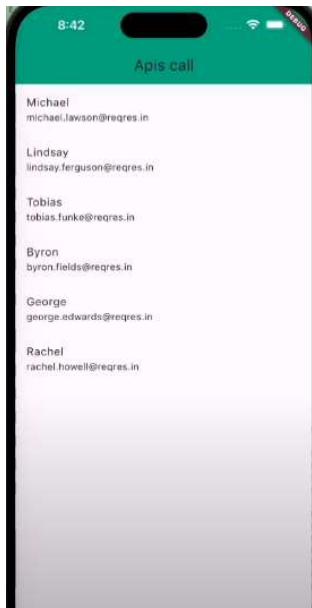
  @override
  void initState() {
    Provider.of<ApiProvider>(context, listen: false).getData();
    super.initState();
  }

  @override
  Widget build(BuildContext context) {
    final provider = Provider.of<ApiProvider>(context);
    return Scaffold(
      appBar: AppBar(
        backgroundColor: Colors.teal,
        title: const Text("Apis call"),
      ), // AppBar

      body: provider.loading?
        const Center(child: CircularProgressIndicator()): // CircularProgressIndicator()

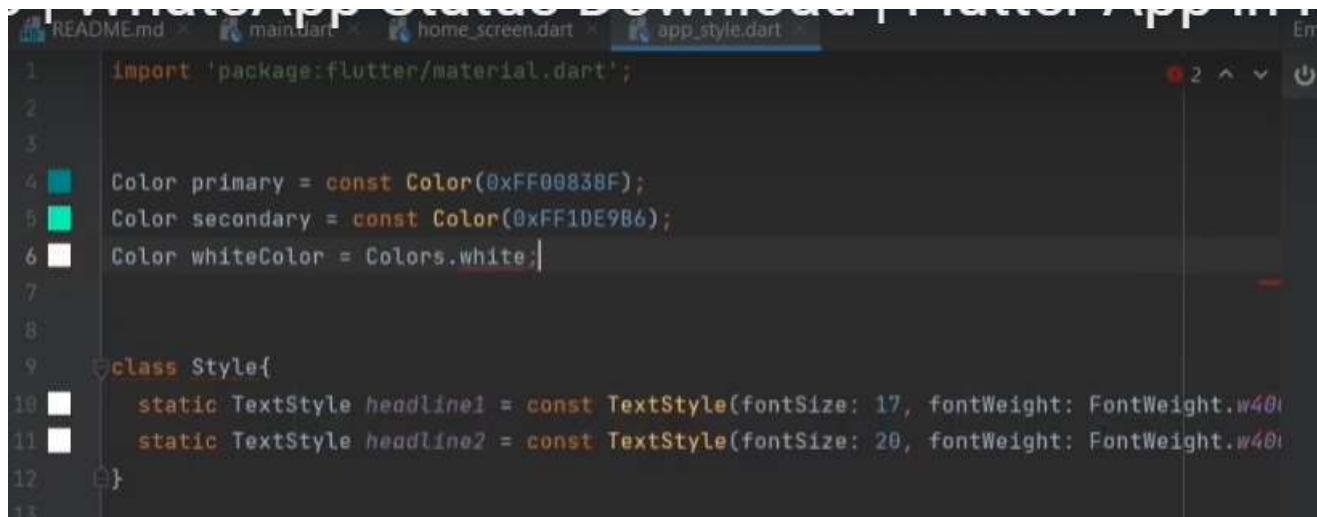
        ListView.builder(
          itemCount: provider.model.data.length,
          itemBuilder: (context, index){
            return ListTile(
              title: Text(provider.model.data[index].firstName.toString()),
              subtitle: Text(provider.model.data[index].email.toString()),
            ); // ListTile
          }) // ListView.builder
    ); // Scaffold
  }
}
```

Output



- Whatsapp status downloader:

<https://youtu.be/aqRdDnmQN7g?list=PLQfqkPIP5hGjKlxifw8KkvtzxLb8fntJs>

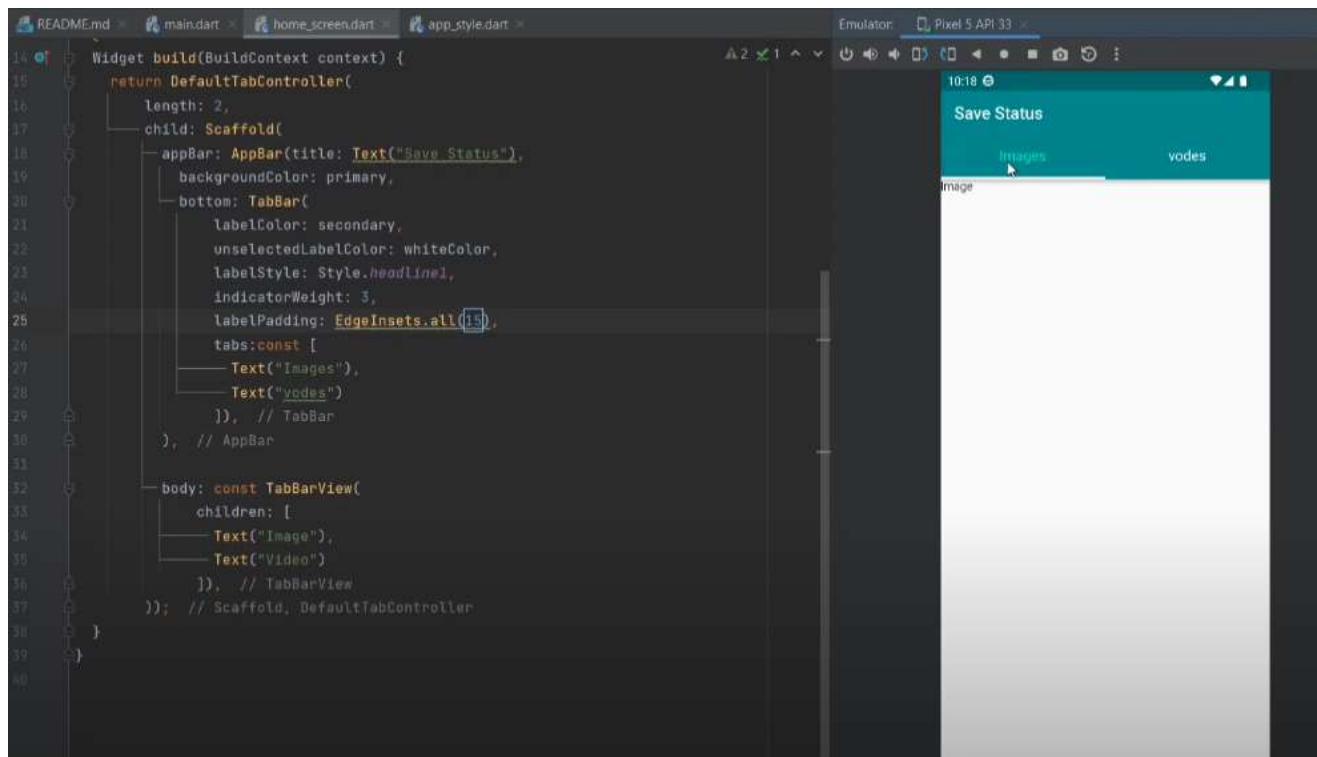


```
import 'package:flutter/material.dart';

Color primary = const Color(0xFF00838F);
Color secondary = const Color(0xFF1DE9B6);
Color whiteColor = Colors.white;

class Style{
    static TextStyle headline1 = const TextStyle(fontSize: 17, fontWeight: FontWeight.w400);
    static TextStyle headline2 = const TextStyle(fontSize: 20, fontWeight: FontWeight.w400);
}
```

Top Navigation Bar



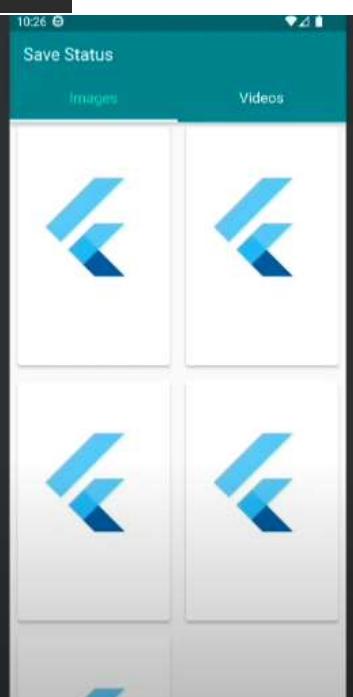
```
Widget build(BuildContext context) {
    return DefaultTabController(
        length: 2,
        child: Scaffold(
            appBar: AppBar(title: Text("Save Status"),
                backgroundColor: primary,
                bottom: TabBar(
                    labelColor: secondary,
                    unselectedLabelColor: whiteColor,
                    labelStyle: Style.headline1,
                    indicatorWeight: 3,
                    labelPadding: EdgeInsets.all(15),
                    tabs:const [
                        Text("Images"),
                        Text("Videos")
                    ]), // TabBar
            ), // AppBar

            body: const TabBarView(
                children: [
                    Text("Image"),
                    Text("Video")
                ]), // TabBarView
        )); // Scaffold, DefaultTabController
}
```

Creating image screen

```
        body: TabBarView(
            children: [
                ImageScreen(),
                Text("Video")
            ],
        ), // TabBarView
    ); // Scaffold, DefaultTabController
}
}
```

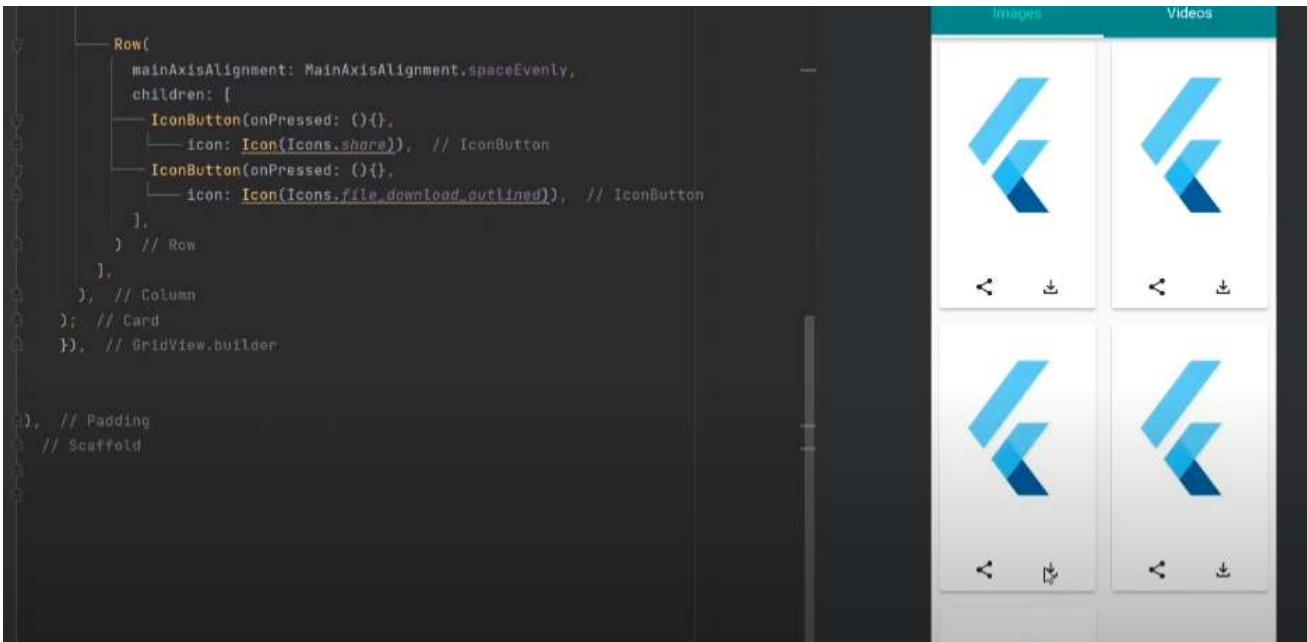
ImageScreen



The screenshot shows the code for the `ImageScreen` class and a preview of the app's interface. The code defines a `GridView` with 5 items per row, where each item is a `Card` containing a `Container` with a `NetworkImage`. The preview shows a 2x2 grid of these cards.

```
State<ImageScreen> createState() => _ImageScreenState();

class _ImageScreenState extends State<ImageScreen> {
    @override
    Widget build(BuildContext context) {
        return Scaffold(
            body: Padding(padding: EdgeInsets.symmetric(horizontal: 5),
            child: GridView.builder(
                itemCount: 5,
                gridDelegate: SliverGridDelegateWithFixedCrossAxisCount(
                    crossAxisCount: 2,
                    crossAxisSpacing: 10.0,
                    childAspectRatio: 0.65,
                    mainAxisSpacing: 10.0,
                ),
                itemBuilder: (context, index){
                    return Card(
                        elevation: 3,
                        child: Column(
                            children: [
                                Container(
                                    height: MediaQuery.of(context).size.width*0.6,
                                    width: MediaQuery.of(context).size.width*0.45,
                                    decoration: BoxDecoration(
                                        borderRadius: BorderRadius.circular(5),
                                        image: const DecorationImage(
                                            fit: BoxFit.fill,
                                            image: NetworkImage("https://yt3.googleusercontent.com/yc/AGII..."),
                                        ),
                                    ),
                                ),
                            ],
                        ),
                    );
                },
            ),
        );
    }
}
```



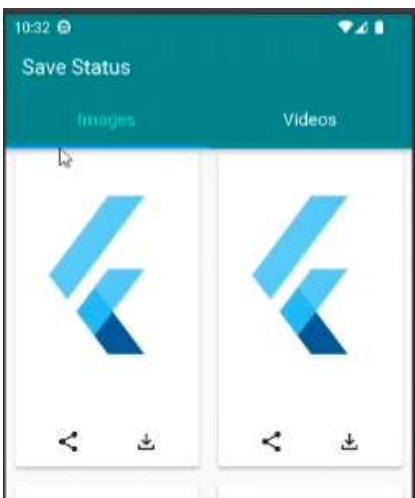
Changing status bar color

```
README.md main.dart home_screen.dart image_screen.dart videos_screen.dart
import 'package:whatsapp_status_download/screens/home_screen.dart';
import 'package:whatsapp_status_download/utils/app_style.dart';

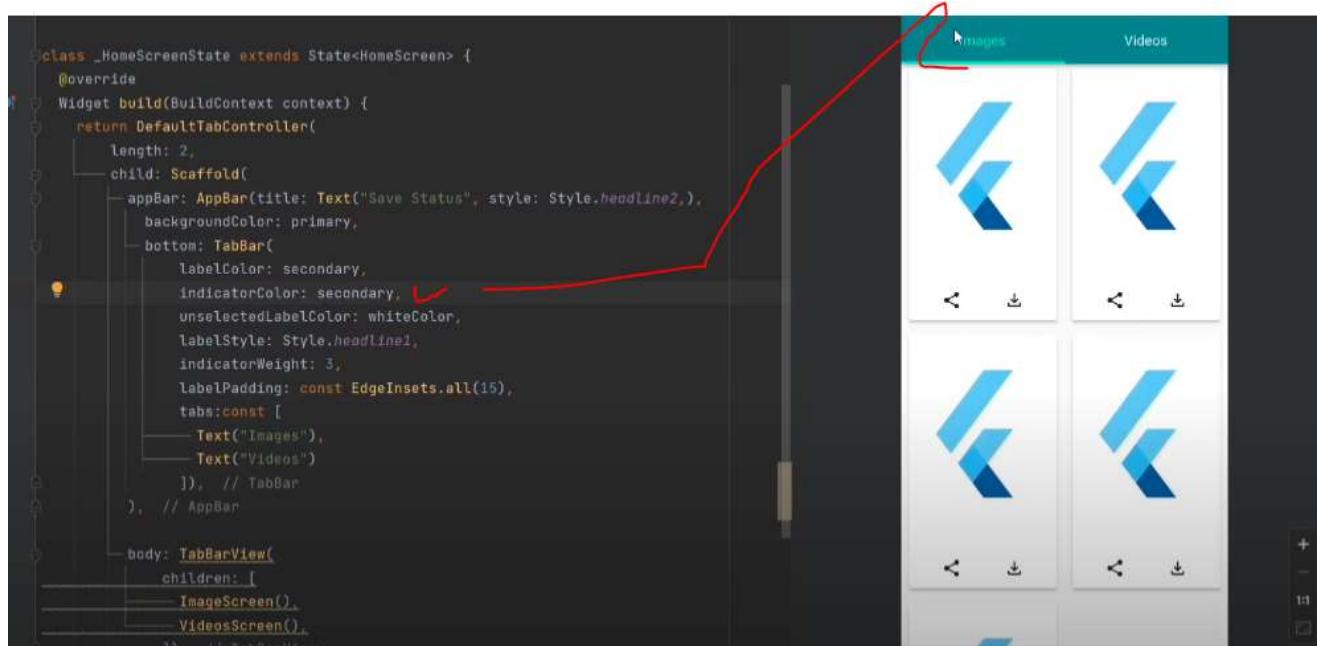
void main() {
    SystemChrome.setSystemUIOverlayStyle(SystemUiOverlayStyle(
        statusBarColor: primary));
    runApp(const MyApp());
}
```

A screenshot of a code editor showing the main.dart file. The file imports necessary packages and defines the main function. Inside the main function, it calls SystemChrome.setSystemUIOverlayStyle with the statusBarColor set to primary. A red bracket highlights this specific line of code.

Output

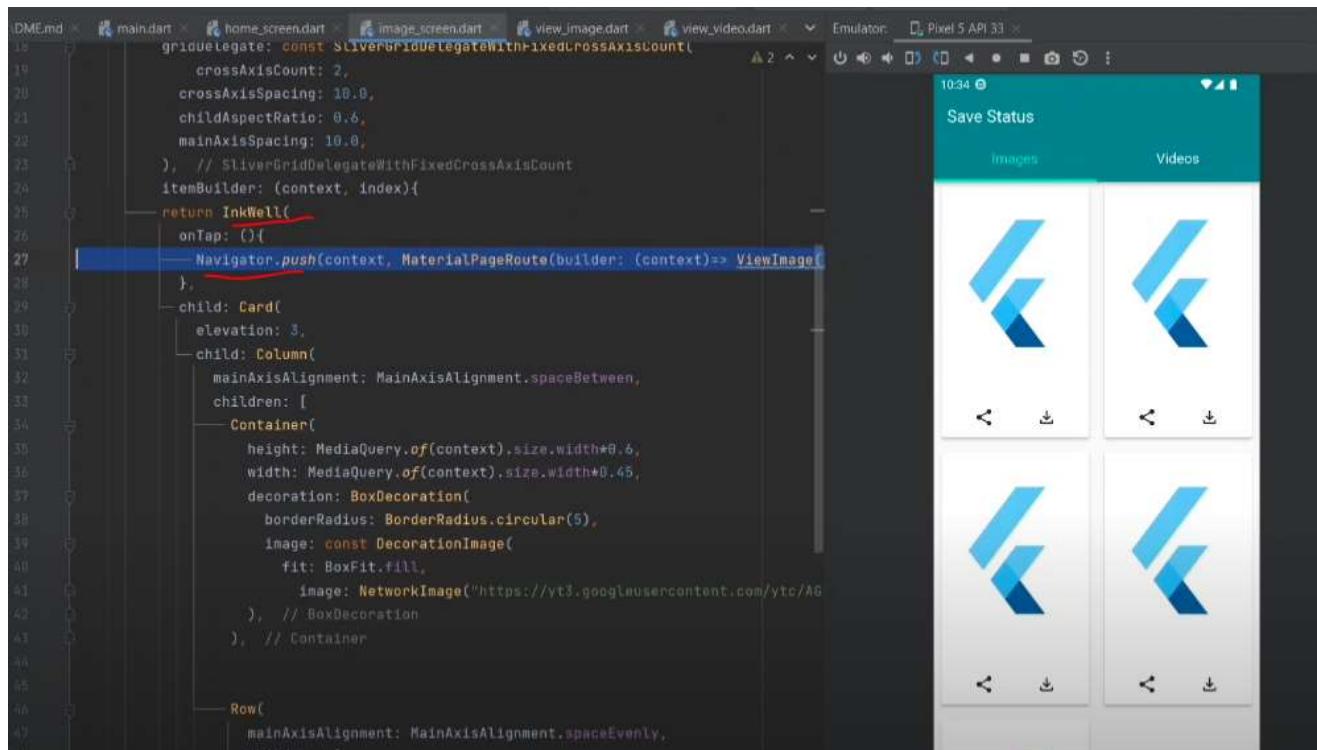


Changing underline or indicator color



Now making similar VideosScreen

Making viewImage and viewVideo screen



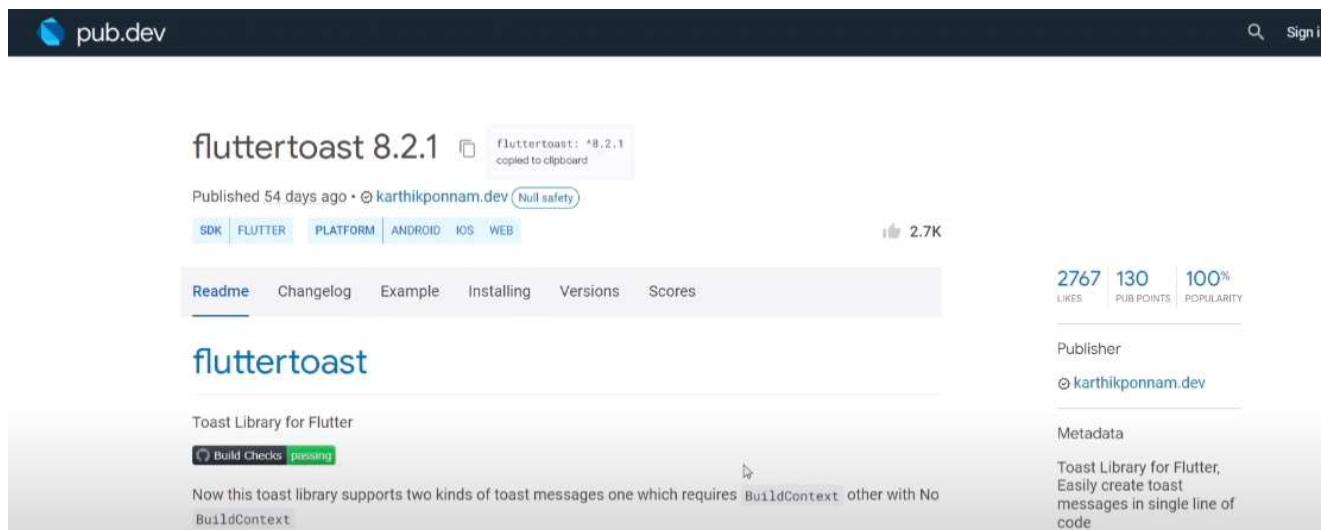
OR (we can apply InkWell to navigate, on Container)

```
return Card(
  elevation: 3,
  child: Column(
    mainAxisAlignment: MainAxisAlignment.spaceBetween,
    children: [
      InkWell(
        onTap: (){
          Navigator.push(context, MaterialPageRoute(builder: (context)=> View
        },
        child: Container(
          height: MediaQuery.of(context).size.width*0.6,
          width: MediaQuery.of(context).size.width*0.45,
          decoration: BoxDecoration(
            borderRadius: BorderRadius.circular(5),
            image: const DecorationImage(
              fit: BoxFit.fill,
              image: NetworkImage("https://yt3.googleusercontent.com/yc/AG
            ), // BoxDecoration
          ), // Container
        ), // InkWell

      Row(
        mainAxisSize: MainAxisSize.spaceEvenly,

```

Installing Flutter toast



The screenshot shows the fluttertoast package page on pub.dev. At the top, there's a navigation bar with a search icon and a sign-in button. Below it, the package name "fluttertoast 8.2.1" is displayed with a copy-to-clipboard icon. A note indicates "copied to clipboard". The package was published 54 days ago by karthikponnam.dev, with null safety support.

Key features listed include "Readme", "Changelog", "Example", "Installing", "Versions", and "Scores". On the right, metrics show 2767 likes, 130 pub points, and 100% popularity. The publisher is listed as karthikponnam.dev. Metadata notes that the package is a "Toast Library for Flutter" used for easily creating toast messages in single-line code.

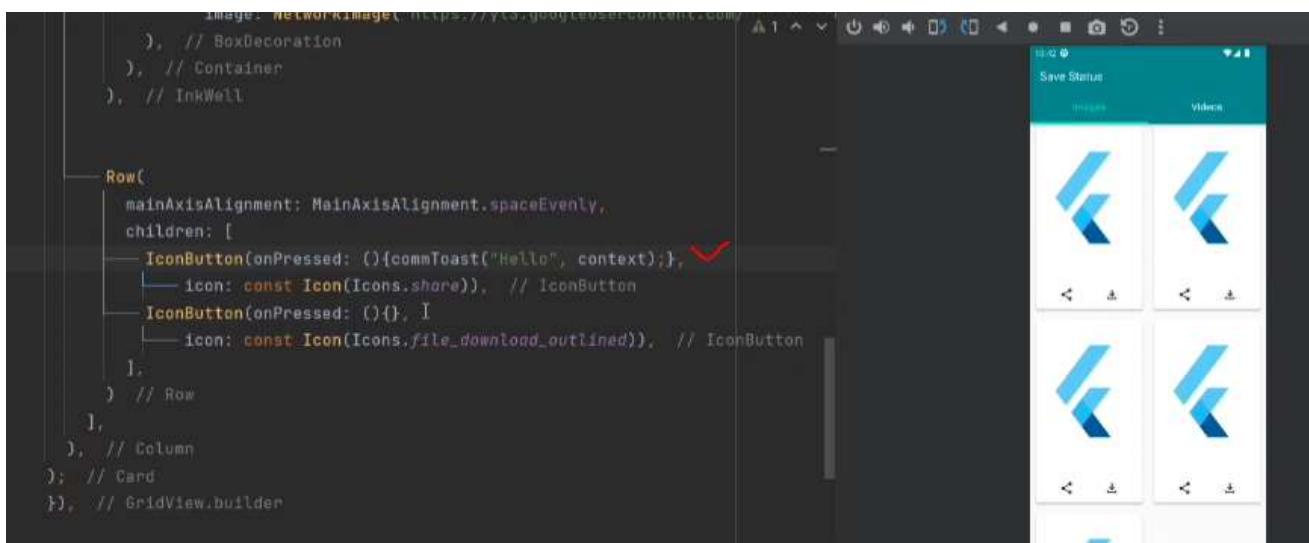
Build status is shown as "Build Checks passing". The description states that the library supports two kinds of toast messages: one requiring `BuildContext` and another with `No BuildContext`.

The screenshot shows the Android Studio interface with the project 'whatsapp_status_download' open. The code editor displays the file 'common_toast.dart'. The code imports 'package:flutter/cupertino.dart', 'package:fluttertoast/fluttertoast.dart', and 'package:whatsapp_status_download/utils/app_style.dart'. It defines a function 'commToast(String message, BuildContext context)' that uses 'Fluttertoast.showToast()' to display a toast message with specific parameters: msg: message, toastLength: Toast.LENGTH_SHORT, gravity: ToastGravity.BOTTOM, timeInSecForIosWeb: 1, textColor: secondary, backgroundColor: whiteColor, and fontSize: 16.

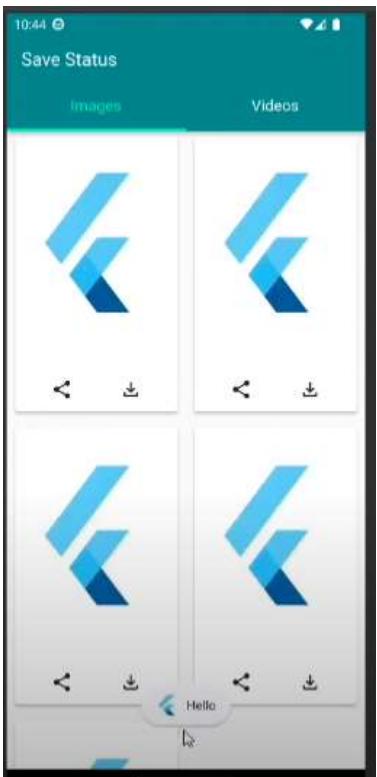
```
import 'package:flutter/cupertino.dart';
import 'package:fluttertoast/fluttertoast.dart';
import 'package:whatsapp_status_download/utils/app_style.dart';

commToast(String message, BuildContext context){
    Fluttertoast.showToast(
        msg: message,
        toastLength: Toast.LENGTH_SHORT,
        gravity: ToastGravity.BOTTOM,
        timeInSecForIosWeb: 1,
        textColor: secondary,
        backgroundColor: whiteColor,
        fontSize: 16,
    );
}
```

Using toast

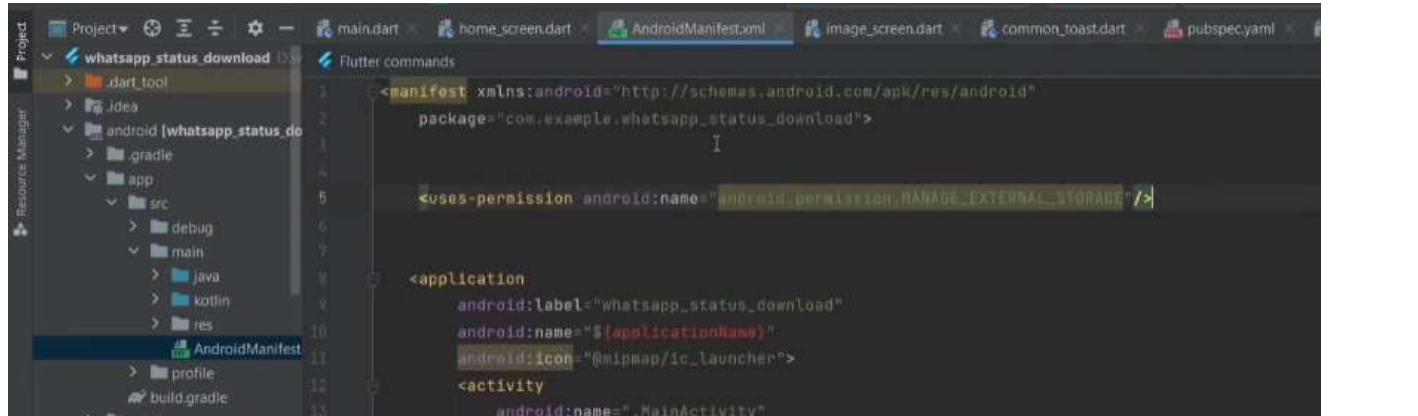


Output



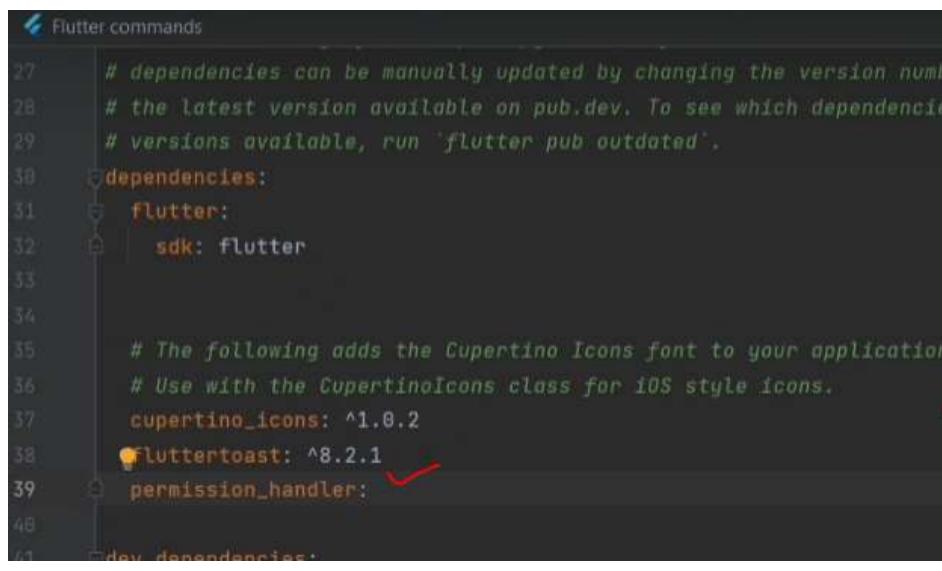
Creating services of whatsapp status downloader app

1. Gallery permission



```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.whatsapp_status_download">
    <uses-permission android:name="android.permission.MANAGE_EXTERNAL_STORAGE"/>
    <application
        android:label="whatsapp_status_download"
        android:name="${applicationName}"
        android:icon="@mipmap/ic_launcher">
        <activity
            android:name=".MainActivity">

```



```
# dependencies can be manually updated by changing the version number
# the latest version available on pub.dev. To see which dependencies
# versions available, run `flutter pub outdated`.
dependencies:
  flutter:
    sdk: flutter

  # The following adds the Cupertino Icons font to your application.
  # Use with the CupertinoIcons class for iOS style icons.
  cupertino_icons: ^1.0.2
  fluttertoast: ^8.2.1
  permission_handler: ^8.0.4

dev_dependencies:
```



```
class HomeScreen extends StatefulWidget {
  const HomeScreen({Key? key}) : super(key: key);

  @override
  State<HomeScreen> createState() => _HomeScreenState();
}

class _HomeScreenState extends State<HomeScreen> {

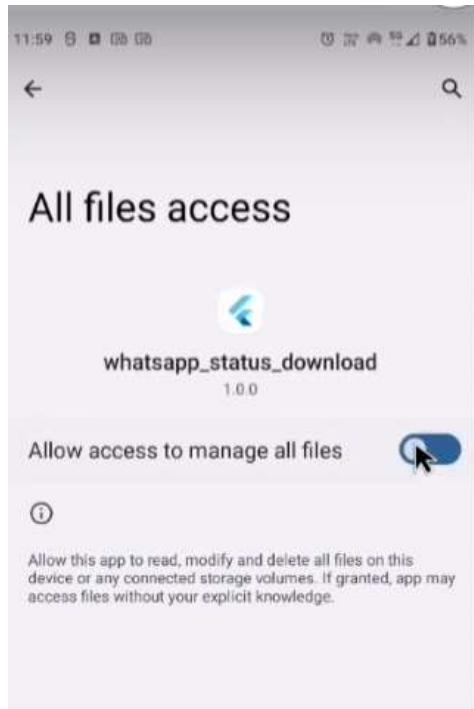
  void _permission() async{
    Map<Permission, PermissionStatus> result = await [
      Permission.storage,
      Permission.manageExternalStorage,
    ].request();

    if(result[Permission.storage] == PermissionStatus.granted && result[Permission.manageExternalStorage] == PermissionStatus.granted){

```

```
@override  
void initState() {  
    _permission();  
    super.initState();  
}
```

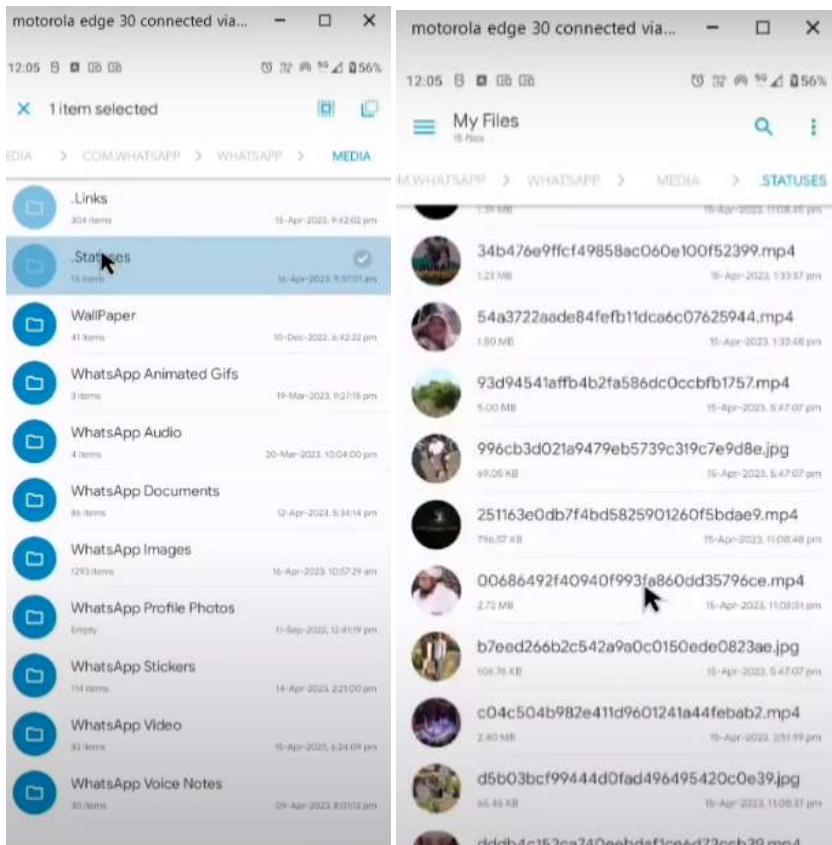
Output



Setting up Provider for state management

```
dependencies:  
  flutter:  
    sdk: flutter  
  
  # The following adds the Cupertino Icons font to your application.  
  # Use with the CupertinoIcons class for iOS style icons.  
  cupertino_icons: ^1.0.2  
  fluttertoast: ^8.2.1  
  permission_handler:  
    provider:
```

Adding images and videos in map



```

home_screen.dart app_provider.dart image_screen.dart pubspec.yaml view_image.dart
1 import 'dart:io';
2 import 'package:flutter/widgets.dart';
3 import 'package:provider/provider.dart';
4
5
6 class AppProvider extends ChangeNotifier{
7
8     List<FileSystemEntity> getImage = [];
9     List<FileSystemEntity> getVideo = [];
10
11     bool isWhatsappAvailable = false;  I
12
13
14
15     getStatus(String endPoint){
16
17         final directory = Directory("/storage/emulated/0/Android/media/com.whatsapp/WhatsApp/Media/.Statuses");

```

```

final directory = Directory("/storage/emulated/0/Android/media/com.whatsapp/WhatsApp/Media/.Statuses");

if(directory.existsSync()){
    final item = directory.listSync();

    isWhatsappAvailable = true;
    notifyListeners();

    if(endPoint == ".mp4"){
        getVideo = item.where((element) => element.path.endsWith(".mp4")).toList();
        notifyListeners();
    }else{
        getImage = item.where((element) => element.path.endsWith(".jpg")).toList();
        notifyListeners();
    }
}

}else{
    isWhatsappAvailable = false;
    notifyListeners();
}
}
}

```

To GET video thumbnail using package

```

# The following adds the Cupertino Icons font to your application.
# Use with the CupertinoIcons class for iOS style icons.
cupertino_icons: ^1.0.2
flutertoast: ^8.2.1
permission_handler:
provider:
video_thumbnail: |
I

```

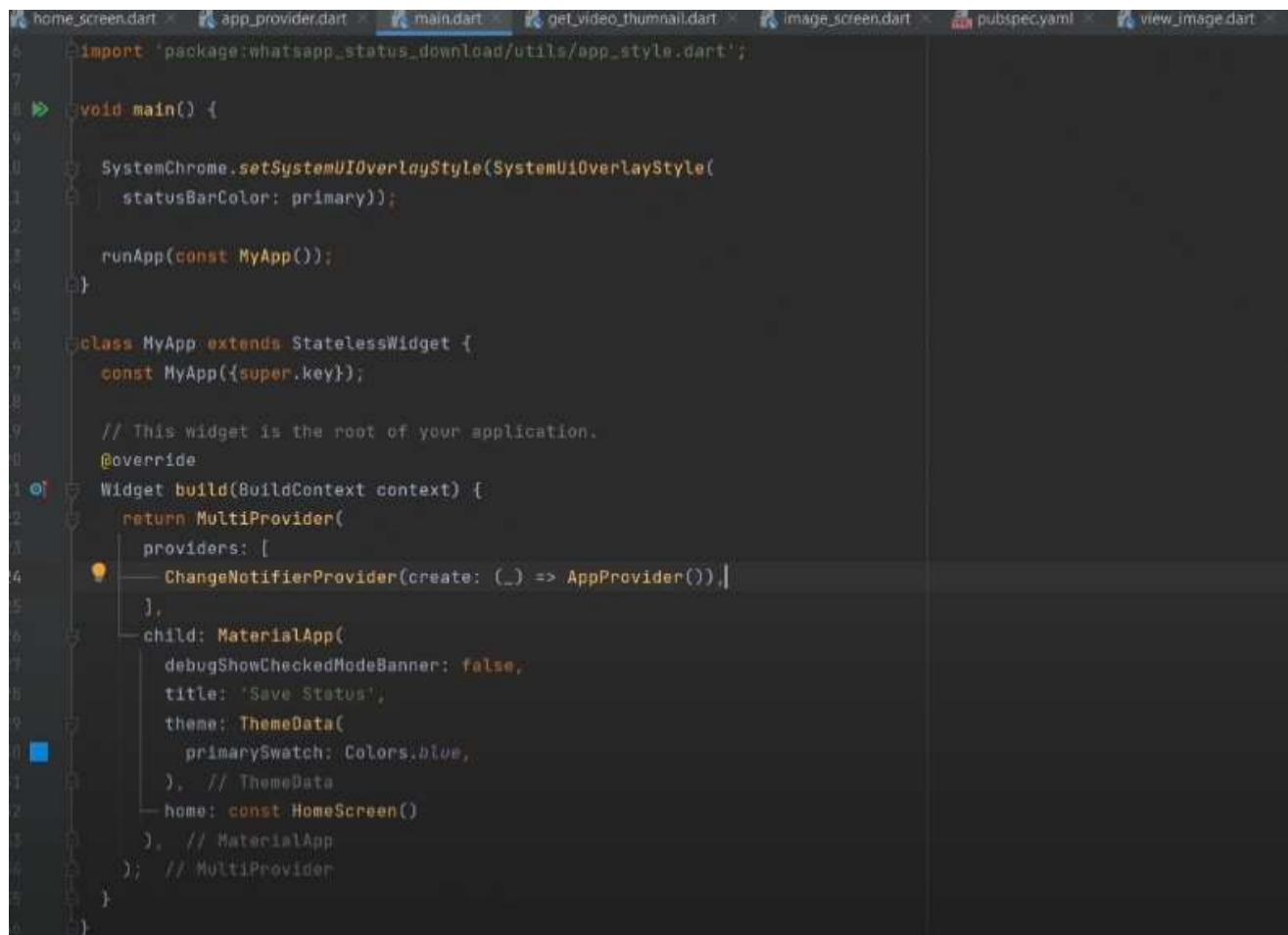
```

home_screen.dart -> app_provider.dart -> get_video_thumbnail.dart -> image_screen.dart -> pubspec.yaml
1 import 'package:video_thumbnail/video_thumbnail.dart';
2
3
4 Future<String> getThumbnail(String path) async{
5
6     String? thumb = await VideoThumbnail.thumbnailFile(video: path);
7
8     return thumb!;
9
10
11 }

```

INTEGRATING services with UI for whatsapp status downloader

Wrapping Provider in main



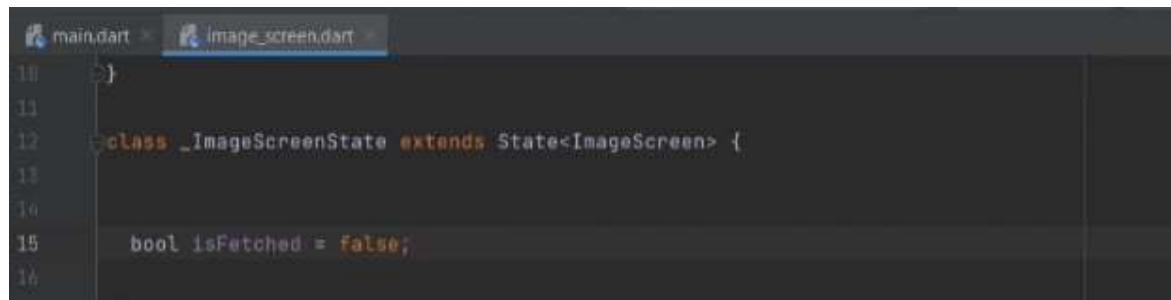
```
import 'package:whatsapp_status_download/utils/app_style.dart';

void main() {
    SystemChrome.setSystemUIOverlayStyle(SystemUiOverlayStyle(
        statusBarColor: primary));
    runApp(const MyApp());
}

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

    // This widget is the root of your application.
    @override
    Widget build(BuildContext context) {
        return MultiProvider(
            providers: [
                ChangeNotifierProvider(create: (_) => AppProvider()),
            ],
            child: MaterialApp(
                debugShowCheckedModeBanner: false,
                title: 'Save Status',
                theme: ThemeData(
                    primarySwatch: Colors.blue,
                ), // ThemeData
                home: const HomeScreen()
            ), // MaterialApp
        ); // MultiProvider
    }
}
```

Updating image_screen.dart



```
}
```

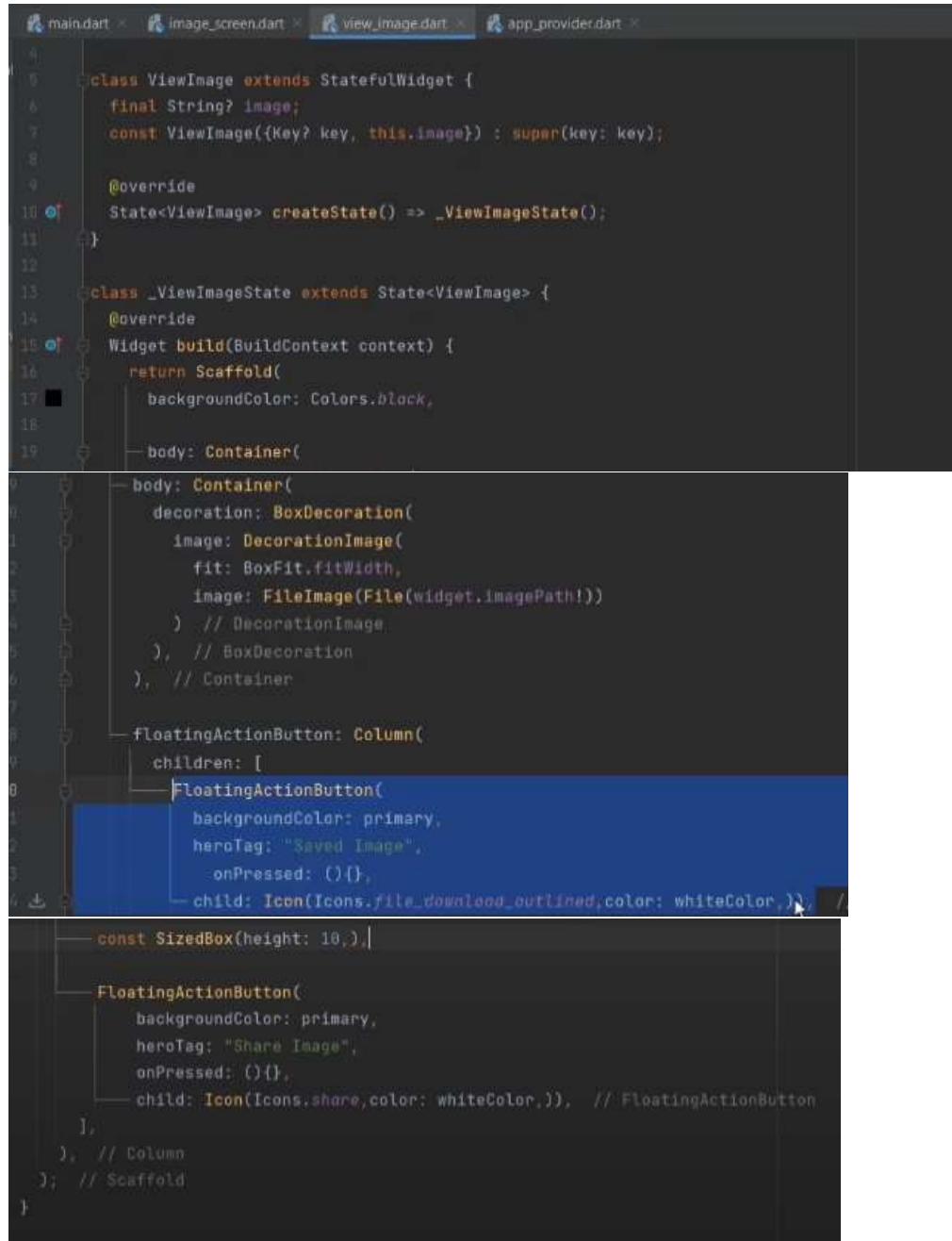
```
class _ImageScreenState extends State<ImageScreen> {
    bool isFetched = false;
```

```
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: Consumer<AppProvider>(
        builder: (context, file, child) {
          if(isFetched == false){
            file.getStatus(".jpg");
            Future.delayed(const Duration(microseconds: 1), (){
              isFetched == true;
            });
          } // Future.delayed
        }
      )
    );
    return file.isWhatsAppAvailable == false?
      Center( child: Text("No WhatsApp available", style: Style.headline2,)):
      file.getImage.isEmpty?
      Center( child: Text("No Image Found", style: Style.headline2,)):
      Padding(padding: const EdgeInsets.symmetric(horizontal: 5),
      child: GridView.builder(
        itemCount: file.getImage.length,
        gridDelegate: const SliverGridDelegateWithFixedCrossAxisCount(
          crossAxisCount: 2,
          crossAxisSpacing: 10.0,
          childAspectRatio: 0.6,
          mainAxisSpacing: 10.0,
        ), // SliverGridDelegateWithFixedCrossAxisCount
        itemBuilder: (context, index){
          return Card(
            elevation: 3,
            child: Column(
              mainAxisAlignment: MainAxisAlignment.spaceBetween,
              children: [
                InkWell(
                  onTap: (){
                    Navigator.push(context, MaterialPageRoute(builder: (context)=> ViewImage(
                      image : file.getImage[index].path));
                  });
                ),
                child: Container(
                  height: MediaQuery.of(context).size.width*0.6,
                  width: MediaQuery.of(context).size.width*0.45,
                  decoration: BoxDecoration(
                    borderRadius: BorderRadius.circular(5),
                    image: DecorationImage(
                      fit: BoxFit.fill,
                      image: FileImage(File(file.getImage[index].path))) // DecorationImage
                  ), // BoxDecoration
                ), // Container
              ], // inkWell
            ),
            Row(
              mainAxisAlignment: MainAxisAlignment.spaceEvenly,
              children: [

```

We passed the Image onPressed while navigating to ViewImage screen

Accessing the Image in ViewImage screen by defining variable and using widget.Image



The screenshot shows a code editor with four tabs at the top: main.dart, image_screen.dart, view_image.dart, and app_provider.dart. The view_image.dart tab is active, displaying the following Dart code:

```
class ViewImage extends StatefulWidget {
    final String? image;
    const ViewImage({key? key, this.image}) : super(key: key);

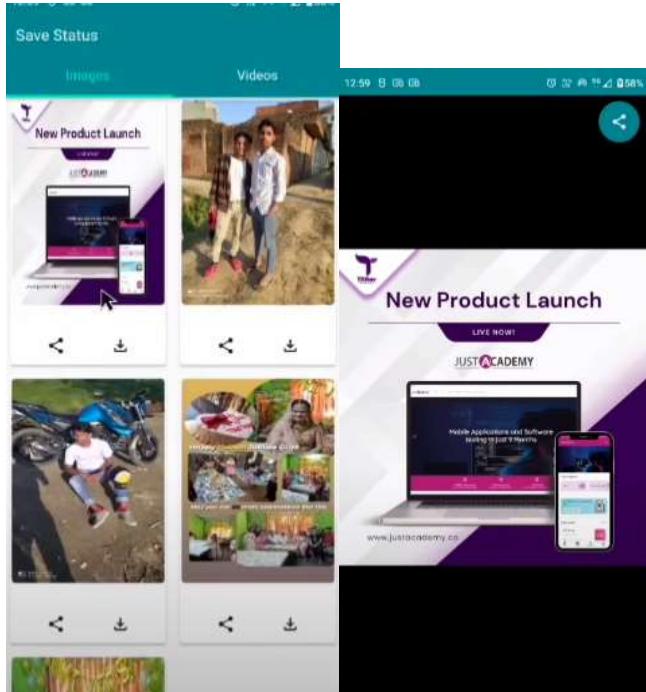
    @override
    State<ViewImage> createState() => _ViewImageState();
}

class _ViewImageState extends State<ViewImage> {
    @override
    Widget build(BuildContext context) {
        return Scaffold(
            backgroundColor: Colors.black,
            body: Container(
                decoration: BoxDecoration(
                    image: DecorationImage(
                        fit: BoxFit.fitWidth,
                        image: FileImage(File(widget.imagePath!))
                    ) // DecorationImage
                ), // BoxDecoration
            ), // Container

            floatingActionButton: Column(
                children: [
                    FloatingActionButton(
                        backgroundColor: primary,
                        heroTag: "Saved Image",
                        onPressed: (){},
                        child: Icon(Icons.file_download_outlined,color: whiteColor,) // FloatingActionButton
                    ),
                    const SizedBox(height: 10,),

                    FloatingActionButton(
                        backgroundColor: primary,
                        heroTag: "Share Image",
                        onPressed: (){},
                        child: Icon(Icons.share,color: whiteColor,), // FloatingActionButton
                    ),
                ],
            ), // Column
        ); // Scaffold
}
```

Output



mainAxisAlignment

```
  FloatingActionButton: Column(
    mainAxisAlignment: MainAxisAlignment.end,
    children: [
      FloatingActionButton(
        backgroundColor: primary,
        heroTag: "Saved Image",
        onPressed: (){},
        child: Icon(Icons.file_download_outlined,color: whiteColor,), // FloatingActionButton

        const SizedBox(height: 10,),

        FloatingActionButton(
          backgroundColor: primary,
          heroTag: "Share Image",
          onPressed: (){},
          child: Icon(Icons.share,color: whiteColor,), // FloatingActionButton
      ),
    ],
  ), // Column
); // Scaffold
}
```



Adding downloading and sharing feature in app

Installing

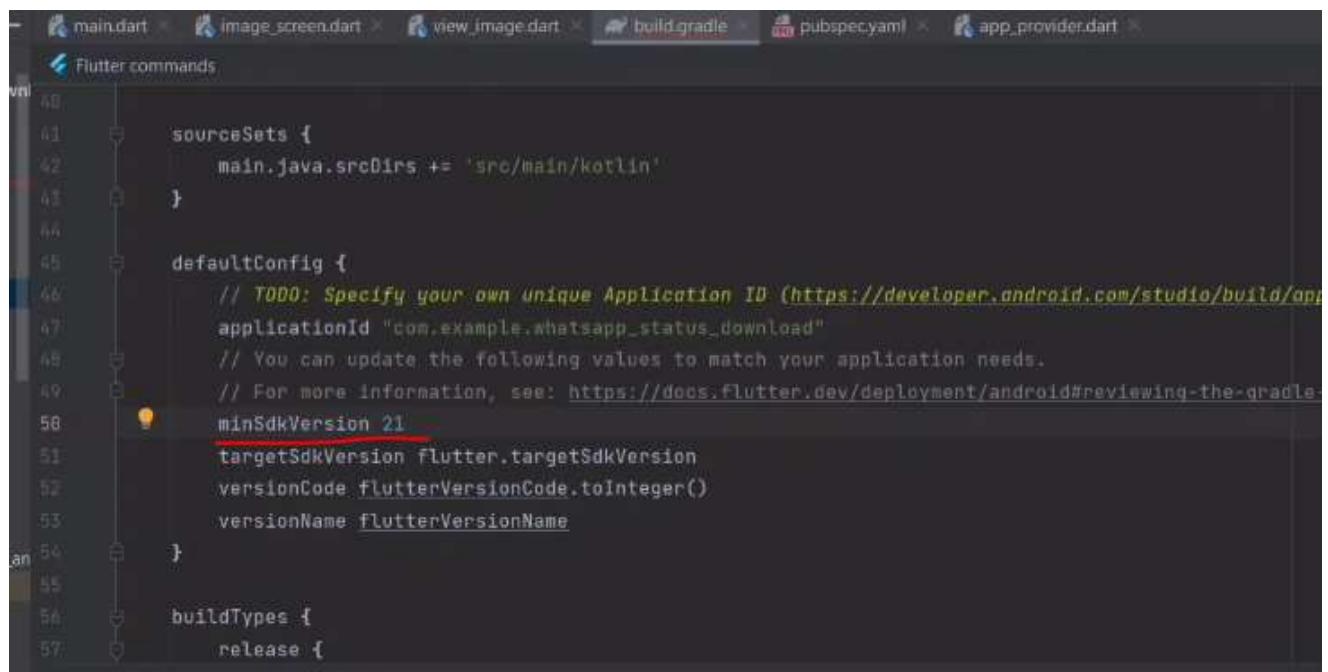
```
Flutter commands
33
34
35      # The following adds the CupertinoIcons font to your application.
36      # Use with the CupertinoIcons class for iOS style icons.
37      cupertino_icons: ^1.0.2
38      fluttertoast: ^8.2.1
39      permission_handler:
40      provider:
41      video_thumbnail:
42      image_gallery_saver:
43      flutter_native_api| ] I
44
45      dev_dependencies:
46      flutter_test:
47      sdk: flutter
48
49      # The "flutter_lints" package below contains a set of recommended lints to
50      # encourage good coding practices. The lint set provided by the package is
51      # activated in the 'analysis_options.yaml' file located at the root of yo
```

view_image.dart

```
- floatingActionButton: Column(
  mainAxisAlignment: MainAxisAlignment.end,
  children: [
    FloatingActionButton(
      backgroundColor: primary,
      heroTag: "Saved Image",
      onPressed: (){
        ImageGallerySaver.saveFile(widget.image!).then((value){
          commToast("Saved Successfully", context);
        }).onError((error, stackTrace){
          commToast("Error", context);
        });
      },
      child: Icon(Icons.file_download_outlined,color: whiteColor,), // FloatingActionButton
    ),
    const SizedBox(height: 10,),
  ],
), // Column
); // Scaffold
}
```

```
FloatingActionButton(
  backgroundColor: primary,
  heroTag: "Share Image",
  onPressed: (){
    FlutterNativeApi.shareImage(widget.image!);
  },
  child: Icon(Icons.share,color: whiteColor,), // FloatingActionButton
),
], // Column
); // Scaffold
}
```

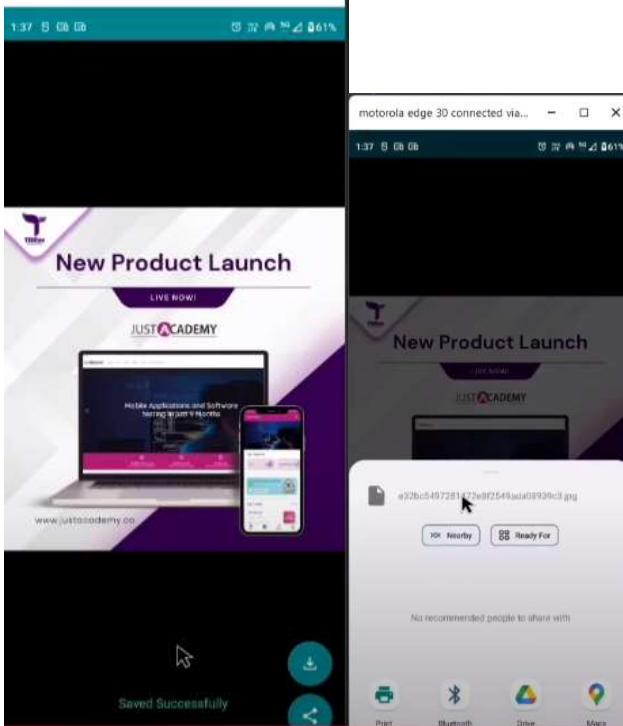
Also changed minSdkVersion to 21



The screenshot shows the Android Studio interface with the build.gradle file open. The code editor highlights the line 'minSdkVersion 21' in red, indicating a potential issue or warning. The file contains standard Gradle configuration for an Android application, including sourceSets, defaultConfig, and buildTypes sections.

```
vnl
41     sourceSets {
42         main.java.srcDirs += 'src/main/kotlin'
43     }
44
45     defaultConfig {
46         // TODO: Specify your own unique Application ID (https://developer.android.com/studio/build/applicationId "com.example.whatsapp_status_download"
47         // You can update the following values to match your application needs.
48         // For more information, see: https://docs.flutter.dev/deployment/android#reviewing-the-gradle
49         minSdkVersion 21
50         targetSdkVersion flutter.targetSdkVersion
51         versionCode flutterVersionCode.toInt()
52         versionName flutterVersionName
53     }
54
55     buildTypes {
56         release {
57             
```

Output



Doing for video

```
    file.getVideo.isEmpty?
    Center( child: Text("No Image Found", style: Style.headline2),)

    Padding(padding: const EdgeInsets.symmetric(horizontal: 5),
    child: GridView.builder(
        itemCount: file.getVideo.length,
        gridDelegate: const SliverGridDelegateWithFixedCrossAxisCount(
            crossAxisCount: 2,
            crossAxisSpacing: 10.0,
            childAspectRatio: 0.6,
            mainAxisSpacing: 10.0,
        ), // SliverGridDelegateWithFixedCrossAxisCount
        itemBuilder: (context, index){

    return FutureBuilder<String>(
        future: getThumbnail(file.getVideo[index].path),
        builder: (context, snapshot) {
            return Card(
                elevation: 3,
                child: Column(
                    mainAxisAlignment: MainAxisAlignment.spaceBetween,
                    children: [
                        InkWell(
                            onTap: (){
                                Navigator.push(context, MaterialPageRoute(builder: (context)=> ViewVideo(video :file.getVideo[index].path)));
                            },
                        ),
                    ],
                ),
            );
        },
    );
}); // Padding
}); // Gridview
}); // Padding
```

Card in below can be seen in above image

```
return FutureBuilder<String>(
    future: getThumbnail(file.getVideo[index].path),
    builder: (context, snapshot) {
        if(snapshot.hasData){
            return Card(...); // Card
        }else{
            return const SizedBox();
        }
    });
}); // FutureBuilder
}); // Gridview
}); // Padding
```

And view_video.dart

```
main.dart image_screen.dart view_image.dart videos_screen.dart view_video.dart

class ViewVideo extends StatefulWidget {
    final String? video;
    const ViewVideo({Key? key, this.video}) : super(key: key);

    @override
    State<ViewVideo> createState() => _ViewVideoState();
}

class _ViewVideoState extends State<ViewVideo> {
    @override
    Widget build(BuildContext context) {
        return Scaffold(
            backgroundColor: Colors.black,
            floatingActionButton: Column(
                mainAxisAlignment: MainAxisAlignment.end,
                children: [
                    FloatingActionButton(
                        backgroundColor: primary,
                        heroTag: "Saved Image",
                        onPressed: () {
                            ImageGallerySaver.saveFile(widget.video!).then((value) {
                                commToast("Saved Successfully", context);
                            }).onError((error, stackTrace) {
                                commToast("Error", context);
                            });
                        },
                        child: Icon(Icons.file_download_outlined, color: whiteColor,)),
                ],
                const SizedBox(height: 10),
                Row(
                    mainAxisAlignment: MainAxisAlignment.spaceEvenly,
                    children: [
                        IconButton(onPressed: () {FlutterNativeApi.shareImage(file.getVideo[index].path); },
                            icon: const Icon(Icons.share)), // IconButton
                        IconButton(onPressed: () {ImageGallerySaver.saveFile(file.getVideo[index].path).then((value) {
                            commToast("Saved Successfully", context);
                        });
                        },
                            icon: const Icon(Icons.file_download_outlined)), // IconButton
                    ],
                ) // Row
            ],
        ); // Card
    }
    return const SizedBox();
}
```

To play video using package chewie

Installing

```
# The following adds the Cupertino Icons font to your application.
# Use with the CupertinoIcons class for iOS style icons.
cupertino_icons: ^1.0.2
fluttertoast: ^8.2.1
permission_handler:
provider:
video_thumbnail:
image_gallery_saver:
flutter_native_api:
chewie:
video_player:
```

Updating view_video.dart

```
main.dart image_screen.dart view_image.dart videos_screen.dart view_video.dart pubspec.yaml app_pro
1 import 'package:flutter/material.dart';
2 import 'package:flutter_native_api/flutter_native_api.dart';
3 import 'package:image_gallery_saver/image_gallery_saver.dart'; I
4 import 'package:chewie/chewie.dart';
5 import '../utils/app_style.dart';
6 import '../utils/common_toast.dart';
7 import 'package:video_player/video_player.dart';

8
9 class ViewVideo extends StatefulWidget {
10   final String? video;
11   const ViewVideo({Key? key, this.video}) : super(key: key);
12
13   @override
14   State<ViewVideo> createState() => _ViewVideoState();
15 }
16
17 class _ViewVideoState extends State<ViewVideo> {
```

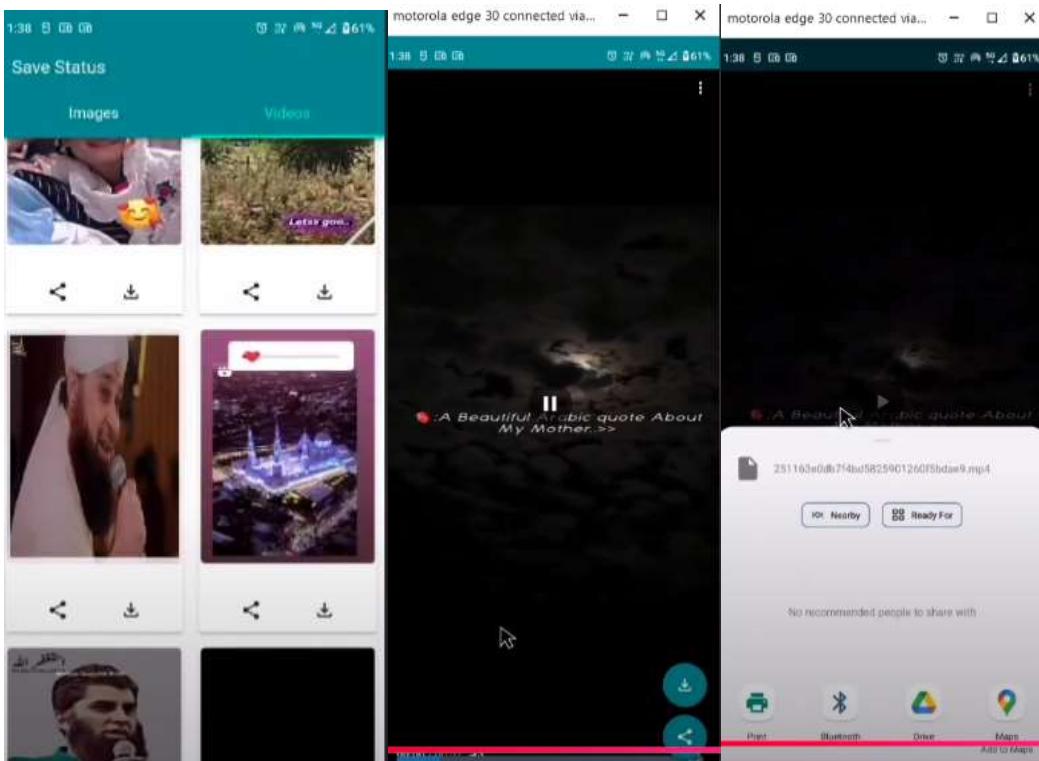
```
ChewieController? _chewieController;

@Override
void initState() {
  _chewieController = ChewieController(
    autoInitialize: true,
    looping: true,
    aspectRatio: 6/7,
    errorBuilder: ((context, errorMessage){
      return commToast(errorMessage, context); I
    }),
    videoPlayerController: VideoPlayerController.file(File(widget.video!),
  )) ; // VideoPlayerController.file, ChewieController
super.initState();
}

@Override
Widget build(BuildContext context) {
```

```
@override  
Widget build(BuildContext context) {  
  return Scaffold(  
    backgroundColor: Colors.black,  
  
    body: Chewie(controller: _chewieController!),  
  
    floatingActionButton: Column(  
      mainAxisAlignment: MainAxisAlignment.end,
```

Output

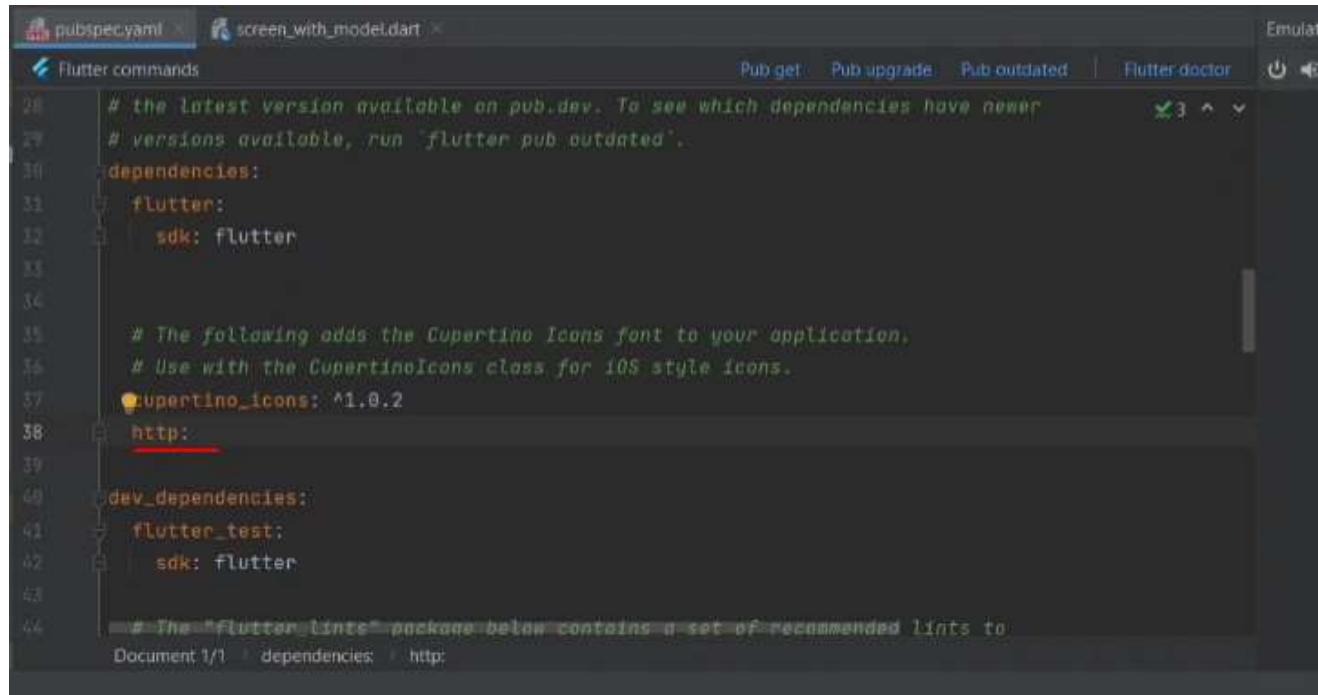


Api integration

Animation

API INTEGRATION

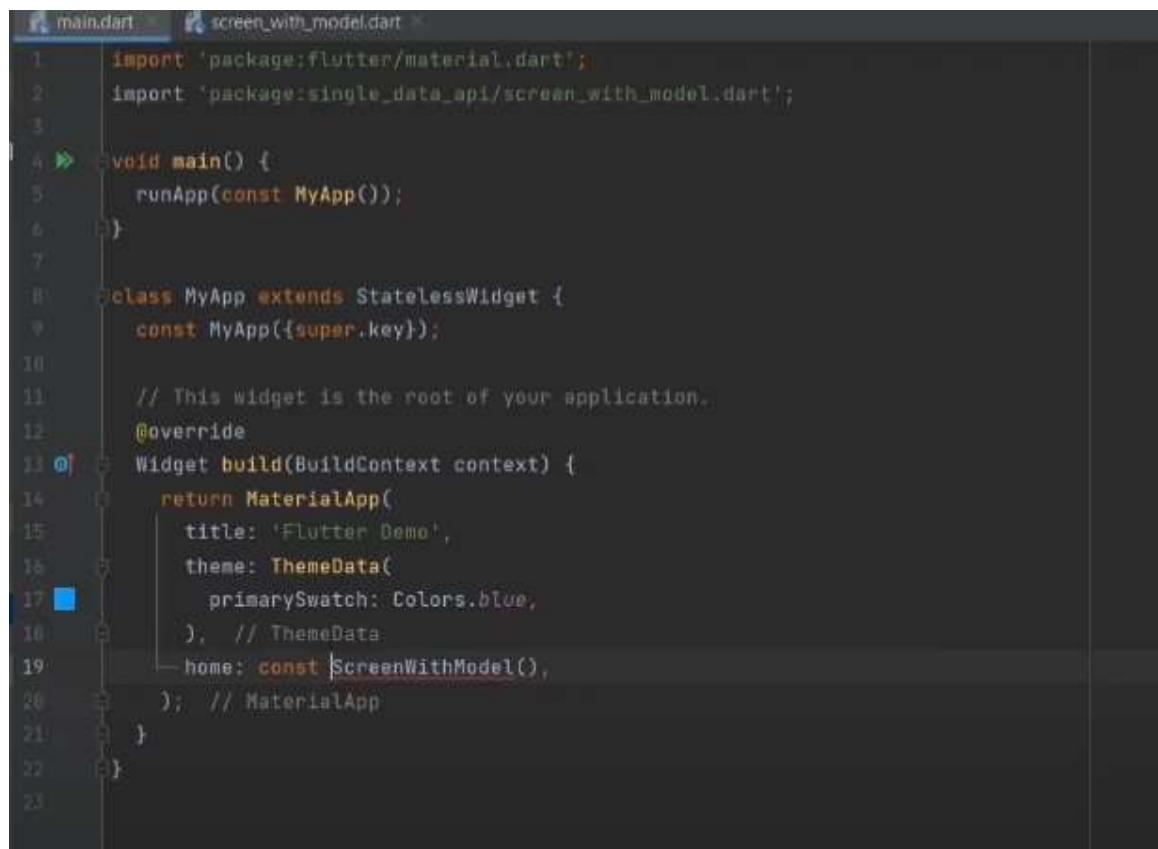
Install “http” by adding in “pubspec.yaml” file



```
28 # the latest version available on pub.dev. To see which dependencies have newer
29 # versions available, run `flutter pub outdated`.
30 dependencies:
31   flutter:
32     sdk: flutter
33
34
35   # The following adds the Cupertino Icons font to your application.
36   # Use with the CupertinoIcons class for iOS style icons.
37   cupertino_icons: ^1.0.2
38   http:           
39
40   dev_dependencies:
41     flutter_test:
42       sdk: flutter
43
44   # The "flutter_lints" package below contains a set of recommended lints to
Document 1/1  dependencies: http:
```

1. SINGLE API

a) With model



```
1 import 'package:flutter/material.dart';
2 import 'package:single_data_api/screen_with_model.dart';
3
4 void main() {
5   runApp(const MyApp());
6 }
7
8 class MyApp extends StatelessWidget {
9   const MyApp({super.key});
10
11   // This widget is the root of your application.
12   @override
13   Widget build(BuildContext context) {
14     return MaterialApp(
15       title: 'Flutter Demo',
16       theme: ThemeData(
17         primarySwatch: Colors.blue,
18       ), // ThemeData
19       home: const ScreenWithModel(),
20     ); // MaterialApp
21   }
22 }
```

<https://jsonplaceholder.typicode.com/posts/1>

GET https://jsonplaceholder.typicode.com/posts/1

Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies

Query Params

Key	Value	Description		Bulk Edit
Key	Value	Description		

Body Cookies Headers (24) Test Results

Pretty Raw Preview Visualize JSON ↻

```

1  {
2     "userId": 1,
3     "id": 1,
4     "title": "sunt aut facere repellat provident occaecati excepturi optio reprehenderit",
5     "body": "quia et suscipit\\nsuscipit recusandae consequuntur expedita et cum\\nreprehenderit molestiae ut ut quas totam\\nnostrum rerum est
6         autem sunt rem eveniet architecto"
7 }
```

Status: 200 OK Time: 1237 ms Size: 1.24 KB Save Response ↻

using reponse to make model

https://javiercbk.github.io/json_to_dart/

JSON to Dart

Paste your JSON in the textarea below, click convert and get your Dart classes for free.

JSON

```

1  {
2     "userId": 1,
3     "id": 1,
4     "title": "sunt aut facere repellat provident occaecati excepturi optio reprehenderit",
5     "body": "quia et suscipit\\nsuscipit recusandae consequuntur expedita et cum\\nreprehenderit molestiae ut ut quas totam\\nnostrum rerum est
6         autem sunt rem eveniet architecto"
7 }
```

class SinglePostWithModel {
 int? userId;
 int? id;
 String? title;
 String? body;

SinglePostWithModel({this.userId, this.id, this.title, this.body});

SinglePostWithModel.fromJson(Map<String, dynamic> json) {
 userId = json['userid'];
 id = json['id'];
 title = json['title'];
 body = json['body'];
}

Map<String, dynamic> toJson() {
 final Map<String, dynamic> data = new Map<String, dynamic>();
 data['userId'] = this.userId;
 data['id'] = this.id;
 data['title'] = this.title;
 data['body'] = this.body;
 return data;

SinglePostWithModel

Generate Dart Use private fields

Copy Dart code to clipboard

```
pubspec.yaml screen_with_model.dart api.services.dart single_post_model.dart
1 class SinglePostWithModel {
2     int? userId;
3     int? id;
4     String? title;
5     String? body;
6
7     SinglePostWithModel({this.userId, this.id, this.title, this.body});
8
9     SinglePostWithModel.fromJson(Map<String, dynamic> json) {
10         userId = json['userId'];
11         id = json['id'];
12         title = json['title'];
13         body = json['body'];
14     }
15
16     Map<String, dynamic> toJson() {
17         final Map<String, dynamic> data = new Map<String, dynamic>();
18         data['userId'] = this.userId;
19         data['id'] = this.id;
20         data['title'] = this.title;
21         data['body'] = this.body;
22         return data;
23     }
24 }
25
```

ApiServices.dart file

```
pubspec.yaml screen_with_model.dart api.services.dart
5
6 class ApiServices{
7
8     // With Model
9     Future<SinglePostWithModel?> getSinglePostWithModel()async{
10
11         try{
12
13             var response = await http.get(Uri.parse("https://jsonplaceholder.typicode.com/posts/1"));
14             if(response.statusCode == 200){
15                 SinglePostWithModel model = SinglePostWithModel.fromJson(json.decode(response.body));
16                 return model;
17             }
18         }catch (e){
19             print(e.toString());
20         }
21         return null;
22
23     }
24 }
```

USING API

```
import 'package:flutter/material.dart';
import 'package:single_data_api/api_services.dart';
import 'package:single_data_api/single_post_model.dart';

class ScreenWithModel extends StatefulWidget {
  const ScreenWithModel({Key? key}) : super(key: key);

  @override
  State<ScreenWithModel> createState() => _ScreenWithModelState();
}

class _ScreenWithModelState extends State<ScreenWithModel> {

  bool isReady = false;
  SinglePostWithModel singlePostWithModel = SinglePostWithModel();
  _getSinglePost(){
    isReady = true;
    ApiServices().getSinglePostWithModel().then((value){
      setState(() {
        singlePostWithModel = value!;
        isReady = false;
      });
    }).onError((error, stackTrace){
      print(error);
      setState(() {
        isReady = false;
      });
    });
  }
}

@Override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      centerTitle: true,
      title: const Text("Single"),
    ), // AppBar

    body: isReady == true?
    const Center(child: CircularProgressIndicator(),):

    Column(
      mainAxisAlignment: MainAxisAlignment.center,
      children: [
        Text(singlePostWithModel.userId.toString(), style: const TextStyle(fontSize: 20, color: Colors.purple)),
        Text(singlePostWithModel.title.toString(), style: const TextStyle(fontSize: 18, color: Colors.pink)),
        Text(singlePostWithModel.body.toString(), style: const TextStyle(fontSize: 16, color: Colors.brown))
      ], // Column
  ); // Scaffold
}
```

OUTPUT



b)Single Api without model

```
Future<dynamic> getSinglePostWithoutModel()async{
try{

    var response = await http.get(Uri.parse("https://jsonplaceholder.typicode.com/posts/1"));
    if(response.statusCode == 200){

        final body = response.body;
        final data = jsonDecode(body);
        return data;
    }
} catch (e){
    print(e.toString());
}
return null;
}
```

Update main.dart

```
1 import 'package:flutter/material.dart';
2 import 'package:single_data_api/screen_with_model.dart';
3 import 'package:single_data_api/screen_without_model.dart';
4
5 void main() {
6     runApp(const MyApp());
7 }
8
9 class MyApp extends StatelessWidget {
10     const MyApp({super.key});
11
12     // This widget is the root of your application.
13     @override
14     Widget build(BuildContext context) {
15         return MaterialApp(
16             title: 'Flutter Demo',
17             theme: ThemeData(
18                 primarySwatch: Colors.blue,
19             ), // ThemeData
20             home: const ScreenWithoutModel(),
21         ); // MaterialApp
22     }
23 }
```

Using api

```
5   class ScreenWithoutModel extends StatefulWidget {
6     const ScreenWithoutModel({Key? key}) : super(key: key);
7
8     @override
9     State<ScreenWithoutModel> createState() => _ScreenWithoutModelState();
10  }
11
12  class _ScreenWithoutModelState extends State<ScreenWithoutModel> {
13
14
15    dynamic singlePost;
16    bool isReady = false;
17
18    _getSinglePostWithoutModel(){
19      isReady = true;
20      ApiServices().getSinglePostWithoutModel().then((value){
21        setState(() {
22          singlePost = value;
23          isReady = false;
24        });
25      }).onError((error, stackTrace){
26        print(error);
27        isReady = false;
28      });
29    }
30  }
```

```
33
34  @override
35  void initState() {
36    _getSinglePostWithoutModel();
37    super.initState();
38  }
```

```
@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      centerTitle: true,
      title: const Text("Post Without Model"),
    ), // AppBar

    body: isReady == true?
      const Center(child: CircularProgressIndicator(),):

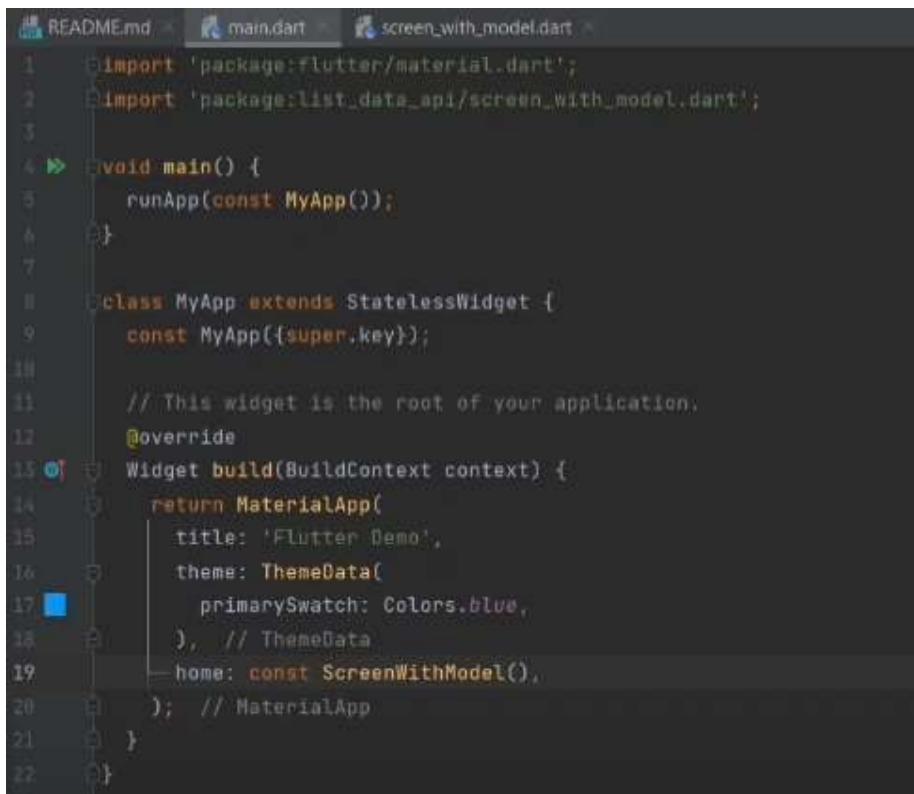
    Column(
      mainAxisAlignment: MainAxisAlignment.center,
      children: [
        Text(singlePost["userId"].toString(), style: TextStyle(color: Colors.pink, fontSize: 18)),
        Text(singlePost["title"].toString(), style: TextStyle(color: Colors.black, fontSize: 18)),
        Text(singlePost["body"].toString(), style: TextStyle(color: Colors.teal, fontSize: 18)),
      ],
    ), // Column
  ); // Scaffold
}
```

output

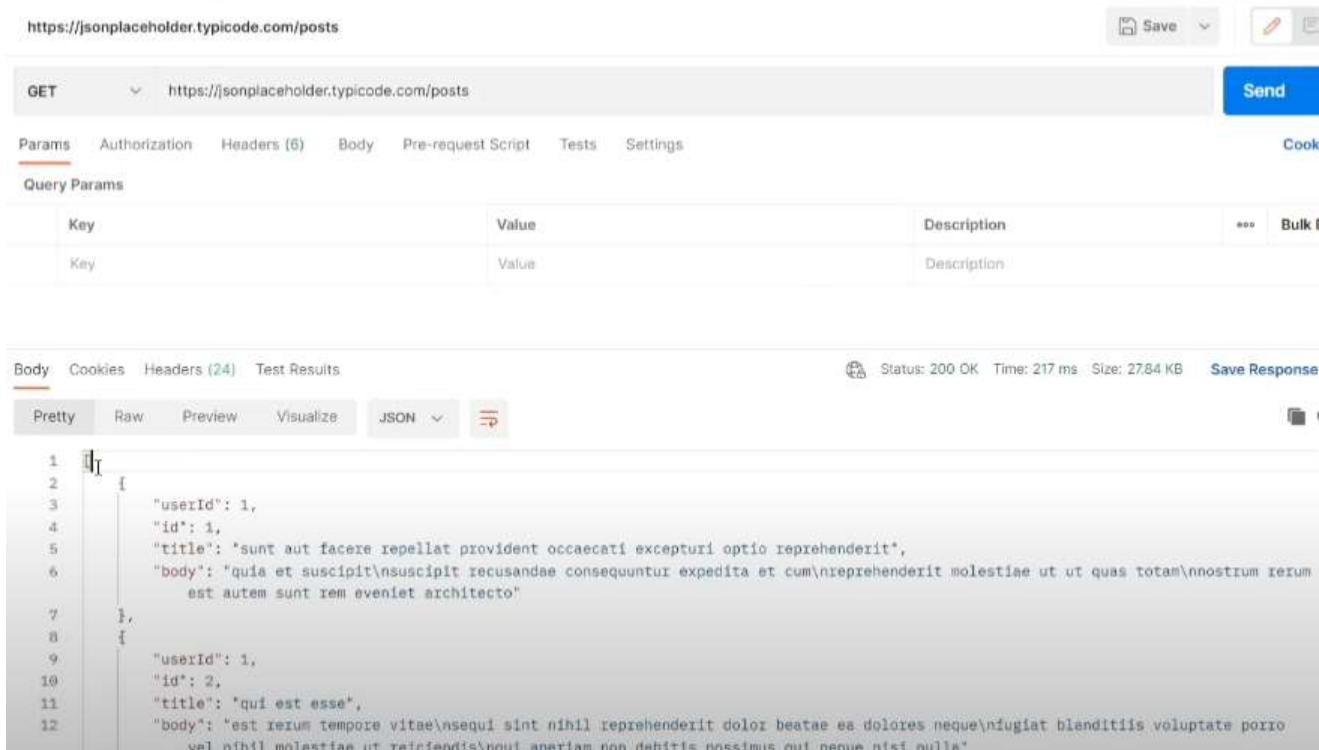


2. List api

a) With model



```
1 import 'package:flutter/material.dart';
2 import 'package:list_date_api/screen_with_model.dart';
3
4 void main() {
5   runApp(const MyApp());
6 }
7
8 class MyApp extends StatelessWidget {
9   const MyApp({super.key});
10
11   // This widget is the root of your application.
12   @override
13   Widget build(BuildContext context) {
14     return MaterialApp(
15       title: 'Flutter Demo',
16       theme: ThemeData(
17         primarySwatch: Colors.blue,
18       ), // ThemeData
19       home: const ScreenWithModel(),
20     ); // MaterialApp
21   }
22 }
```



https://jsonplaceholder.typicode.com/posts

Save   

GET https://jsonplaceholder.typicode.com/posts Send

Params Authorization Headers (5) Body Pre-request Script Tests Settings Cook

Query Params

Key	Value	Description	...	Bulk
Key	Value	Description	...	

Body Cookies Headers (24) Test Results

Pretty Raw Preview Visualize JSON 

```
1 [
2   {
3     "userId": 1,
4     "id": 1,
5     "title": "sunt aut facere repellat provident occaecati excepturi optio reprehenderit",
6     "body": "quia et suscipit\\nsuscipit recusandae consequuntur expedita et cum\\nreprehenderit molestiae ut ut quas totam\\nnostrum rerum est autem sunt rem eveniet architecto"
7   },
8   {
9     "userId": 1,
10    "id": 2,
11    "title": "qui est esse",
12    "body": "est ierum tempore vitae\\nsequi sint nihil reprehenderit dolor beatue ea dolores neque\\niugiat blanditiis voluptate porro vel nihil molestiae ut reiciendis\\nqui aperiam non debitis possimus qui neque nisi nulla"
13  }
]
```

Making model of api response

JSON to Dart

Paste your JSON in the textarea below, click convert and get your Dart classes for free.

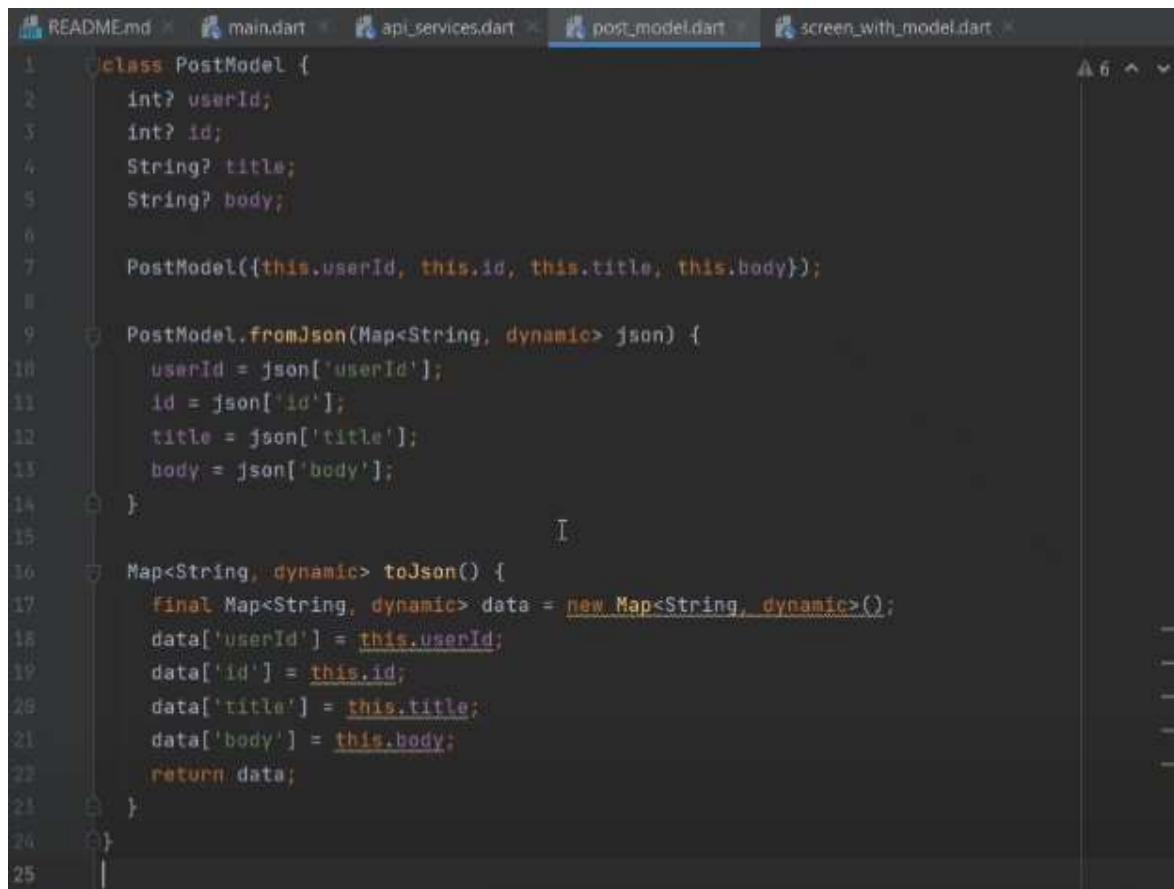
JSON

```
583 },
584 {
585   "userId": 10,
586   "id": 98,
587   "title": "laboriosam dolor volupta",
588   "body": "doloremque ex facilis sit",
589 },
590 {
591   "userId": 10,
592   "id": 99,
593   "title": "temporibus sit alias del",
594   "body": "quo deleniti praesentium",
595 },
596 {
597   "userId": 10,
598   "id": 100,
599   "title": "at nam consequatur ea la",
600   "body": "cupiditate quo est a modi",
601 }
```

PostModel

Use private fields

post_model.dart



```
class PostModel {
    int? userId;
    int? id;
    String? title;
    String? body;

    PostModel({this.userId, this.id, this.title, this.body});

    PostModel.fromJson(Map<String, dynamic> json) {
        userId = json['userId'];
        id = json['id'];
        title = json['title'];
        body = json['body'];
    }

    Map<String, dynamic> toJson() {
        final Map<String, dynamic> data = new Map<String, dynamic>();
        data['userId'] = this.userId;
        data['id'] = this.id;
        data['title'] = this.title;
        data['body'] = this.body;
        return data;
    }
}
```

ApiServices.dart

```
= 200){  
= List<PostModel>.from(json.decode(response.body).map((x)=> PostModel.fromJson(x)));|
```

The screenshot shows a code editor window with several tabs at the top: README.md, main.dart, api_services.dart (which is the active tab), post_model.dart, and screen_with_model.dart. The code in the editor is as follows:

```
import 'package:list_data_api/post_model.dart';  
import 'package:http/http.dart' as http;  
  
class ApiServices{  
  
  Future<List<PostModel>?> getPostWithModel(){  
  
    try{  
  
      var response = await http.get(Uri.parse("https://jsonplaceholder.typicode.com/post"));  
  
      if(response.statusCode == 200){  
        List<PostModel> model = List<PostModel>.from(json.decode(response.body).map((x)=>  
          return model;|  
    }  
  
    catch (e){  
      print(e.toString());  
    }  
  
    return null;  
  }  
}
```

Using api

```
 9  @override
10  State<ScreenWithModel> createState() => _ScreenWithModelState();
11 }
12
13 class _ScreenWithModelState extends State<ScreenWithModel> {
14
15   bool isReady = false;
16   List<PostModel> postModel = [];
17   _getPost(){
18     isReady = true;
19     ApiServices().getPostWithModel().then((value){
20       setState(() {
21         postModel = value!;
22         isReady = false;
23       });
24     });
25   }
26
27   @override
28   void initState() {
29     _getPost();
30     super.initState();
31   }

```

```
return Scaffold(
  appBar: AppBar(
    centerTitle: true,
    title: const Text("Posts With Model"),
  ), // AppBar

  body: isReady == true?
  const Center(child: CircularProgressIndicator(),):

  ListView.builder(
    itemCount: postModel.length,
    itemBuilder: (context, index){
      return Card(
        child: ListTile(
          leading: Text(postModel[index].userId.toString()),
          title: Text(postModel[index].title.toString()),
          subtitle: Text(postModel[index].body.toString()),
        ), // ListTile
      ); // Card
    }, // ListView.builder
  ); // Scaffold
}
```

Output



b) Without model

api-services.dart

```
Future<dynamic> getPostWithOutModel() async {
  try{
    var response = await http.get(Uri.parse("https://jsonplaceholder.typicode.com/posts"));
    if(response.statusCode == 200){
      final model = jsonDecode(response.body);
      return model;
    }
  }catch (e){
    print(e.toString());
  }

  return null;
}
```

main.dart

```
main.dart  api_services.dart  screen_without_model.dart ×
1 import 'package:flutter/material.dart';
2 import 'package:list_data_api/screen_with_model.dart';
3
4 void main() {
5   runApp(const MyApp());
6 }
7
8 class MyApp extends StatelessWidget {
9   const MyApp({super.key});
10
11   // This widget is the root of your application.
12   @override
13   Widget build(BuildContext context) {
14     return MaterialApp(
15       title: 'Flutter Demo',
16       theme: ThemeData(
17         primarySwatch: Colors.blue,
18       ), // ThemeData
19       home: const ScreenWithoutModel(),
20     ); // MaterialApp
21   }
22 }
```

```
main.dart <-- api_services.dart <-- screen_without_model.dart
```

```
3 import 'api_services.dart';
4
5
6 class ScreenWithOutModel extends StatefulWidget {
7     const ScreenWithOutModel({Key? key}) : super(key: key);
8
9
10 @override
11     State<ScreenWithOutModel> createState() => _ScreenWithOutModelState();
12 }
13
14 class _ScreenWithOutModelState extends State<ScreenWithOutModel> {
15
16     bool isReady = false;
17     dynamic postList = [];
18     _getPost(){
19         isReady = true;
20         ApiServices().getPostWithOutModel().then((value){
21             setState(() {
22                 postList = value;
23                 isReady = false;
24             });
25         }).onError((error, stackTrace){
26             print(error);
27         });
28     }
29
30
31     @override
32     void initState() {
33         _getPost();
```

```

@Override
void initState() {
    _getPost();
    super.initState();
}

@Override
Widget build(BuildContext context) {
    return Scaffold(
        appBar: AppBar(
            centerTitle: true,
            title: Text("Posts WithOut Model"),
        ), // AppBar
        body: isReady == true?
            const Center(child: CircularProgressIndicator(),):
            ListView.builder(
                itemCount: postList.length,
                itemBuilder: (context, index){
                    return Card(
                        child: ListTile(
                            leading: Text(postList[index]["id"].toString()),
                            title: Text(postList[index]["title"].toString()),
                            subtitle: Text(postList[index]["body"].toString()),
                        ), // ListTile
                    ); // Card
                }) // ListView.builder
}

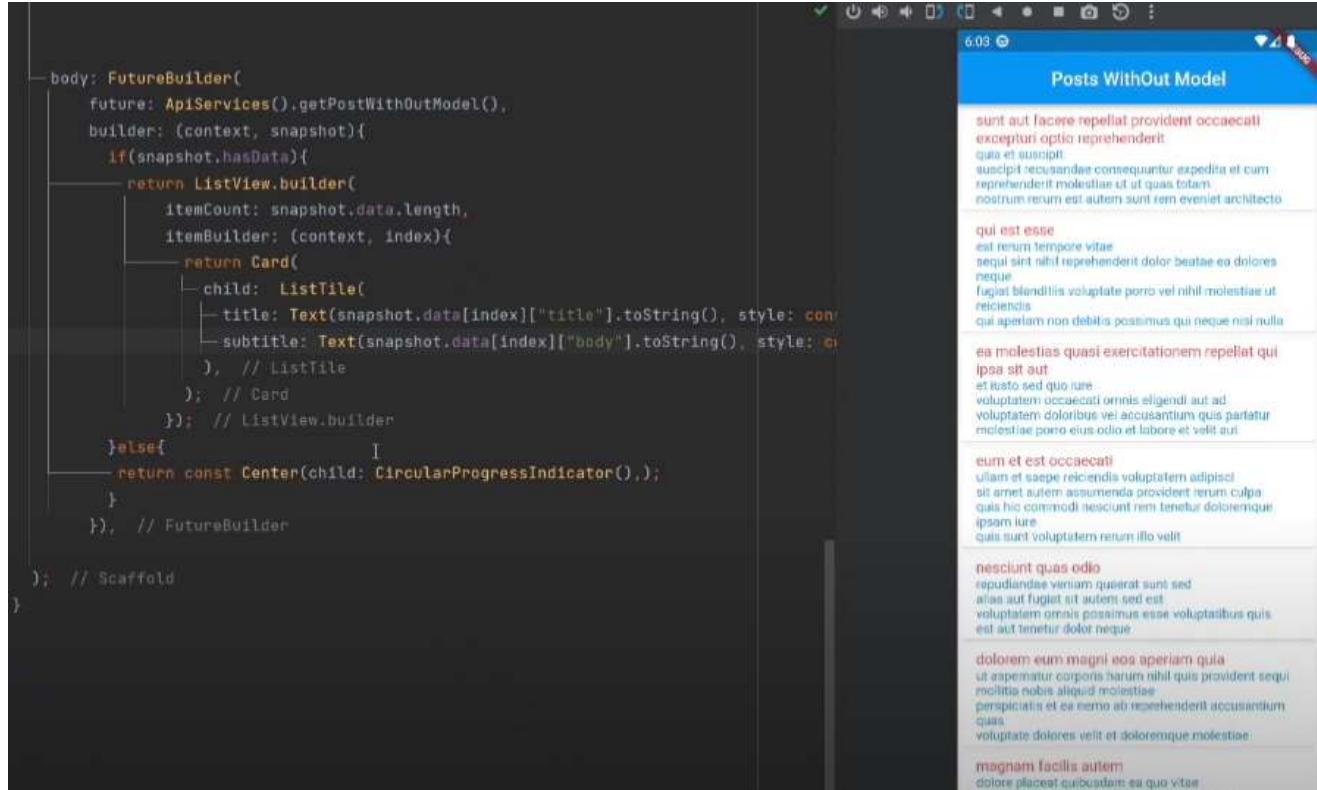
```

Output



Using Future Builder without handling api

- Used to show loader on particular section of screen not on full screen since we don't use isLoading inside initState



c) List inside normal response

The screenshot shows the Reqres API documentation and a Postman request for the endpoint `/api/unknown`.

Reqres API Documentation:

- Request: `/api/unknown`
- Response: 200

Postman Request:

- Method: GET
- URL: `https://reqres.in/api/unknown`
- Headers:
 - Content-Type: application/json
- Body (JSON):

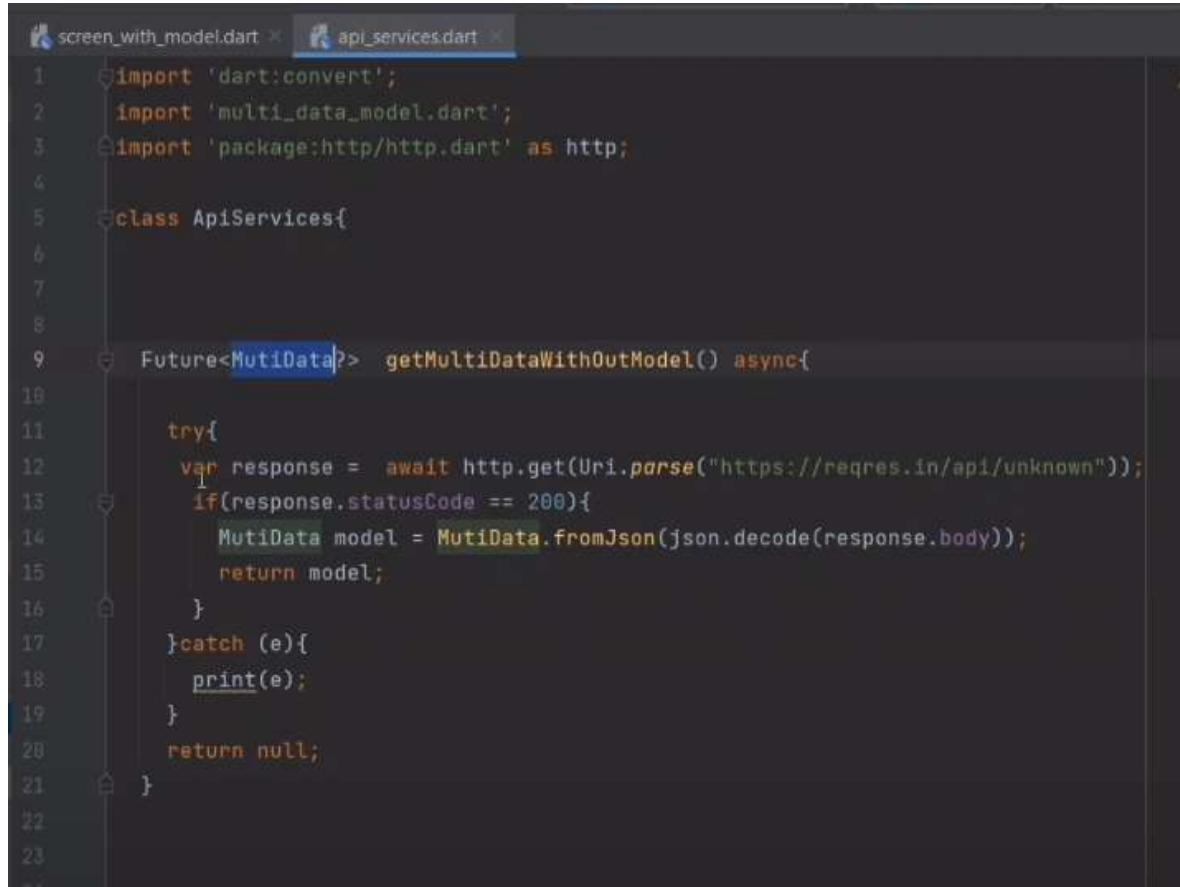
```
{  
  "page": 1,  
  "per_page": 6,  
  "total": 12,  
  "total_pages": 2,  
  "data": [  
    {  
      "id": 1,  
      "name": "cerulean",  
      "year": 2000,  
      "color": "#9882D1",  
      "pantone_value": "15-4020"  
    },  
    {  
      "id": 2,  
      "name": "fuchsia rose",  
      "year": 2001,  
      "color": "#C74375",  
      "pantone_value": "17-2031"  
    },  
    {  
      "id": 3,  
      "name": "true red",  
      "year": 2002,  
      "color": "#E63935",  
      "pantone_value": "19-1646"  
    }]  
}
```
- Tests:
 - Assertion: Status: 200 OK
- Settings:
 - Cookies: (empty)

Model

```
1  class MutiData {
2      int? page;
3      int? perPage;
4      int? total;
5      int? totalPages;
6      List<Data>? data;
7      Support? support;
8
9      MutiData(
10          {this.page,
11          this.perPage,
12          this.total,
13          this.totalPages,
14          this.data,
15          this.support});
16
17      MutiData.fromJson(Map<String, dynamic> json) {
18          page = json['page'];
19          perPage = json['per_page'];
20          total = json['total'];
21          totalPages = json['total_pages'];
22          if (json['data'] != null) {
23              data = <Data>[];
24              json['data'].forEach((v) {
25                  data!.add(new Data.fromJson(v));
26              });
27          }
28          support =
29          json['support'] != null ? new Support.fromJson(json['support']) : null;
30      }
31  }
```

```
65      Map<String, dynamic> toJson() {
66          final Map<String, dynamic> data = new Map<String, dynamic>();
67          data['id'] = this.id;
68          data['name'] = this.name;
69          data['year'] = this.year;
70          data['color'] = this.color;
71          data['pantone_value'] = this.pantoneValue;
72          return data;
73      }
74  }
75
76  class Support {
77      String? url;
78      String? text;
79
80      Support({this.url, this.text});
81
82      Support.fromJson(Map<String, dynamic> json) {
83          url = json['url'];
84          text = json['text'];
85      }
86
87      Map<String, dynamic> toJson() {
88          final Map<String, dynamic> data = new Map<String, dynamic>();
89          data['url'] = this.url;
90          data['text'] = this.text;
91          return data;
92      }
93  }
94  }
```

api-services.dart



```
1 import 'dart:convert';
2 import 'multi_data_model.dart';
3 import 'package:http/http.dart' as http;
4
5 class ApiService{
6
7
8
9     Future<MultiData?> getMultiDataWithOutModel() async{
10
11         try{
12             var response = await http.get(Uri.parse("https://reqres.in/api/unknown"));
13             if(response.statusCode == 200){
14                 MultiData model = MultiData.fromJson(json.decode(response.body));
15                 return model;
16             }
17         }catch (e){
18             print(e);
19         }
20         return null;
21     }
22
23 }
```

Using

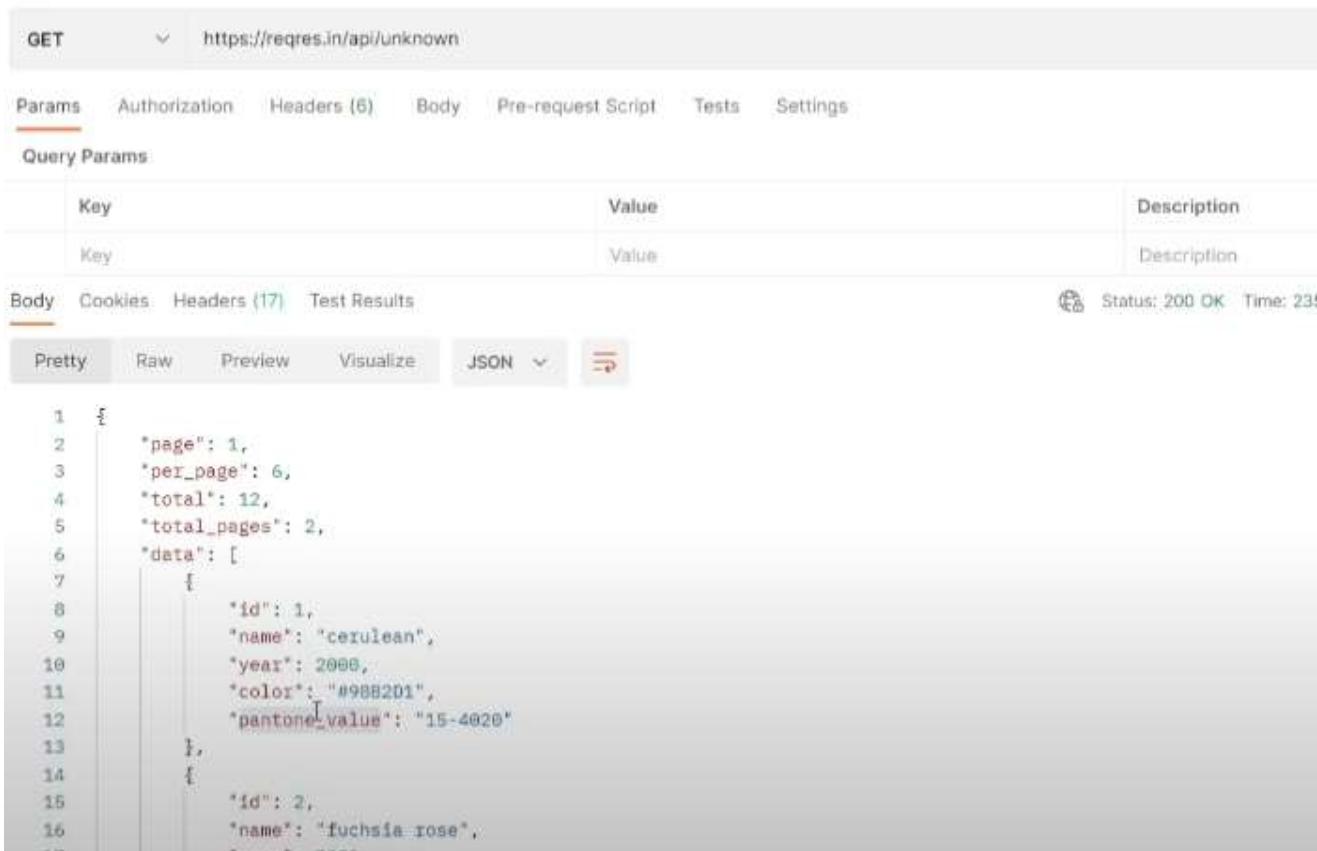
```
screen_with_model.dart ✘ multi_data_model.dart ✘ api.services.dart ✘
```

```
9  @override
10 } State<ScreenWithModel> createState() => _ScreenWithModelState();
11 }
12
13 class _ScreenWithModelState extends State<ScreenWithModel> {
14
15     MultiData multiData = MultiData();
16     _getMultiData(){
17         ApiServices().getMultiDataWithoutModel().then((value){
18             setState(() {
19                 multiData = value!;
20             });
21         }).onError((error, stackTrace){
22             print(error);
23         });
24     }
25
26
27
28     @override
29     void initState() {
30         _getMultiData();
31         super.initState();
32     }
33
34
35     Column(
36         children: [
37             Text(multiData.page.toString(), style: const TextStyle(color: Colors.red, fontSize: 24)),
38             Text(multiData.total.toString(), style: const TextStyle(color: Colors.teal, fontS
39             Text(multiData.totalPages.toString(), style: const TextStyle(color: Colors.teal, fontS
40             Text(multiData.support!.text.toString(), style: const TextStyle(color: Colors.teal, fontS
41
42
43
44         Expanded(
45             child: ListView.builder(
46                 itemCount: multiData.data!.length,
47                 itemBuilder: (context, i){
48                     return Card(
49                         child: ListTile(
50                             title: Text(multiData.data![i].name.toString()),
51                             subtitle: Text(multiData.data![i].pantoneValue.toString()),
52                         ), // ListTile
53                     ); // Card
54                 }) // ListView.builder, Expanded
55             ],
56         ), // Column
57     ); // Scaffold
58 }
```

Output



d) Without model



GET https://reqres.in/api/unknown

Params Authorization Headers (6) Body Pre-request Script Tests Settings

Query Params

Key	Value	Description
Key	Value	Description

Body Cookies Headers (17) Test Results

Status: 200 OK Time: 23ms

Pretty Raw Preview Visualize JSON

```
1  {
2      "page": 1,
3      "per_page": 6,
4      "total": 12,
5      "total_pages": 2,
6      "data": [
7          {
8              "id": 1,
9              "name": "cerulean",
10             "year": 2000,
11             "color": "#9882D1",
12             "pantone_value": "15-4020"
13         },
14         {
15             "id": 2,
16             "name": "fuchsia-rose",
17             "year": 2001
18         }
19     ]
20 }
```

api-services.dart

```
Future<dynamic> getMultiDataWithoutModel() async{
    try{
        var response = await http.get(Uri.parse("https://reqres.in/api/unknown"));
        if(response.statusCode == 200){
            final model = jsonDecode(response.body);
            return model;
        }
    }catch (e){
        print(e);
    }
    return null;
}
```

Main.dart

A screenshot of a Flutter development environment showing the `main.dart` file. The code defines the `MyApp` widget, which is the root of the application. It uses a `MaterialApp` with a blue theme and a `ScreenWithoutModel` as the initial screen.

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

  // This widget is the root of your application.
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Flutter Demo',
      theme: ThemeData(
        // This is the theme of your application.
        //
        // Try running your application with "flutter run". You'll see the
        // application has a blue toolbar. Then, without quitting the app, try
        // changing the primarySwatch below to Colors.green and then invoke
        // "hot reload" (press "r" in the console where you ran "flutter run",
        // or simply save your changes to "hot reload" in a Flutter IDE).
        // Notice that the counter didn't reset back to zero; the application
        // is not restarted.
        primarySwatch: Colors.blue,
      ), // ThemeData
      home: const ScreenWithoutModel(),
    ); // MaterialApp
  }
}
```

Using

A screenshot of a Flutter development environment showing the `ScreenWithoutModel` class implementation. It extends `StatefulWidget` and includes a `_ScreenState` state class. The `_ScreenState` class handles data retrieval from an API and updates the UI when the data is ready.

```
import 'package:multi_data_api/api_services.dart';

class ScreenWithoutModel extends StatefulWidget {
  const ScreenWithoutModel({Key? key}) : super(key: key);

  @override
  State<ScreenWithoutModel> createState() => _ScreenWithoutModelState();
}

class _ScreenWithoutModelState extends State<ScreenWithoutModel> {

  bool isReady = false;
  dynamic multiData;
  _getMultiData(){
    isReady = true;
    ApiServices().getMultiDataWithOutModel().then((value){
      setState(() {
        multiData = value;
        isReady = false;
      });
    });
  }
}

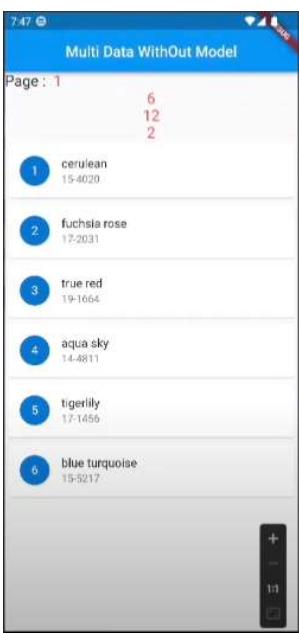
@Override
void initState() {
  _getMultiData();
  super.initState();
}
```

```

Column(
  children: [
    Row(
      children: [
        Text(multiData["page"].toString(), style: const TextStyle(color: Colors.redAccent)),
        Text(multiData["page"].toString(), style: const TextStyle(color: Colors.redAccent)),
      ],
    ), // Row
    Text(multiData["per_page"].toString(), style: const TextStyle(color: Colors.redAccent)),
    Text(multiData["total"].toString(), style: const TextStyle(color: Colors.redAccent)),
    Text(multiData["total_pages"].toString(), style: const TextStyle(color: Colors.redAccent)),
  ],
),
Expanded(
  child: ListView.builder(
    itemCount: multiData["data"].length,
    itemBuilder: (context, index){
      return Card(
        child: ListTile(
          leading: CircleAvatar(
            child: Text(multiData["data"][index]["id"].toString()),
          ), // CircleAvatar
          title: Text(multiData["data"][index]["name"].toString()),
          subtitle: Text(multiData["data"][index]["pantone_value"].toString())
        ), // ListTile
      ); // Card
    }
  )
)

```

Output



- Pass headers in api

The screenshot shows the Plati Fake Store API documentation. On the left, there's a sidebar with sections like About, REST API (Products, Filter products, Categories, Users, **Auth JWT**, Files, Swagger Docs), GraphQL (Products, Filter products, Categories, Users, Auth JWT, Playground), and Resources (Postman, Insomnia). The main content area is titled "Get user with session". It says: "You can get the profile the current user with session if in the headers include the `Authorization` key with the value `Bearer <your access token>` to `/auth/profile`". Below this is a "Request:" section with a code block: [GET] https://api.escuelajs.co/api/v1/auth/profile # Headers { "Authorization": "Bearer <your access token>" } and a "Response:" section with a JSON object: { "id": 1, "email": "john@mail.com", "password": "changeme", "name": "Jhon", "role": "customer", "avatar": "https://api.lorem.space/image/face?w=64&h=48&r=867" }. To the right, there's an "On this page" sidebar with links to Overview, Authentication, Get user with session (which is highlighted in blue), and Get a new Access Token with a Refresh Token.

The screenshot shows a Flutter IDE interface with several tabs: main.dart, home_screen.dart, api_services.dart (which is the active tab), and pubspec.yaml. The api_services.dart file contains the following code:

```
import 'package:http/http.dart' as http;

class ApiService {
  Future<dynamic> getUserDetails () async{
    String token = "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiJSImlhdCI6MTcwODE2MDg0NCw1ZXhwIjoxNzA5MDk2MDQ0fQ.QwBFJ4ip6q876zDc9R0Kj60";
    var response = await http.get(Uri.parse("https://api.escuelajs.co/api/v1/auth/profile"),
      headers: {
        "Authorization": "Bearer $token"
      });
    if(response.statusCode == 200){
      return jsonDecode(response.body);
    }else{
      return null;
    }
  }
}
```

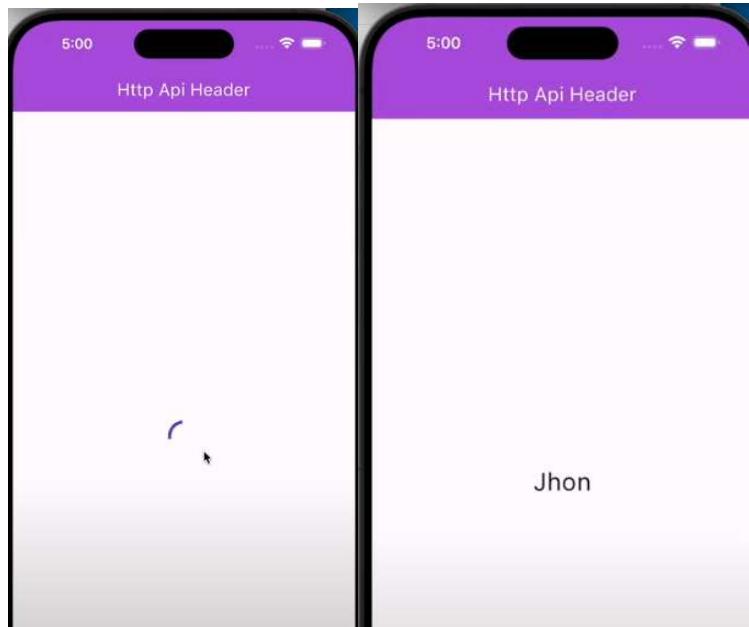
The code defines a class ApiService with a method getUserDetails that makes a GET request to the /auth/profile endpoint of the API. It includes a hardcoded access token in the headers. A yellow dot on the line "headers: {" indicates a breakpoint.

Using

```
main.dart  home_screen.dart  api_services.dart  pubspec.yaml

16
17:     getUserDetails () {
18:       setState(() {
19:         isLoading = true;
20:       });
21:       ApiServices().getUserDetails().then((value){
22:         print('userDetails ==> ${value.toString()}' );
23:         setState(() {
24:           userDetails = value;
25:           isLoading = false;
26:         });
27:       }).onError((error, stackTrace){
28:         print(error.toString());
29:       });
30:     };
31:   }
32:
33:
34:   @override
35:   void initState() {
36:     getUserDetails();
37:     super.initState();
38:   }
39:
40:   @override
41:   Widget build(BuildContext context) {
42:     return Scaffold(
43:       appBar: AppBar(
44:         centerTitle: true,
45:         backgroundColor: CupertinoColors.systemPurple,
46:         title: const Text("Http Api Header", style: TextStyle(color: Colors.white)),
47:       ), // AppBar
48:
49:       body: Center(
50:         child: isLoading == true?
51:             const CircularProgressIndicator():
52:
53:             Text(userDetails['name'].toString(), textScaleFactor: 2),
54:       ), // Center
55:     ); // Scaffold
56:   }
57:
```

Output



- Upload image

 Platzi Fake Store API

Search

About

- Introduction
- Showcase

REST API

- Products
- Filter products
- Categories
- Users
- Auth JWT
- Files**
- Swagger Docs

GraphQL

- Products
- Filter products
- Categories
- Users
- Auth JWT
- Playground

Resources

- Postman
- Insomnia

Files

Upload File

You can upload file by using the `/files/upload` endpoint but in header the `Content-Type` should be `multipart/form-data`.

Request:

```
[POST] https://api.escuelajs.co/api/v1/files/upload
# Body
{
  "file": "<Binary File>"
}
```

The response is like this:

```
{
  "originalname": "Djhv7NO - Imgur.png",
  "filename": "f3a5.png",
  "location": "https://api.escuelajs.co/api/v1/files/f3a5.png"
}
```

POST https://api.escuelajs.co/api/v1/files/upload.

Params Authorization Headers (8) **Body** Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary

Key	Value	...
<input checked="" type="checkbox"/> file	Simulator Screenshot - iPhone 15 Pro Max - 2024-02-17 at 17.08.35.png	x
Key	Value	...

Body Cookies Headers (12) Test Results

Pretty Raw Preview Visualize JSON

Status: 201 Created Time: 2.93 s Size: 961 B Save

201 Created

The request has been fulfilled and resulted in a new resource being created.

```

1: {
2:   "originalname": "Simulator Screenshot - iPhone 15 Pro Max - 2024-02-17 at 17.08.35.png",
3:   "filename": "d13a.png",
4:   "location": "https://api.escuelajs.co/api/v1/files/d13a.png"
5: }

```

image_picker 1.0.7

Published 39 days ago • [flutter.dev](#) Dart 3 compatible

SDK FLUTTER PLATFORM ANDROID IOS LINUX MACOS WEB WINDOWS

6.2K

[Readme](#) [Changelog](#) [Example](#) [Installing](#) [Versions](#) [Scores](#)

Image Picker plugin for Flutter

pub v1.0.7

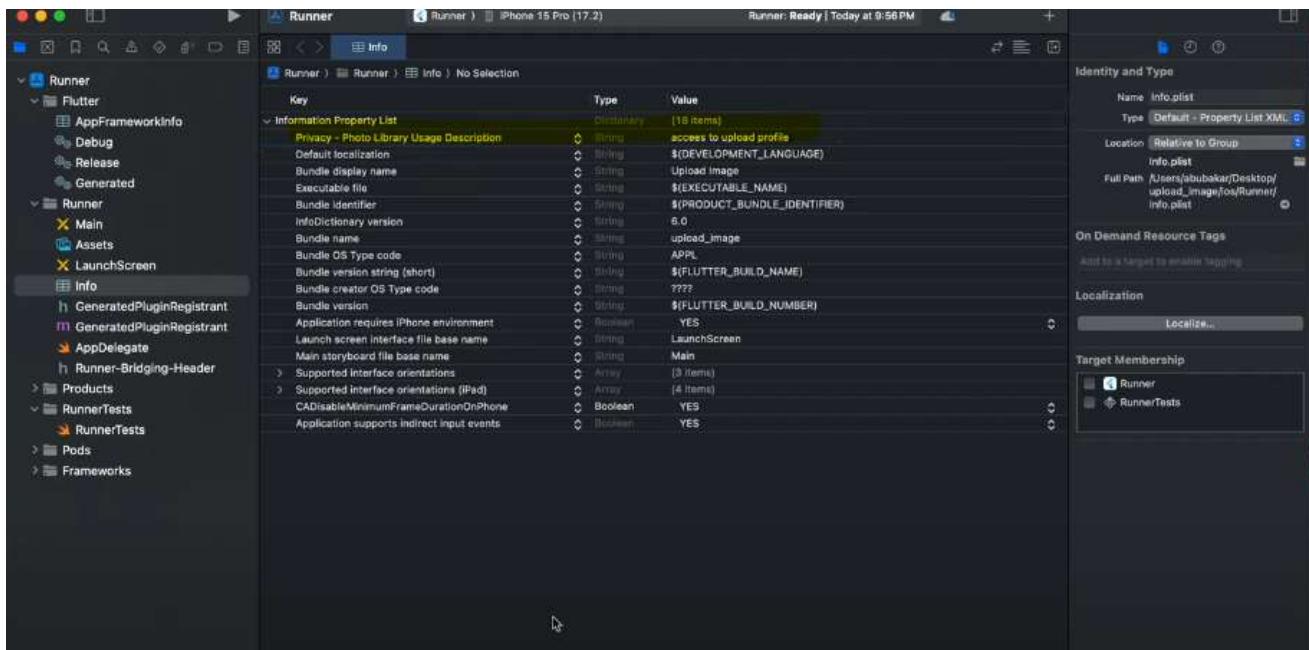
A Flutter plugin for iOS and Android for picking images from the image library, and taking new pictures with the camera.

Android iOS Linux macOS Web Windows

Support **SDK 21+** iOS 11+ Any 10.14+ See [image_picker_for_web](#) Windows 10+

Installation

First, add `image_picker` as a dependency in your `pubspec.yaml` file.



Api

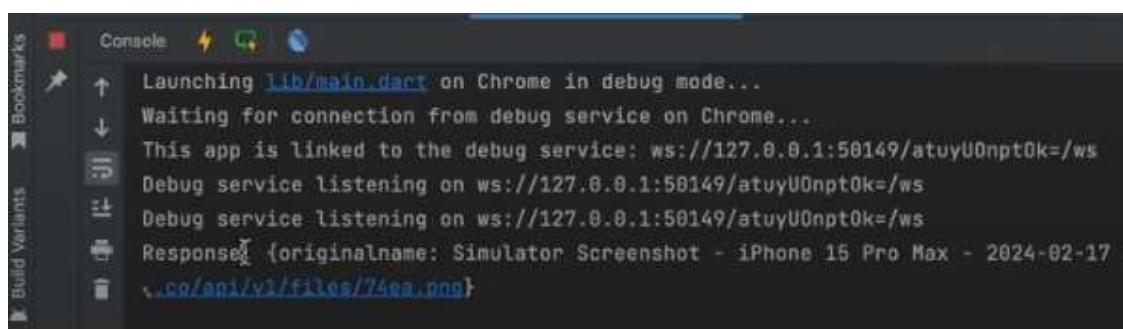
```
class ApiServices {  
  Future<dynamic> uploadImage (Uint8List bytes, String filename) async{  
    var request = http.MultipartRequest('POST', Uri.parse('https://api.escuelajs.co/api/v1/files/upload'));  
  
    var multipartFile = http.MultipartFile(  
      'file',  
      http.ByteStream.fromBytes(bytes),  
      bytes.length,  
      filename: filename  
    ); // http.MultipartFile  
  
    request.files.add(multipartFile);  
    StreamedResponse response = await request.send();  
  
    if(response.statusCode == 201){  
      var data = await response.stream.bytesToString();  
      return jsonDecode(data);  
    }else {  
      return null;  
    }  
  }  
}
```

Calling

```
class _HomeScreenState extends State<HomeScreen> {  
  @override  
  Widget build(BuildContext context) {  
    return Scaffold(  
      appBar: AppBar(  
        centerTitle: true,  
        backgroundColor: Colors.teal,  
        title: const Text('Android, iOS & Web', style: TextStyle(color: Colors.white)),  
      ), // AppBar  
  
      body: Center(  
        child: ElevatedButton(  
          onPressed: ()async{  
            final ImagePicker picker = ImagePicker();  
            final XFile? image = await picker.pickImage(source: ImageSource.gallery);  
  
            if(image != null){  
              Uint8List bytes = await image.readAsBytes();  
  
              ApiServices().uploadImage(bytes, image.name).then((value){  
                print(value.toString());  
              });  
            }  
          },  
        ),  
      ),  
    );  
  }  
}
```

```
        ApiServices().uploadImage(bytes, image.name).then((value){
            print(value.toString());
        }).onError((error, stackTrace){
            print(error.toString());
        });
    },
    style: ElevatedButton.styleFrom(
        backgroundColor: Colors.teal,
        foregroundColor: Colors.white
    ),
    child: const Text("Upload Image"), // ElevatedButton
```

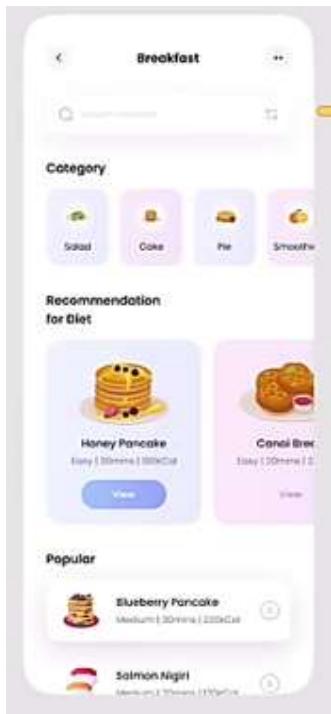
Output



The screenshot shows the Android Studio interface with the 'Console' tab selected. The output window displays the following log messages:

- Launching lib/main.dart on Chrome in debug mode...
- Waiting for connection from debug service on Chrome...
- This app is linked to the debug service: ws://127.0.0.1:50149/atuyU0nptOk=/ws
- Debug service listening on ws://127.0.0.1:50149/atuyU0nptOk=/ws
- Debug service listening on ws://127.0.0.1:50149/atuyU0nptOk=/ws
- Response {originalname: Simulator Screenshot - iPhone 15 Pro Max - 2024-02-17 11:00:00 UTC, name: /storage/emulated/0/Android/data/com.example.flutterapp/files/74ea.png}

- Designing UI



• Adding fonts in flutter

ADD FONT TO PROJECT

```

File Edit Selection View Go Run Terminal Help
pubspec.yaml - fitness - Visual Studio Code
EXPLORER main.dart home.dart pubspec.yaml
pubspec.yaml
70   # For details regarding adding assets from package dependencies,
71   # see https://flutter.dev/assets-and-images/#from-packages
72
73   # To add custom fonts to your application, add a fonts section here,
74   # in this "flutter" section. Each entry in this list should have a
75   # "family" key with the font family name, and a "fonts" key with a
76   # list giving the asset and other descriptors for the font. For
77   # example:
78
79   fonts:
80     - family: Poppins
81       fonts:
82         - asset: fonts/Poppins-Bold.ttf
83           weight: 700
84         - asset: fonts/Poppins-Medium.ttf
85           weight: 500
85         - asset: fonts/Poppins-Regular.ttf
86           weight: 400
86         - asset: fonts/Poppins-SemiBold.ttf
87           weight: 600
88
89     # - family: Trajan Pro
90     #   fonts:
91     #     - asset: fonts/TrajanPro.ttf
92     #       weight: 700
93     #     - asset: fonts/TrajanPro_Bold.ttf
93     #       weight: 700
94
95   # For details regarding fonts from package dependencies,
96   # see https://flutter.dev/custom-fonts/#from-packages
97

```

Ln 88, Col 22 Spaces: 2 UTF-8 CRLF { } YAML Pixel 3a API 30 (android-x86 emulator) ⚡

ADD FONT TO PROJECT

```

File Edit Selection View Go Run Terminal Help
main.dart - fitness - Visual Studio Code
EXPLORER main.dart home.dart
main.dart
11   // This widget is the root of your application.
12   @override
13   Widget build(BuildContext context) {
14     return MaterialApp(
15       debugShowCheckedModeBanner: false,
16       theme: ThemeData(fontFamily: 'Poppins'),
17       home: HomePage(),
18     );
19   }
20
21
22
23

```

Ln 23, Col 12 Spaces: 2 UTF-8 CRLF { } YAML Pixel 3a API 30 (android-x86 emulator) ⚡

• App Bar

IMPLEMENT APPBAR

The screenshot shows the Visual Studio Code interface with the file `home.dart` open. The code defines a `HomePage` class that extends `StatelessWidget`. In the `build` method, an `AppBar` is created with a title "Breakfast". The title's style is set to a bold black font of size 18. A yellow lightbulb icon is visible on the line where the `fontWeight: FontWeight.bold` is specified. To the right of the editor, a mobile device simulator displays the app with the title "Breakfast" at the top.

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

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

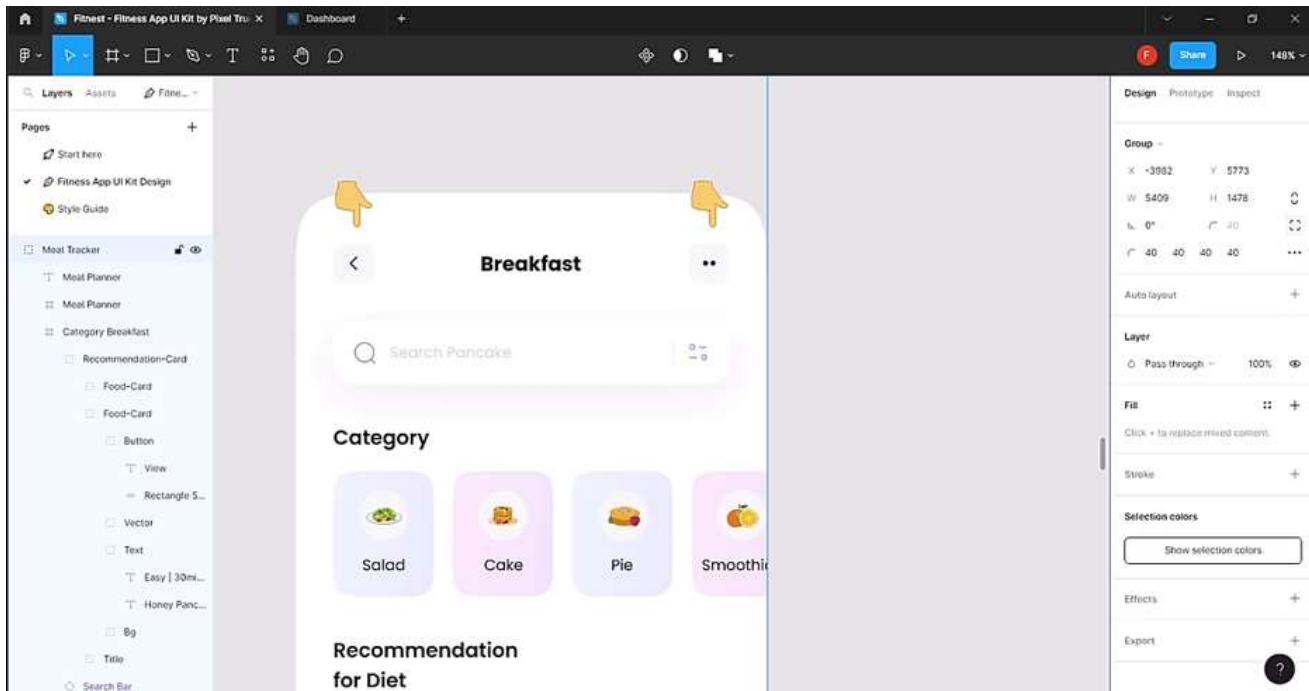
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text(
          'Breakfast',
          style: TextStyle(
            color: Colors.black,
            fontSize: 18,
            fontWeight: FontWeight.bold
          ),
        ),
        centerTitle: true,
      ),
    );
  }
}
```

IMPLEMENT APPBAR

This screenshot is similar to the one above, but it includes a line of code that adds a white background color to the `AppBar`. A yellow hand cursor icon is positioned over this line of code. The mobile device simulator on the right shows the updated app with a white header bar.

```
backgroundColor: Colors.white,
```

Adding icons in appbar



➤ flutter_svg

flutter_svg 1.1.6

Published 3 months ago • @dnfield.dev (Dart 3 ready) • Latest: 1.1.6 / Prerelease: 2.0.0-dev.2

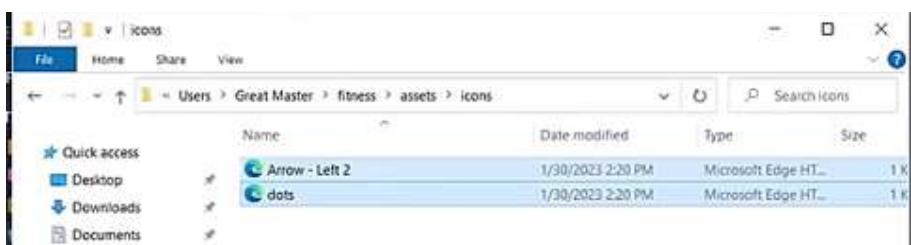
SDK FLUTTER PLATFORM ANDROID IOS LINUX MACOS WEB WINDOWS

3.5K

[Readme](#) [Changelog](#) [Example](#) [Installing](#) [Versions](#) [Scores](#)

flutter_svg

[pub v1.1.6](#) [build passing](#) [coverage](#)



Make icons folder inside assets and define path in pubspec.yaml

The screenshot shows the VS Code interface with the following details:

- EXPLORER** view: Shows the project structure with a red bracket highlighting the `assets/icons` folder.
- pubspec.yaml** file content:

```
47  # activated in the analysis_options.yaml file located
48  # package. See that file for information about deactivating specific lint
49  # rules and activating additional ones.
50  flutter_lints: ^2.0.0
51
52  # For information on the generic Dart part of this file, see the
53  # following page: https://dart.dev/tools/pub/pubspec
54
55  # The following section is specific to Flutter packages.
56  flutter:
57
58      # The following line ensures that the Material Icons font is
59      # included with your application, so that you can use the icons in
60      # the material Icons class.
61      uses-material-design: true
62
63      # To add assets to your application, add an assets section, like this:
64      assets:
65          - assets/icons/
66          # - images/a_dot_ham.jpeg
67
```

- PROBLEMS**: 3 issues.
- OUTPUT**, **DEBUG CONSOLE**, **TERMINAL** tabs at the bottom.

Icon

The screenshot shows the VS Code interface with the following details:

- home.dart** file content:

```
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      title: Text(
        'Breakfast',
        style: TextStyle(
          color: Colors.black,
          fontSize: 18,
          fontWeight: FontWeight.bold
        ), // TextStyle
      ), // Text
      backgroundColor: Colors.white,
      elevation: 0.0,
      centerTitle: true,
      Leading: Container(
        margin: EdgeInsets.all(10),
        alignment: Alignment.center,
        child: SvgPicture.asset(
          'assets/icons/Arrow - Left 2.svg',
          height: 20,
          width: 20,
        ), // SvgPicture.asset
      decoration: BoxDecoration(
        color: Color(0xffff7F88),
        borderRadius: BorderRadius.circular(10)
      ), // BoxDecoration
    ), // Container
  ), // AppBar
); // Scaffold
```

- OUTPUT** tab: Shows the output of the app running on an Android emulator with the title "Breakfast".

Leading to get leftside icon

Wrapping the icons in actions:[] to get right side icon

Wrapping in gestureDetector to get onTap function



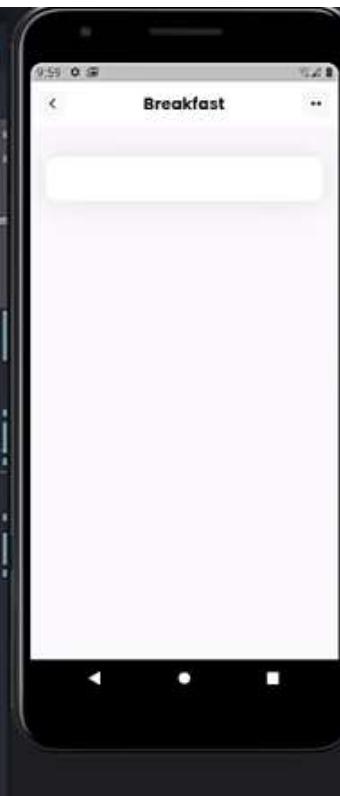
```
main.dart home.dart
  pages > home.dart > HomePage > build
11   title: Text(
12     'Breakfast',
13     style: TextStyle(
14       color: Colors.black,
15       fontSize: 18,
16       fontWeight: FontWeight.bold
17     ), // TextStyle
18   ), // Text
19   backgroundColor: Colors.white,
20   elevation: 0.0,
21   centerTitle: true,
22   Leading: GestureDetector(
23     onTap: () {
24       child: Container(
25         margin: EdgeInsets.all(10),
26         alignment: Alignment.center,
27         child: SvgPicture.asset(
28           'assets/icons/Arrow - Left 2.svg',
29           height: 20,
30           width: 20,
31         ),
32         decoration: BoxDecoration(
33           color: Color(0xFFF7FBF8),
34           borderR // BoxDecoration radius: BorderRadius.circular(10)
35         ),
36       ),
37     ), // Container
38     actions: [
39       GestureDetector(
40         onTap: () {
```

- adding input



Code snippet showing the initial state of the `build` method in `HomePage`:

```
lib> pages > home.dart > HomePage > build
4   class HomePage extends StatelessWidget {
5     const HomePage({super.key});
6
7     @override
8     Widget build(BuildContext context) {
9       return Scaffold(
10       appBar: appBar(),
11       body: Column(
12         children: [
13           Container(
14             margin: EdgeInsets.only(top: 40, left: 20, right: 20),
15             decoration: BoxDecoration(
16               boxShadow: [
17                 BoxShadow(
18                   color: Color(0xff1D1617).withOpacity(0.11),
19                   blurRadius: 40,
20                   spreadRadius: 0.0
21                 ) // BoxShadow
22               ],
23             ), // BoxDecoration
24             child: TextField(
25               decoration: InputDecoration(
26                 filled: true,
27                 fillColor: Colors.white,
28               ), // InputDecoration
29             ), // TextField
30           ), // Container
31         ],
32       ), // Column
33     ); // Scaffold
34   }
35 }
```



Code snippet showing the state of the `build` method after adding input fields:

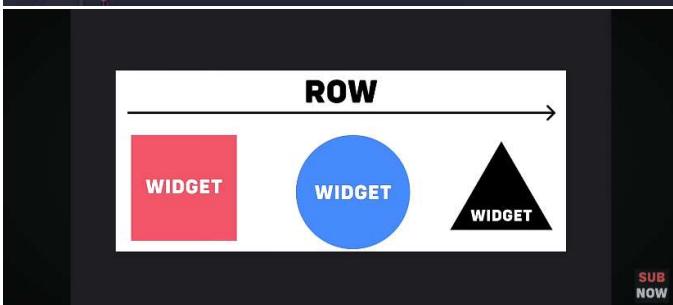
```
lib> pages > home.dart > HomePage > build
12       children: [
13         Container(
14           margin: EdgeInsets.only(top: 40, left: 20, right: 20),
15           decoration: BoxDecoration(
16             boxShadow: [
17               BoxShadow(
18                 color: Color(0xff1D1617).withOpacity(0.11),
19                 blurRadius: 40,
20                 spreadRadius: 0.0
21               ) // BoxShadow
22             ],
23           ), // BoxDecoration
24           child: TextField(
25             decoration: InputDecoration(
26               filled: true,
27               fillColor: Colors.white,
28               contentPadding: EdgeInsets.all(15),
29               border: OutlineInputBorder(
30                 borderRadius: BorderRadius.circular(15),
31                 borderSide: BorderSide.none
32               ), // OutlineInputBorder
33             ), // InputDecoration
34           ), // TextField
35         ), // Container
36       ],
37     ), // Column
38   ); // Scaffold
39 }
40 
```

The code adds a `contentPadding: EdgeInsets.all(15)` to the `InputDecoration` of the `TextField`, which creates a 15-unit padding around the text input area.

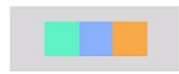
➤ Adding icons in input using prefix and suffix



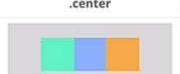
```
lib> pages > home.dart > HomePage > build
15     decoration: BoxDecoration(
16       boxShadow: [
17         BoxShadow(
18           color: Color(0xff1D1E1F).withOpacity(0.11),
19           blurRadius: 40,
20           spreadRadius: 0.8
21         ) // BoxShadow
22       ]
23     ), // BoxDecoration
24     child: TextField(
25       decoration: InputDecoration(
26         filled: true,
27         fillColor: Colors.white,
28         contentPadding: EdgeInsets.all(15),
29         prefixIcon: Padding(
30           padding: const EdgeInsets.all(12),
31           child: SvgPicture.asset('assets/icons/Search.svg'),
32         ), // Padding
33         suffixIcon: SvgPicture.asset('assets/icons/Filter.svg'),
34         border: OutlineInputBorder(
35           borderRadius: BorderRadius.circular(15),
36           borderSide: BorderSide.none
37         ), // OutlineInputBorder
38       ), // InputDecoration
39     ), // TextField
40   ), // Container
41 ],
42 ), // Column
43 ); // Scaffold
44 }
```

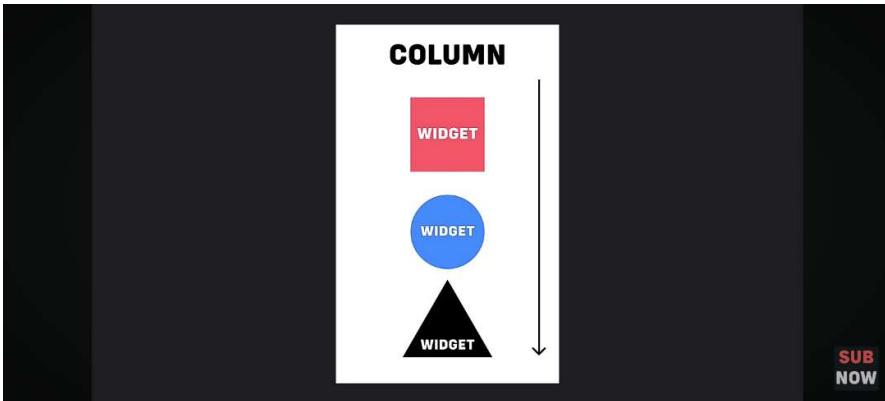


CROSS AXIS ALIGNMENT

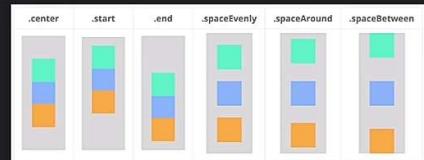
.center	.start	.end	.stretch
			

MAIN AXIS ALIGNMENT

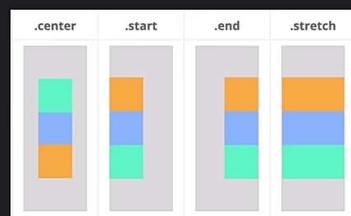
.center	.start	.end
		
.spaceEvenly	.spaceAround	.spaceBetween
		



MAIN AXIS ALIGNMENT



CROSS AXIS ALIGNMENT



SUB
NOW

➤ Adding hint text and vertical separator

```

5 > pages > home.dart > build
23     ), // BoxDecoration
24     child: TextField(
25       decoration: InputDecoration(
26         filled: true,
27         fillColor: Colors.white,
28         contentPadding: EdgeInsets.all(15),
29         hintText: 'Search Pancake',
30         hintStyle: TextStyle(
31           color: Color(0xffffffff),
32           fontSize: 14
33         ), // TextStyle
34         prefixIcon: Padding(
35           padding: const EdgeInsets.all(12),
36           child: SvgPicture.asset('assets/icons/Search.svg'),
37         ), // Padding
38         suffixIcon: Container(
39           width: 100,
40           child: IntrinsicHeight(
41             child: Row(
42               mainAxisAlignment: MainAxisAlignment.end,
43               children: [
44                 VerticalDivider(
45                   color: Colors.black,
46                   indent: 10,
47                   endIndent: 10,
48                   thickness: 0.1,
49                 ), // VerticalDivider
50                 Padding(
51                   padding: const EdgeInsets.all(8.0),
52                   child: SvgPicture.asset('assets/icons/Filter.svg'),
53                 )
54               ],
55             ),
56           ),
57         )
58       )
59     )
60   ),
61   child: Container(
62     padding: EdgeInsets.all(15),
63     child: Column(
64       children: [
65         Container(
66           padding: EdgeInsets.all(10),
67           child: Row(
68             mainAxisAlignment: MainAxisAlignment.spaceBetween,
69             children: [
70               Text('Pancake'),
71               Text('Bacon'),
72               Text('Eggs')
73             ],
74           ),
75         ),
76         Container(
77           padding: EdgeInsets.all(10),
78           child: Row(
79             mainAxisAlignment: MainAxisAlignment.spaceBetween,
80             children: [
81               Text('Pancake'),
82               Text('Bacon'),
83               Text('Eggs')
84             ],
85           ),
86         )
87       ],
88     )
89   )
90 )
91 )
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 )
117 )
118 )
119 )
120 )
121 )
122 )
123 )
124 )
125 )
126 )
127 )
128 )
129 )
130 )
131 )
132 )
133 )
134 )
135 )
136 )
137 )
138 )
139 )
140 )
141 )
142 )
143 )
144 )
145 )
146 )
147 )
148 )
149 )
150 )
151 )
152 )
153 )
154 )
155 )
156 )
157 )
158 )
159 )
160 )
161 )
162 )
163 )
164 )
165 )
166 )
167 )
168 )
169 )
170 )
171 )
172 )
173 )
174 )
175 )
176 )
177 )
178 )
179 )
180 )
181 )
182 )
183 )
184 )
185 )
186 )
187 )
188 )
189 )
190 )
191 )
192 )
193 )
194 )
195 )
196 )
197 )
198 )
199 )
200 )
201 )
202 )
203 )
204 )
205 )
206 )
207 )
208 )
209 )
210 )
211 )
212 )
213 )
214 )
215 )
216 )
217 )
218 )
219 )
220 )
221 )
222 )
223 )
224 )
225 )
226 )
227 )
228 )
229 )
230 )
231 )
232 )
233 )
234 )
235 )
236 )
237 )
238 )
239 )
240 )
241 )
242 )
243 )
244 )
245 )
246 )
247 )
248 )
249 )
250 )
251 )
252 )
253 )
254 )
255 )
256 )
257 )
258 )
259 )
259 )
260 )
261 )
262 )
263 )
264 )
265 )
266 )
267 )
268 )
269 )
270 )
271 )
272 )
273 )
274 )
275 )
276 )
277 )
278 )
279 )
279 )
280 )
281 )
282 )
283 )
284 )
285 )
286 )
287 )
288 )
289 )
289 )
290 )
291 )
292 )
293 )
294 )
295 )
296 )
297 )
298 )
299 )
299 )
300 )
301 )
302 )
303 )
304 )
305 )
306 )
307 )
308 )
309 )
309 )
310 )
311 )
312 )
313 )
314 )
315 )
316 )
317 )
318 )
319 )
319 )
320 )
321 )
322 )
323 )
324 )
325 )
326 )
327 )
328 )
329 )
329 )
330 )
331 )
332 )
333 )
334 )
335 )
336 )
337 )
338 )
339 )
339 )
340 )
341 )
342 )
343 )
344 )
345 )
346 )
347 )
348 )
349 )
349 )
350 )
351 )
352 )
353 )
354 )
355 )
356 )
357 )
358 )
359 )
359 )
360 )
361 )
362 )
363 )
364 )
365 )
366 )
367 )
368 )
369 )
369 )
370 )
371 )
372 )
373 )
374 )
375 )
376 )
377 )
378 )
379 )
379 )
380 )
381 )
382 )
383 )
384 )
385 )
386 )
387 )
388 )
389 )
389 )
390 )
391 )
392 )
393 )
394 )
395 )
396 )
397 )
398 )
399 )
399 )
400 )
401 )
402 )
403 )
404 )
405 )
406 )
407 )
408 )
409 )
409 )
410 )
411 )
412 )
413 )
414 )
415 )
416 )
417 )
418 )
419 )
419 )
420 )
421 )
422 )
423 )
424 )
425 )
426 )
427 )
428 )
429 )
429 )
430 )
431 )
432 )
433 )
434 )
435 )
436 )
437 )
438 )
439 )
439 )
440 )
441 )
442 )
443 )
444 )
445 )
446 )
447 )
448 )
449 )
449 )
450 )
451 )
452 )
453 )
454 )
455 )
456 )
457 )
458 )
459 )
459 )
460 )
461 )
462 )
463 )
464 )
465 )
466 )
467 )
468 )
469 )
469 )
470 )
471 )
472 )
473 )
474 )
475 )
476 )
477 )
478 )
479 )
479 )
480 )
481 )
482 )
483 )
484 )
485 )
486 )
487 )
488 )
489 )
489 )
490 )
491 )
492 )
493 )
494 )
495 )
496 )
497 )
498 )
499 )
499 )
500 )
501 )
502 )
503 )
504 )
505 )
506 )
507 )
508 )
509 )
509 )
510 )
511 )
512 )
513 )
514 )
515 )
516 )
517 )
518 )
519 )
519 )
520 )
521 )
522 )
523 )
524 )
525 )
526 )
527 )
528 )
529 )
529 )
530 )
531 )
532 )
533 )
534 )
535 )
536 )
537 )
538 )
539 )
539 )
540 )
541 )
542 )
543 )
544 )
545 )
546 )
547 )
548 )
549 )
549 )
550 )
551 )
552 )
553 )
554 )
555 )
556 )
557 )
558 )
559 )
559 )
560 )
561 )
562 )
563 )
564 )
565 )
566 )
567 )
568 )
569 )
569 )
570 )
571 )
572 )
573 )
574 )
575 )
576 )
577 )
578 )
579 )
579 )
580 )
581 )
582 )
583 )
584 )
585 )
586 )
587 )
588 )
589 )
589 )
590 )
591 )
592 )
593 )
594 )
595 )
596 )
597 )
598 )
599 )
599 )
600 )
601 )
602 )
603 )
604 )
605 )
606 )
607 )
608 )
609 )
609 )
610 )
611 )
612 )
613 )
614 )
615 )
616 )
617 )
618 )
619 )
619 )
620 )
621 )
622 )
623 )
624 )
625 )
626 )
627 )
628 )
629 )
629 )
630 )
631 )
632 )
633 )
634 )
635 )
636 )
637 )
638 )
639 )
639 )
640 )
641 )
642 )
643 )
644 )
645 )
646 )
647 )
648 )
649 )
649 )
650 )
651 )
652 )
653 )
654 )
655 )
656 )
657 )
658 )
659 )
659 )
660 )
661 )
662 )
663 )
664 )
665 )
666 )
667 )
668 )
669 )
669 )
670 )
671 )
672 )
673 )
674 )
675 )
676 )
677 )
678 )
679 )
679 )
680 )
681 )
682 )
683 )
684 )
685 )
686 )
687 )
688 )
689 )
689 )
690 )
691 )
692 )
693 )
694 )
695 )
696 )
697 )
698 )
699 )
699 )
700 )
701 )
702 )
703 )
704 )
705 )
706 )
707 )
708 )
709 )
709 )
710 )
711 )
712 )
713 )
714 )
715 )
716 )
717 )
718 )
719 )
719 )
720 )
721 )
722 )
723 )
724 )
725 )
726 )
727 )
728 )
729 )
729 )
730 )
731 )
732 )
733 )
734 )
735 )
736 )
737 )
738 )
739 )
739 )
740 )
741 )
742 )
743 )
744 )
745 )
746 )
747 )
748 )
749 )
749 )
750 )
751 )
752 )
753 )
754 )
755 )
756 )
757 )
758 )
759 )
759 )
760 )
761 )
762 )
763 )
764 )
765 )
766 )
767 )
768 )
769 )
769 )
770 )
771 )
772 )
773 )
774 )
775 )
776 )
777 )
778 )
779 )
779 )
780 )
781 )
782 )
783 )
784 )
785 )
786 )
787 )
788 )
789 )
789 )
790 )
791 )
792 )
793 )
794 )
795 )
796 )
797 )
798 )
799 )
799 )
800 )
801 )
802 )
803 )
804 )
805 )
806 )
807 )
808 )
809 )
809 )
810 )
811 )
812 )
813 )
814 )
815 )
816 )
817 )
818 )
819 )
819 )
820 )
821 )
822 )
823 )
824 )
825 )
826 )
827 )
828 )
829 )
829 )
830 )
831 )
832 )
833 )
834 )
835 )
836 )
837 )
838 )
839 )
839 )
840 )
841 )
842 )
843 )
844 )
845 )
846 )
847 )
848 )
849 )
849 )
850 )
851 )
852 )
853 )
854 )
855 )
856 )
857 )
858 )
859 )
859 )
860 )
861 )
862 )
863 )
864 )
865 )
866 )
867 )
868 )
869 )
869 )
870 )
871 )
872 )
873 )
874 )
875 )
876 )
877 )
878 )
879 )
879 )
880 )
881 )
882 )
883 )
884 )
885 )
886 )
887 )
888 )
889 )
889 )
890 )
891 )
892 )
893 )
894 )
895 )
896 )
897 )
898 )
898 )
899 )
900 )
901 )
902 )
903 )
904 )
905 )
906 )
907 )
908 )
909 )
909 )
910 )
911 )
912 )
913 )
914 )
915 )
916 )
917 )
918 )
919 )
919 )
920 )
921 )
922 )
923 )
924 )
925 )
926 )
927 )
928 )
929 )
929 )
930 )
931 )
932 )
933 )
934 )
935 )
936 )
937 )
938 )
939 )
939 )
940 )
941 )
942 )
943 )
944 )
945 )
946 )
947 )
948 )
949 )
949 )
950 )
951 )
952 )
953 )
954 )
955 )
956 )
957 )
958 )
959 )
959 )
960 )
961 )
962 )
963 )
964 )
965 )
966 )
967 )
968 )
969 )
969 )
970 )
971 )
972 )
973 )
974 )
975 )
976 )
977 )
978 )
979 )
979 )
980 )
981 )
982 )
983 )
984 )
985 )
986 )
987 )
988 )
989 )
989 )
990 )
991 )
992 )
993 )
994 )
995 )
996 )
997 )
998 )
999 )
999 )
1000 )
1001 )
1002 )
1003 )
1004 )
1005 )
1006 )
1007 )
1008 )
1009 )
1009 )
1010 )
1011 )
1012 )
1013 )
1014 )
1015 )
1016 )
1017 )
1018 )
1019 )
1019 )
1020 )
1021 )
1022 )
1023 )
1024 )
1025 )
1026 )
1027 )
1028 )
1029 )
1029 )
1030 )
1031 )
1032 )
1033 )
1034 )
1035 )
1036 )
1037 )
1038 )
1039 )
1039 )
1040 )
1041 )
1042 )
1043 )
1044 )
1045 )
1046 )
1047 )
1048 )
1049 )
1049 )
1050 )
1051 )
1052 )
1053 )
1054 )
1055 )
1056 )
1057 )
1058 )
1059 )
1059 )
1060 )
1061 )
1062 )
1063 )
1064 )
1065 )
1066 )
1067 )
1068 )
1069 )
1069 )
1070 )
1071 )
1072 )
1073 )
1074 )
1075 )
1076 )
1077 )
1078 )
1079 )
1079 )
1080 )
1081 )
1082 )
1083 )
1084 )
1085 )
1086 )
1087 )
1088 )
1089 )
1089 )
1090 )
1091 )
1092 )
1093 )
1094 )
1095 )
1096 )
1097 )
1098 )
1098 )
1099 )
1099 )
1100 )
1101 )
1102 )
1103 )
1104 )
1105 )
1106 )
1107 )
1108 )
1109 )
1109 )
1110 )
1111 )
1112 )
1113 )
1114 )
1115 )
1116 )
1117 )
1118 )
1119 )
1119 )
1120 )
1121 )
1122 )
1123 )
1124 )
1125 )
1126 )
1127 )
1128 )
1129 )
1129 )
1130 )
1131 )
1132 )
1133 )
1134 )
1135 )
1136 )
1137 )
1138 )
1139 )
1139 )
1140 )
1141 )
1142 )
1143 )
1144 )
1145 )
1146 )
1147 )
1148 )
1149 )
1149 )
1150 )
1151 )
1152 )
1153 )
1154 )
1155 )
1156 )
1157 )
1158 )
1159 )
1159 )
1160 )
1161 )
1162 )
1163 )
1164 )
1165 )
1166 )
1167 )
1168 )
1169 )
1169 )
1170 )
1171 )
1172 )
1173 )
1174 )
1175 )
1176 )
1177 )
1178 )
1179 )
1179 )
1180 )
1181 )
1182 )
1183 )
1184 )
1185 )
1186 )
1187 )
1188 )
1189 )
1189 )
1190 )
1191 )
1192 )
1193 )
1194 )
1195 )
1196 )
1197 )
1198 )
1198 )
1199 )
1199 )
1200 )
1201 )
1202 )
1203 )
1204 )
1205 )
1206 )
1207 )
1208 )
1209 )
1209 )
1210 )
1211 )
1212 )
1213 )
1214 )
1215 )
1216 )
1217 )
1218 )
1219 )
1219 )
1220 )
1221 )
1222 )
1223 )
1224 )
1225 )
1226 )
1227 )
1228 )
1229 )
1229 )
1230 )
1231 )
1232 )
1233 )
1234 )
1235 )
1236 )
1237 )
1238 )
1239 )
1239 )
1240 )
1241 )
1242 )
1243 )
1244 )
1245 )
1246 )
1247 )
1248 )
1249 )
1249 )
1250 )
1251 )
1252 )
1253 )
1254 )
1255 )
1256 )
1257 )
1258 )
1259 )
1259 )
1260 )
1261 )
1262 )
1263 )
1264 )
1265 )
1266 )
1267 )
1268 )
1269 )
1269 )
1270 )
1271 )
1272 )
1273 )
1274 )
1275 )
1276 )
1277 )
1278 )
1279 )
1279 )
1280 )
1281 )
1282 )
1283 )
1284 )
1285 )
1286 )
1287 )
1288 )
1289 )
1289 )
1290 )
1291 )
1292 )
1293 )
1294 )
1295 )
1296 )
1297 )
1298 )
1298 )
1299 )
1299 )
1300 )
1301 )
1302 )
1303 )
1304 )
1305 )
1306 )
1307 )
1308 )
1309 )
1309 )
1310 )
1311 )
1312 )
1313 )
1314 )
1315 )
1316 )
1317 )
1318 )
1319 )
1319 )
1320 )
1321 )
1322 )
1323 )
1324 )
1325 )
1326 )
1327 )
1328 )
1329 )
1329 )
1330 )
1331 )
1332 )
1333 )
1334 )
1335 )
1336 )
1337 )
1338 )
1339 )
1339 )
1340 )
1341 )
1342 )
1343 )
1344 )
1345 )
1346 )
1347 )
1348 )
1349 )
1349 )
1350 )
1351 )
1352 )
1353 )
1354 )
1355 )
1356 )
1357 )
1358 )
1359 )
1359 )
1360 )
1361 )
1362 )
1363 )
1364 )
1365 )
1366 )
1367 )
1368 )
1369 )
1369 )
1370 )
1371 )
1372 )
1373 )
1374 )
1375 )
1376 )
1377 )
1378 )
1379 )
1379 )
1380 )
1381 )
1382 )
1383 )
1384 )
1385 )
1386 )
1387 )
1388 )
1389 )
1389 )
1390 )
1391 )
1392 )
1393 )
1394 )
1395 )
1396 )
1397 )
1398 )
1398 )
1399 )
1399 )
1400 )
1401 )
1402 )
1403 )
1404 )
1405 )
1406 )
1407 )
1408 )
1409 )
1409 )
1410 )
1411 )
1412 )
1413 )
1414 )
1415 )
1416 )
1417 )
1418 )
1419 )
1419 )
1420 )
1421 )
1422 )
1423 )
1424 )
1425 )
1426 )
1427 )
1428 )
1429 )
1429 )
1430 )
1431 )
1432 )
1433 )
1434 )
1435 )
1436 )
1437 )
1438 )
1439 )
1439 )
1440 )
1441 )
1442 )
1443 )
1444 )
1445 )
1446 )
1447 )
1448 )
1449 )
1449 )
1450 )
1451 )
1452 )
1453 )
1454 )
1455 )
1456 )
1457 )
1458 )
1459 )
1459 )
1460 )
1461 )
1462 )
1463 )
1464 )
1465 )
1466 )
1467 )
1468 )
1469 )
1469 )
1470 )
1471 )
1472 )
1473 )
1474 )
1475 )
1476 )
1477 )
1478 )
1479 )
1479 )
1480 )
1481 )
1482 )
1483 )
1484 )
1485 )
1486 )
1487 )
1488 )
1489 )
1489 )
1490 )
1491 )
1492 )
1493 )
1494 )
1495 )
1496 )
1497 )
1498 )
1498 )
1499 )
1499 )
1500 )
1501 )
1502 )
1503 )
1504 )
1505 )
1506 )
1507 )
1508 )
1509 )
1509 )
1510 )
1511 )
1512 )
1513 )
1514 )
1515 )
1516 )
1517 )
1518 )
1519 )
1519 )
1520 )
1521 )
1522 )
1523 )
1524 )
1525 )
1526 )
1527 )
1528 )
1529 )
1529 )
1530 )
1531 )
1532 )
1533 )
1534 )
1535 )
1536 )
1537 )
1538 )
1539 )
1539 )
1540 )
1541 )
1542 )
1543 )
1544 )
1545 )
1546 )
1547 )
1548 )
1549 )
1549 )
1550 )
1551 )
1552 )
1553 )
1554 )
1555 )
1556 )
1557 )
1558 )
1559 )
1559 )
1560 )
1561 )
1562 )
1563 )
1564 )
1565 )
1566 )
1567 )
1568 )
1569 )
1569 )
1570 )
1571 )
1572 )
1573 )
1574 )
1575 )
1576 )
1577 )
1578 )
1579 )
1579 )
1580 )
1581 )
1582 )
1583 )
1584 )
1585 )
1586 )
1587 )
1588 )
1589 )
1589 )
1590 )
1591 )
1592 )
1593 )
1594 )
1595 )
1596 )
1597 )
1598 )
1598 )
1599 )
1599 )
1600 )
1601 )
1602 )
1603 )
1604 )
1605 )
1606 )
1607 )
1608 )
1609 )
1609 )
1610 )
1611 )
1612 )
1613 )
1614 )
1615 )
1616 )
1617 )
1618 )
1619 )
1619 )
1620 )
1621 )
1622 )
1623 )
1624 )
1625 )
1626 )
1627 )
1628 )
1629 )
1629 )
1630 )
1631 )
1632 )
1633 )
1634 )
1635 )
1636 )
1637 )
1638 )
1639 )
1639 )
1640 )
1641 )
1642 )
1643 )
1644 )
1645 )
1646 )
1647 )
1648 )
1649 )
1649 )
1650 )
1651 )
1652 )
1653 )
1654 )
1655 )
1656 )
1657 )
1658 )
1659 )
1659 )
1660 )
1661 )
1662 )
1663 )
1664 )
1665 )
1666 )
1667 )
1668 )
1669 )
1669 )
1670 )
1671 )
1672 )
1673 )
1674 )
1675 )
1676 )
1677 )
1678 )
1679 )
1679 )
1680 )
1681 )
1682 )
1683 )
1684 )
1685 )
1686 )
1687 )
1688 )
1689 )
1689 )
1690 )
1691 )
1692 )
1693 )
1694 )
1695 )
1696 )
1697 )
1698 )
1698 )
1699 )
1699 )
1700 )
1701 )
1702 )
1703 )
1704 )
1705 )
1706 )
1707 )
1708 )
1709 )
1709 )
1710 )
1711 )
1712 )
1713 )
1714 )
1715 )
1716 )
1717 )
1718 )
1719 )
1719 )
1720 )
1721 )
1722 )
1723 )
1724 )
1725 )
1726 )
1727 )
1728 )
1729 )
1729 )
1730 )
1731 )
1732 )
1733 )
1734 )
1735 )
1736 )
1737 )
1738 )
1739 )
1739 )
1740 )
1741 )
1742 )
1743 )
1744 )
1745 )
1746 )
1747 )
1748 )
1749 )
1749 )
1750 )
1751 )
1752 )
1753 )
1754 )
1755 )
1756 )
1757 )
1758 )
1759 )
1759 )
1760 )
1761 )
1762 )
1763 )
1764 )
1765 )
1766 )
1767 )
1768 )
1769 )
1769 )
1770 )
1771 )
1772 )
1773 )
1774 )
1775 )
1776 )
1777 )
1778 )
1779 )
1779 )
1780 )
1781 )
1782 )
1783 )
1784 )
1785 )
1786 )
1787 )
1788 )
1789 )
1789 )
1790 )
1791 )
1792 )
1793 )
1794 )
1795 )
1796 )
1797 )
1798 )
1798 )
1799 )
1799 )
1800 )
1801 )
1802 )
1803 )
1804 )
1805 )
1806 )
1807 )
1808 )
1809 )
1809 )
1810 )
1811 )
1812 )
1813 )
1814 )
1815 )
1816 )
1817 )
1818 )
1819 )
1819 )
1820 )
1821 )
1822 )
1823 )
1824 )
1825 )
1826 )
1827 )
1828 )
1829 )
1829 )
1830 )
1831 )
1832 )
1833 )
1834 )
1835 )
1836 )
1837 )
1838 )
1839 )
1839 )
1840 )
1841 )
1842 )
1843 )
1844 )
1845 )
1846 )
1847 )
1848 )
1849 )
1849 )
1850 )
1851 )
1852 )
1853 )
1854 )
1855 )
1856 )
1857 )
1858 )
1859 )
1859 )
1860 )
1861 )
1862 )
1863 )
1864 )
1865 )
1866 )
1867 )
1868 )
1869 )
1869 )
1870 )
1871 )
1872 )
1873 )
1874 )
1875 )
1876 )
1877 )
1878 )
1879 )
1879 )
1880 )
1881 )
1882 )
1883 )
1884 )
1885 )
1886 )
1887 )
1888 )
1889 )
1889 )
1890 )
1891 )
1892 )
1893 )
1894 )
1895 )
1896 )
1897 )
1898 )
1898 )
1899 )
1899 )
1900 )
1901 )
1902 )
1903 )
1904 )
1905 )
1906 )
1907 )
1908 )
1909 )
1909 )
1910 )
1911 )
1912 )
1913 )
1914 )
1915 )
1916 )
1917 )
1918 )
1919 )
1919 )
1920 )
1921 )
1922 )
1923 )
1924 )
1925 )
1926 )
1927 )
1928 )
1929 )
1929 )
1930 )
1931 )
1932 )
1933 )
1934 )
1935 )
1936 )
1937 )
1938 )
1939 )
1939 )
1940 )
1941 )
1942 )
1943 )
1944 )
1945 )
1946 )
1947 )
1948 )
1949 )
1949 )
1950 )
1951 )
1952 )
1953 )
1954 )
1955 )
1956 )
1957 )
1958 )
1959 )
1959 )
1960 )
1961 )
1962 )
1963 )
1964 )
1965 )
1966 )
1967 )
1968 )
1969 )
1969 )
1970 )
1971 )
1972 )
1973 )
1974 )
1975 )
1976 )
1977 )
1978 )
1979 )
1979 )
1980 )
1981 )
1982 )
1983 )
1984 )
1985 )
1986 )
1987 )
1988 )
1989 )
1989 )
1990 )
1991 )
1992 )
1993 )
1994 )
1995 )
1996 )
1997 )
1998 )
1998 )
1999 )
1999 )
2000 )
2001 )
2002 )
2003 )
2004 )
2005 )
2006 )
2007 )
2008 )
2009 )
2009 )
2010 )
2011 )
2012 )
2013 )
2014 )
2015 )
2016 )
2017 )
2018 )
2019 )
2019 )
2020 )
2021 )
2022 )
2023 )
2024 )
2025 )
2026 )
2027 )
2028 )
2029 )
2029 )
2030 )
2031 )
2032 )
2033 )
2034 )
2035 )
2036 )
2037 )
2038 )
2039 )
2039 )
2040 )
2041 )
2042 )
2043 )
2044 )
2045 )
2046 )
2047 )
2048 )
2049 )
2049 )
2050 )
2051 )
2052 )
2053 )
2054 )
2055 )
2056 )
2057 )
2058 )
2059 )
2059 )
2060 )
2061 )
2062 )
2063 )
2064 )
2065 )
2066 )
2067 )
2068 )
2069 )
2069 )
2070 )
2071 )
2072 )
2073 )
2074 )
2075 )
2076 )
2077 )
2078 )
2079 )
2079 )
2080 )
2081 )
2082 )
2083 )
2084 )
2085 )
2086 )
2087 )
2088 )
2089 )
2089 )
2090 )
2091 )
2092 )
2093 )
2094 )
2095 )
2096 )
2097 )
2098 )
2098 )
2099 )
2099 )
2100 )
2101 )
2102 )
2103 )
2104 )
2105 )
2106 )
2107 )
2108 )
2109 )
2109 )
2110 )
2111 )
2112 )
2113 )
2114 )
2115 )
2116 )
2117 )
2118 )
2119 )
2119 )
2120 )
2121 )
2122 )
2123 )
2124 )
2125 )
2126 )
2127 )
2128 )
2129 )
2129 )
2130 )
2131 )
2132 )
2133 )
2134 )
2135 )
2136 )
2137 )
2138 )
2139 )
2139 )
2140 )
2141 )
2142 )
2143 )
2144 )
2145 )
2146 )
2147 )
2148 )
2149 )
2149 )
2150 )
2151 )
2152 )
2153 )
2154 )
2155 )
2156 )
2157 )
2158 )
2159 )
2159 )
2160 )
2161 )
2162 )
2163 )
2164 )
2165 )
2166 )
2167 )
2168 )
2169 )
2169 )
2
```

- Making Container _searchField() {} to simplity things using the datatype it will return
- Adding Category text

The screenshot shows the Dart code for a Flutter application. The code defines a `Widget build(BuildContext context)` method that returns a `Scaffold`. Inside the `Scaffold`, there is an `AppBar` and a `body` which is a `Column`. The `Column` has two children: a `Container _searchField()` and a `Text` widget. The `Text` widget displays the word "Category". To the right of the code editor, a mobile device screen displays the app's interface. The device shows a search bar at the top with the text "Breakfast". Below the search bar is a white search results area with the word "Category" displayed.

```

lib> pages > home.dart > HomePage > build
8   Widget build(BuildContext context) {
9     return Scaffold(
10       appBar: appBar(),
11       backgroundColor: Colors.white,
12       body: Column(
13         children: [
14           _searchField(),
15           SizedBox(height: 48),
16           Column(
17             children: [
18               Text(
19                 'Category',
20                 style: TextStyle(
21                   color: Colors.black,
22                   fontSize: 18,
23                   fontWeight: FontWeight.w600
24                 ), // TextStyle
25               ) // Text
26             ],
27           ) // Column
28         ],
29       ); // Column
30     ); // Scaffold
31   }
32
33   Container _searchField() {
34     return Container(
35       margin: EdgeInsets.only(top: 48, left: 20, right: 20),
36       decoration: BoxDecoration(
37         boxShadow: [

```

crossAxisAlignment start

The screenshot shows the same Dart code as the previous one, but with a modification. The `Column` now has a `crossAxisAlignment: CrossAxisAlignment.start` property. A red checkmark icon is placed next to the `start` value to indicate it is correct. The rest of the code and the resulting app interface (displaying "Category") remain the same.

```

lib> pages > home.dart > HomePage > build
8   Widget build(BuildContext context) {
9     return Scaffold(
10       appBar: appBar(),
11       backgroundColor: Colors.white,
12       body: Column(
13         crossAxisAlignment: CrossAxisAlignment.start, ✓
14         children: [
15           _searchField(),
16           SizedBox(height: 48),
17           Column(
18             children: [
19               Text(
20                 'Category',
21                 style: TextStyle(
22                   color: Colors.black,
23                   fontSize: 18,
24                   fontWeight: FontWeight.w600
25                 ), // TextStyle
26               ) // Text
27             ],
28           ) // Column
29         ],
30       ); // Column
31     ); // Scaffold
32   }
33
34   Container _searchField() {
35     return Container(
36       margin: EdgeInsets.only(top: 48, left: 20, right: 20),
37       decoration: BoxDecoration(

```

Giving small padding from left

```
8 Widget build(BuildContext context) {
9     return Scaffold(
10         appBar: appBar(),
11         backgroundColor: Colors.white,
12         body: Column(
13             mainAxisAlignment: MainAxisAlignment.start,
14             children: [
15                 _searchField(),
16                 SizedBox(height: 40),
17                 Column(
18                     children: [
19                         Padding(
20                             padding: const EdgeInsets.only(left: 20),
21                             child: Text(
22                                 'Category',
23                                 style: TextStyle(
24                                     color: Colors.black,
25                                     fontSize: 18,
26                                     fontWeight: FontWeight.w600
27                                 ), // TextStyle
28                             ), // Text
29                         ) // Padding
30                         ],
31                     ) // Column
32                 ],
33             ), // Column
34         ); // Scaffold
35     }
36
37     Container _searchField() {
```

Adding box for horizontal list



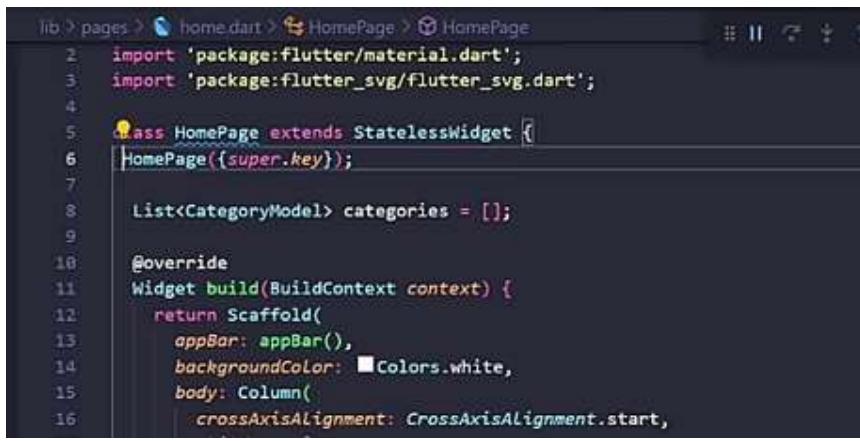
```
main.dart lib > pages > home.dart > HomePage > build
18     crossAxisAlignment: CrossAxisAlignment.start,
19     children: [
20       Padding(
21         padding: const EdgeInsets.only(left: 20),
22         child: Text(
23           'Category',
24           style: TextStyle(
25             color: Colors.black,
26             fontSize: 18,
27             fontWeight: FontWeight.w600
28           ), // TextStyle
29         ), // Text
30       ), // Padding
31       SizedBox(height: 15),
32       Container(
33         height: 150,
34         color: Colors.green,
35         child: ListView.builder(
36           itemBuilder: (context, index) {
37             if (index == 0) {
38               return Text('Salad');
39             } else if (index == 1) {
40               return Text('Cake');
41             }
42             return Container();
43           },
44         ), // ListView.builder
45       ), // Container
46     ), // Column
```

➤ Category model



```
2
3 class CategoryModel {
4   String name;
5   String iconPath;
6   Color boxColor;
7
8   CategoryModel({
9     required this.name,
10    required this.iconPath,
11    required this.boxColor,
12  );
13
14 static List<CategoryModel> getCategories() {
15   List<CategoryModel> categories = [];
16
17   categories.add(
18     CategoryModel(
19       name: 'Salad',
20       iconPath: 'assets/icons/plate.svg',
21       boxColor: Color(0xff92A3FD)
22     ) // CategoryModel
23   );
24
25   categories.add(
26     CategoryModel(
27       name: 'Cake',
28       iconPath: 'assets/icons/pancakes.svg',
29       boxColor: Color(0xffC58BF2)
30     ) // CategoryModel
31   );
32
33   categories.add(
34     CategoryModel(
35       name: 'Pie',
36       iconPath: 'assets/icons/pie.svg',
37       boxColor: Color(0xff92A3FD)
38     ) // CategoryModel
39   );
40 }
```

➤ Using category model



```
lib> pages > home.dart > HomePage > HomePage
1
2 import 'package:flutter/material.dart';
3 import 'package:flutter_svg/flutter_svg.dart';
4
5 class HomePage extends StatelessWidget {
6   HomePage({super.key});
7
8   List<CategoryModel> categories = [];
9
10  @override
11  Widget build(BuildContext context) {
12    return Scaffold(
13      appBar: appBar(),
14      backgroundColor: Colors.white,
15      body: Column(
16        crossAxisAlignment: CrossAxisAlignment.start,
```

➤ Home.dart

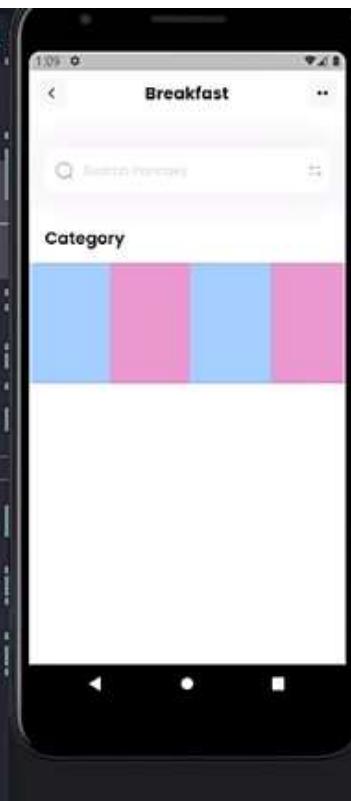
```
lib> pages > home.dart > HomePage > build
+
5   class HomePage extends StatelessWidget {
6     HomePage({super.key});
7
8     List<CategoryModel> categories = [];
9
10    void _getCategories() {
11      categories = CategoryModel.getCategories();
12    }
13
14    @override
15    Widget build(BuildContext context) {
16      _getCategories();
17      return Scaffold(
18        appBar: appBar(),
19        backgroundColor: Colors.white,
20        body: Column(
21          mainAxisAlignment: MainAxisAlignment.start,
22          children: [
23            _searchField(),
24            SizedBox(height: 40),
25            Column(
```

Displaying categories

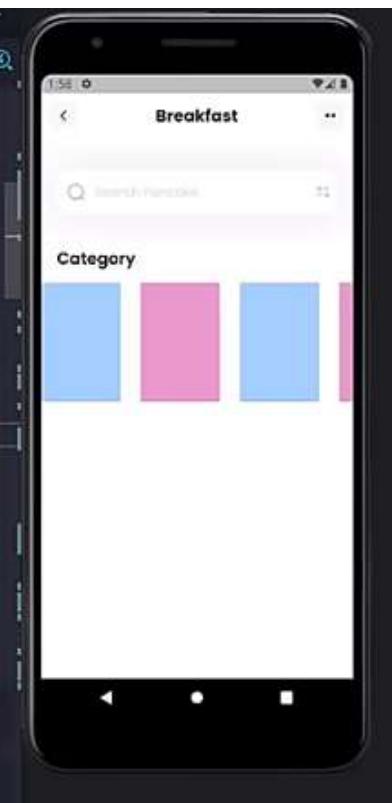
```
lib> pages > home.dart > HomePage > build
+
31           'Category',
32           style: TextStyle(
33             color: Colors.black,
34             fontSize: 18,
35             fontWeight: FontWeight.w600
36           ), // TextStyle
37           ), // Text
38           ), // Padding
39           SizedBox(height: 15),
40           Container(
41             height: 150,
42             child: ListView.builder(
43               itemCount: categories.length,
44               scrollDirection: Axis.horizontal,
45               itemBuilder: (context, index) {
46                 return Container(
47                   width: 100,
48                   decoration: BoxDecoration(
49                     color: categories[index].boxColor
50                   ), // BoxDecoration
51                 ); // Container
52               },
53             ), // ListView.builder
54           ) // Container
55         ],
56       ), // Column
57       ],
58     ), // Column
59   ); // Scaffold
```

Syncing files to device sdk gphone x86...

Hot Reload complete!

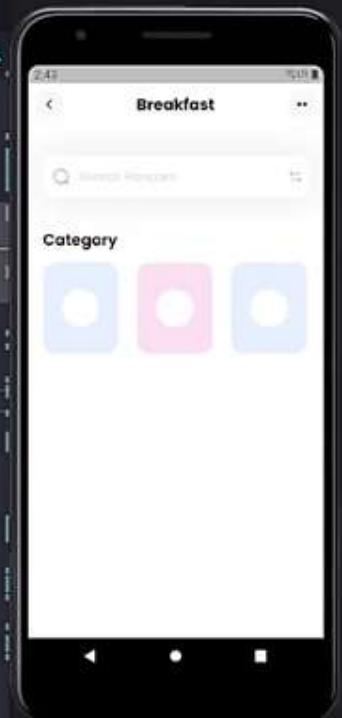


➤ Listview.separated



```
main.dart    home.dart X category_model.dart
lib > pages > home.dart > HomePage > build
      'Category',
      style: TextStyle(
        color: Colors.black,
        fontSize: 18,
        fontWeight: FontWeight.w600
      ), // TextStyle
    ), // Text
  ), // Padding
SizedBox(height: 15),
Container(
  height: 150,
  child: ListView.separated(
    itemCount: categories.length,
    scrollDirection: Axis.horizontal,
    separatorBuilder: (context, index) => SizedBox(width: 25),
    itemBuilder: (context, index) {
      return Container(
        width: 100,
        decoration: BoxDecoration(
          color: categories[index].boxColor
        ), // BoxDecoration
      ); // Container
    },
  ), // ListView.separated
) // Container
],
), // Column
),
), // Column
),
), // Scaffold
```

Adding circle and center the circle



```
main.dart    home.dart X category_model.dart
lib > pages > home.dart > build
      scrollDirection: Axis.horizontal,
      padding: EdgeInsets.only(
        left: 20,
        right: 20
      ), // EdgeInsets.only
      separatorBuilder: (context, index) => SizedBox(width: 25),
      itemBuilder: (context, index) {
        return Container(
          width: 100,
          decoration: BoxDecoration(
            color: categories[index].boxColor.withOpacity(0.3),
            borderRadius: BorderRadius.circular(16)
          ), // BoxDecoration
          child: Column(
            mainAxisAlignment: MainAxisAlignment.center,
            children: [
              Container(
                width: 50,
                height: 50,
                decoration: BoxDecoration(
                  color: Colors.white,
                  shape: BoxShape.circle
                ), // BoxDecoration
              ), // Container
            ],
          ), // Column
        ); // Container
      }, // ListView.separated
)
```

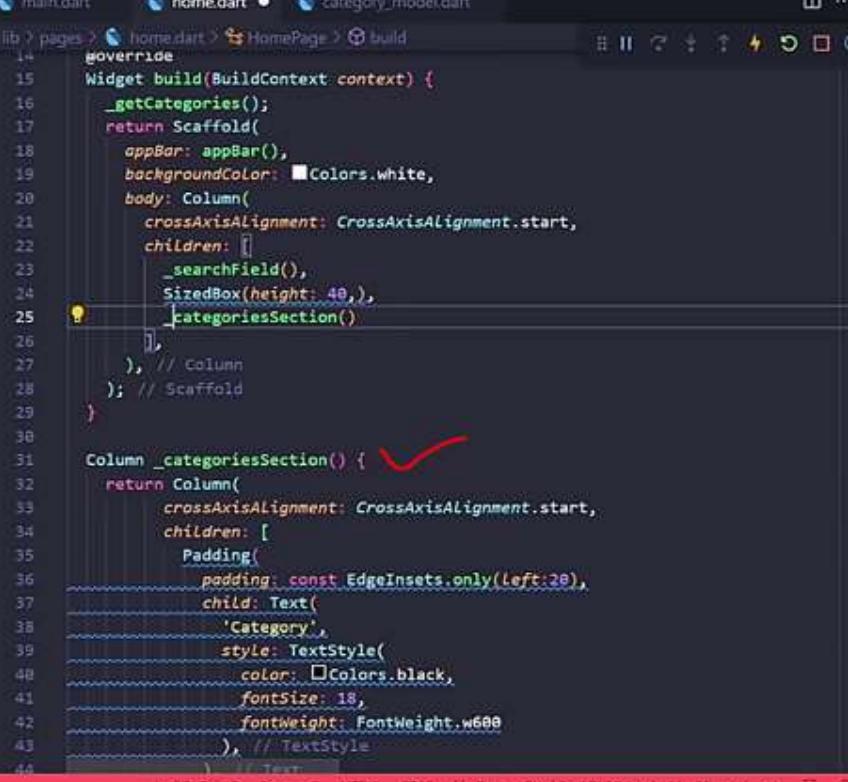
Adding icon



The screenshot shows a Flutter application running on an iPhone X simulator. The title bar says "Breakfast". Below it is a search bar with placeholder text "Search...". A "Category" section follows, containing three items: "Salad" with a green salad icon, "Cake" with a pink cake icon, and "Pie" with a blue pie icon.

```
main.dart    home.dart    category_model.dart
lib > pages > home.dart > HomePage > build
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
return Container(
    width: 100,
    decoration: BoxDecoration(
        color: categories[index].boxColor.withOpacity(0.3),
        borderRadius: BorderRadius.circular(16)
    ), // BoxDecoration
    child: Column(
        mainAxisAlignment: MainAxisAlignment.center,
        children: [
            Container(
                width: 56,
                height: 56,
                decoration: BoxDecoration(
                    color: Colors.white,
                    shape: BoxShape.circle
                ), // BoxDecoration
                child: Padding(
                    padding: const EdgeInsets.all(8.0),
                    child: SvgPicture.asset(categories[index].iconPath),
                ), // Padding
            ), // Container
            Text(
                categories[index].name,
                style: TextStyle(
                    fontWeight: FontWeight.w400,
                    color: Colors.black,
                    fontSize: 14
                ), // TextStyle
            ), // Text
        ],
    ),
),
```

Simplifying by making function for column



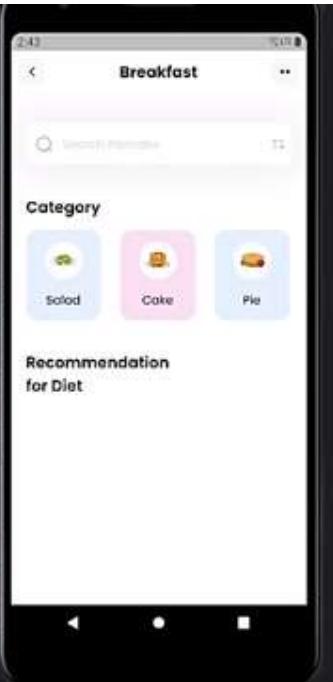
The screenshot shows the code editor with the refactored code. A red checkmark is placed over the `_categoriesSection()` method definition. The code uses a `Column` widget to group the search field and the category section.

```
main.dart    home.dart    category_model.dart
lib > pages > home.dart > HomePage > build
14
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
override
Widget build(BuildContext context) {
    _getCategories();
    return Scaffold(
        appBar: appBar(),
        backgroundColor: Colors.white,
        body: Column(
            crossAxisAlignment: CrossAxisAlignment.start,
            children: [
                searchField(),
                SizedBox(height: 40),
                _categoriesSection()
            ],
        ),
    );
}

Column _categoriesSection() { ✓
    return Column(
        crossAxisAlignment: CrossAxisAlignment.start,
        children: [
            Padding(
                padding: const EdgeInsets.only(left:20),
                child: Text(
                    'Category',
                    style: TextStyle(
                        color: Colors.black,
                        fontSize: 18,
                        fontWeight: FontWeight.w600
                    ), // TextStyle
                ),
            ),
        ],
    );
}
```

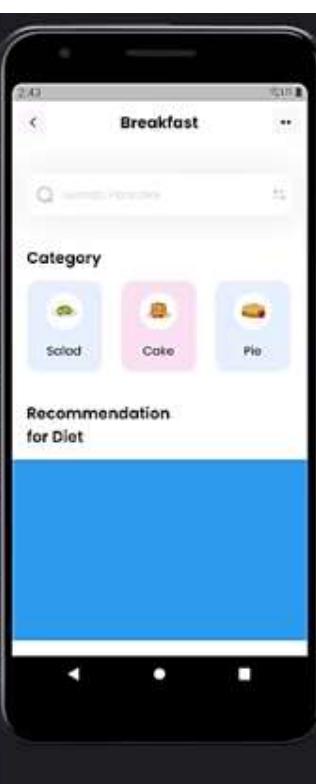
Another text

```
lib > pages > home.dart > HomePage > build
25   children: [
26     _searchField(),
27     SizedBox(height: 40,),
28     _categoriesSection(),
29     SizedBox(height: 40,),
30   ],
31   Column(
32     children: [
33       Padding(
34         padding: const EdgeInsets.only(left:20),
35         child: Text(
36           'Recommendation\nfor Diet',
37           style: TextStyle(
38             color: Colors.black,
39             fontSize: 18,
40             fontWeight: FontWeight.w600
41           ), // TextStyle
42         ), // Padding
43       ],
44     ),
45   ],
46 ), // Column
47 ); // Scaffold
48 }
49
50 Column _categoriesSection() {
51   return Column(
52     crossAxisAlignment: CrossAxisAlignment.start,
```

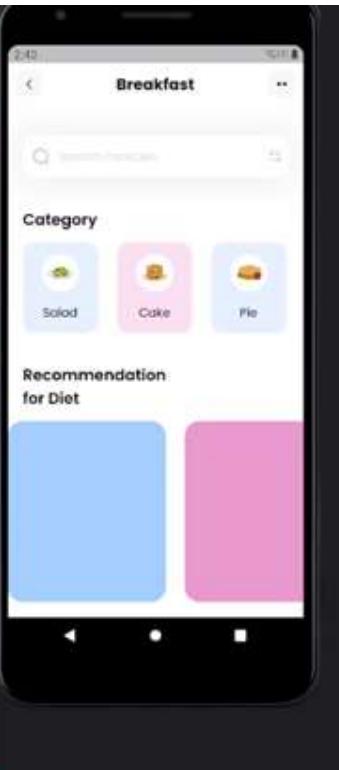


giving color

```
main.dart  home.dart  diet_model.dart  category_model.dart
lib > pages > home.dart > HomePage > build
25   children: [
26     _searchField(),
27     SizedBox(height: 40,),
28     _categoriesSection(),
29     SizedBox(height: 40,),
30   ],
31   Column(
32     crossAxisAlignment: CrossAxisAlignment.start,
33     children: [
34       Padding(
35         padding: const EdgeInsets.only(left:20),
36         child: Text(
37           'Recommendation\nfor Diet',
38           style: TextStyle(
39             color: Colors.black,
40             fontSize: 18,
41             fontWeight: FontWeight.w600
42           ), // TextStyle
43         ), // Padding
44       ),
45       Container(
46         color: Colors.blue,
47         height: 240,
48       ), // Container
49     ],
50   ), // Column
51 },
52 ), // Column
53 ); // Scaffold
54 }
```



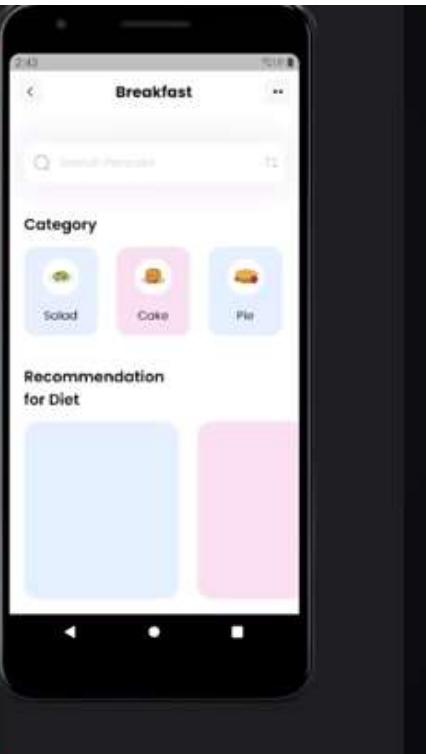
Bottom horizontal list



The screenshot shows the code for a Flutter application's homepage. The UI consists of a top navigation bar with a search icon, followed by a section titled "Category" containing three items: "Salad" (blue), "Cake" (pink), and "Pie" (light blue). Below this is a section titled "Recommendation for Diet" with two large, semi-transparent blue and pink rectangular boxes. The code uses a `ListView.separated` with horizontal scroll direction to create the bottom list.

```
lib> pages > HomePage > build
42     ), // Text
43     ), // Padding
44     SizedBox(height: 15.),
45     Container(
46       height: 240,
47       child: ListView.separated(
48         itemBuilder: (context, index) {
49           return Container(
50             width: 210,
51             decoration: BoxDecoration(
52               color: diets[index].boxColor.w,
53               borderRadius: BorderRadius.circular(20)
54             ), // BoxDecoration
55           ); // Container
56         },
57         separatorBuilder: (context, index) => SizedBox(width: 25.),
58         itemCount: diets.length,
59         scrollDirection: Axis.horizontal,
60       ), // ListView.separated
61     ); // Container
62   ],
63 ), // Column
64 ),
65 ), // Column
66 ); // Scaffold
67
68
69 Column _categoriesSection() {
70   return Column(
71     crossAxisAlignment: CrossAxisAlignment.start,
```

Making opacity less

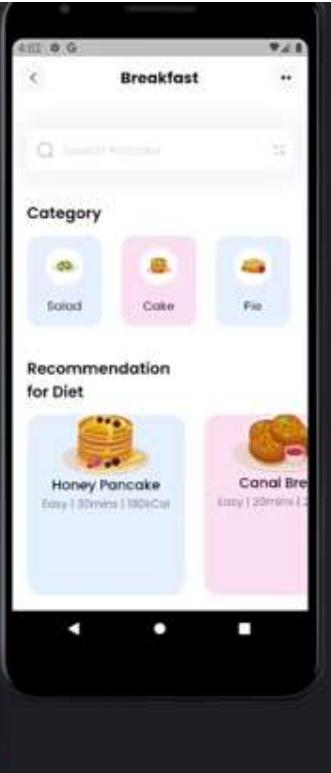


The screenshot shows the same application after modifying the code to reduce the opacity of the list items. The UI remains the same, but the background color of the list items is now lighter, appearing more translucent. The code change is in line 52, where the `color` property of the `BoxDecoration` is modified to include `.withOpacity(0.3)`.

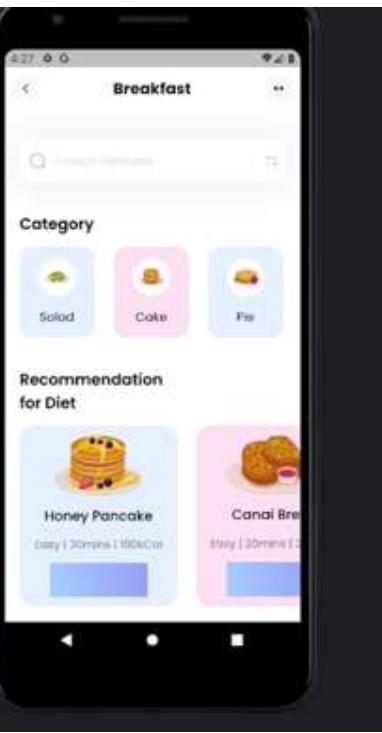
```
lib> pages > HomePage > build
39     color: diets[index].boxColor,
40     fontSize: 18,
41     fontWeight: FontWeight.w600
42   ), // Text
43   ), // Padding
44   SizedBox(height: 15.),
45   Container(
46     height: 240,
47     child: ListView.separated(
48       itemBuilder: (context, index) {
49         return Container(
50           width: 210,
51           decoration: BoxDecoration(
52             color: diets[index].boxColor.withOpacity(0.3),
53             borderRadius: BorderRadius.circular(20)
54           ), // BoxDecoration
55         ); // Container
56       },
57       separatorBuilder: (context, index) => SizedBox(width: 25.),
58       itemCount: diets.length,
59       scrollDirection: Axis.horizontal,
60       padding: EdgeInsets.only(
61         Left: 20,
62         right: 20
63       ), // Padding
64     ), // Container
65   ], // Column
66 ); // Scaffold
67
68
69 Column _categoriesSection() {
70   return Column(
71     crossAxisAlignment: CrossAxisAlignment.start,
```

Adding icon

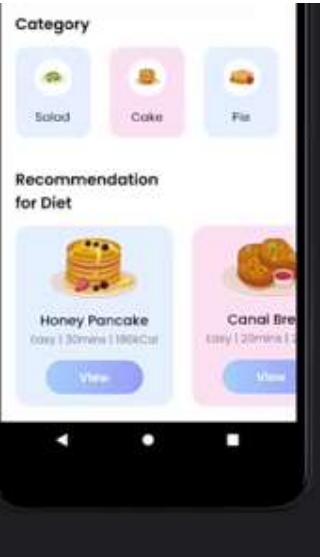
```
lib> pages > home.dart > HomePage > build
59
60         text(
61             diets[index].name,
62             style: TextStyle(
63                 fontWeight: FontWeight.w500,
64                 color: Colors.black,
65                 fontSize: 16
66             ), // TextStyle
67         ), // Text
68         Text(
69             diets[index].level + ' | ' + diets[index].duration +
70             style: TextStyle(
71                 color: Color(0xff7B6872),
72                 fontSize: 13,
73                 fontWeight: FontWeight.w400
74             ), // TextStyle
75         ), // Text
76     ],
77     ), // Column
78 ); // Container
79 },
80 separatorBuilder: (context, index) => SizedBox(width: 25),
81 itemCount: diets.length,
82 scrollDirection: Axis.horizontal,
83 padding: EdgeInsets.only(
84     left: 20,
85     right: 20
86 ), // EdgeInsets.only
87 ), // ListView.separated
88 ) // Container
89 
```



```
lib > pages > home.dart > HomePage > build
52
53   color: diets[index].boxColor.withOpacity(0.3),
54   borderRadius: BorderRadius.circular(20)
55 ),
56   // BoxDecoration
57   child: Column(
58     mainAxisAlignment: MainAxisAlignment.spaceEvenly,
59     children: [
60       SvgPicture.asset(diets[index].iconPath),
61       Text(
62         diets[index].name,
63         style: TextStyle(
64           fontWeight: FontWeight.w500,
65           color: Colors.black,
66           fontSize: 16
67         ), // TextStyle
68       ), // Text
69       Text(
70         diets[index].level + ' | ' + diets[index].duration +
71         style: TextStyle(
72           color: Color(0xFF7B6872),
73           fontSize: 13,
74           fontWeight: FontWeight.w400
75         ), // TextStyle
76       ), // Text
77       Container(
78         height: 45,
79         width: 130,
80         decoration: BoxDecoration(
```



Radius and text



A screenshot of a mobile application interface. At the top, there is a navigation bar with three items: "Salad" (blue), "Coke" (pink), and "Fis" (light blue). Below the navigation bar is a section titled "Recommendation for Diet". This section contains two cards: "Honey Pancake" (blue card) and "Canal Bre" (pink card). Each card has an image of the food, the name, a description, and a "View" button at the bottom.

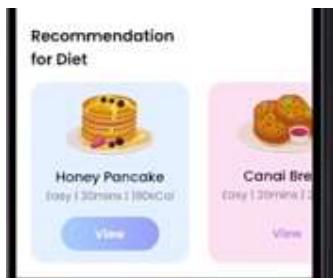
```
Container(
  height: 45,
  width: 130,
  child: Center(
    child: Text(
      'View',
      style: TextStyle(
        color: Colors.white,
        fontWeight: FontWeight.w600,
        fontSize: 14
      ),
    ), // Text
  ), // Center
  decoration: BoxDecoration(
    gradient: LinearGradient(
      colors: [
        Color(0xff9DCEFF),
        Color(0xff92A3FD)
      ],
    ), // LinearGradient
    borderRadius: BorderRadius.circular(50)
  ), // BoxDecoration
), // Container
```

Selected list color



A screenshot of a mobile application interface. A yellow arrow points to a "View" button on the "Canal Bre" card, indicating it is the selected item. The code on the left shows the state of the "View" button based on the index of the selected item.

```
), // Center
decoration: BoxDecoration(
  gradient: LinearGradient(
    colors: [
      diets[index].viewIsSelected ? Color(0xff9DCEFF) : Colors.transparent,
      diets[index].viewIsSelected ? Color(0xff92A3FD) : Colors.transparent
    ],
  ), // LinearGradient
borderRadius: BorderRadius.circular(50)
), // BoxDecoration
// Container
```



- UI Designing



hexcolor 3.0.1

Published 4 months ago | Dart 3 ready

SDK FLUTTER PLATFORM ANDROID IOS LINUX MACOS WEB WINDOWS

252

[Readme](#) [Changelog](#) [Installing](#) [Versions](#) [Scores](#)

[license](#) [GitHub](#) [pub.dev](#) 213

hexcolor_example

how to use the hexcolor plugin.

hex color plugin allows you to add hex color codes to your flutter projects.

Getting Started

creating style file

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

Color yellowColor = HexColor("#f7a41d");
Color greenColor = HexColor("#2ea96d");
Color blueGreyColor = HexColor("#429ebd");
Color indigoColor = HexColor("#58359c");
Color orangeColor = HexColor("#f1770c");
Color lightBlueColor = HexColor("eaceff");
Color redColor = HexColor("#ff0000");
Color whiteColor = Colors.white;
Color blackColor = Colors.black;
Color? grey100 = Colors.grey[100];
Color grey300 = const Color(0xFFE6E6E6);
Color greyColor = Colors.grey;

class Style {
    static TextStyle headlineStyle1 = TextStyle(fontSize: 15,color: blackColor,fontWeight: FontWeight.w500);
    static TextStyle headlineStyle2 = TextStyle(fontSize: 18,color: blackColor);
    static TextStyle headlineStyle3 = TextStyle(fontSize: 18,color: blackColor,fontWeight: FontWeight.bold);
    static TextStyle headlineStyle4 = TextStyle(fontSize: 20,color: blackColor,fontWeight: FontWeight.w300);
    static TextStyle headlineStyle5 = TextStyle(fontSize: 22,color: blackColor,fontWeight: FontWeight.w500);
}
```

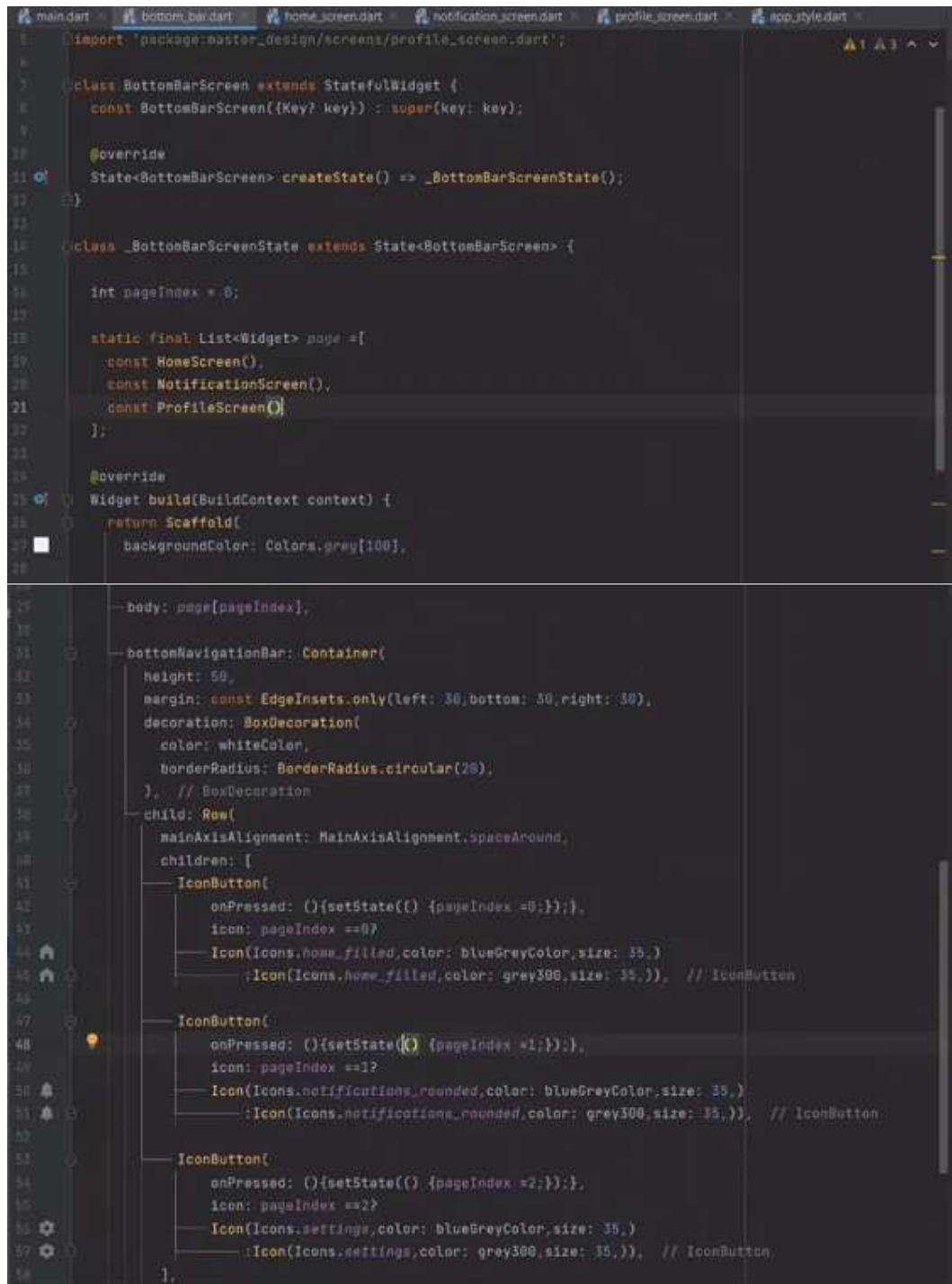
Creating bottom navigation

```
@override
Widget build(BuildContext context) {
    return Scaffold(
        backgroundColor: Colors.grey[100],
        body: page[pageIndex],
        bottomNavigationBar: Container(
            height: 50,
            margin: const EdgeInsets.only(left: 30,bottom: 30,right: 30),
            decoration: BoxDecoration(
                color: whiteColor,
                borderRadius: BorderRadius.circular(20),
            ), // BoxDecoration
            child: Row(
                mainAxisAlignment: MainAxisAlignment.spaceAround,
                children: [
                    IconButton(
                        onPressed: (){setState(() {
                            pageIndex = 0;
                        });
                    },
                    icon: Icon(Icons.home_filled)), // IconButton

                    IconButton(
                        onPressed: (){},
                        icon: Icon(Icons.notifications_rounded)), // IconButton

                    IconButton(
                        onPressed: (){},
                        icon: Icon(Icons.settings)) // IconButton
                ],
            ),
        ),
    );
}
```

Separate file for bottom navigation and navigating based on index



```
import 'package:master_design/screens/profile_screen.dart';

class BottomBarScreen extends StatefulWidget {
  const BottomBarScreen({Key? key}) : super(key: key);

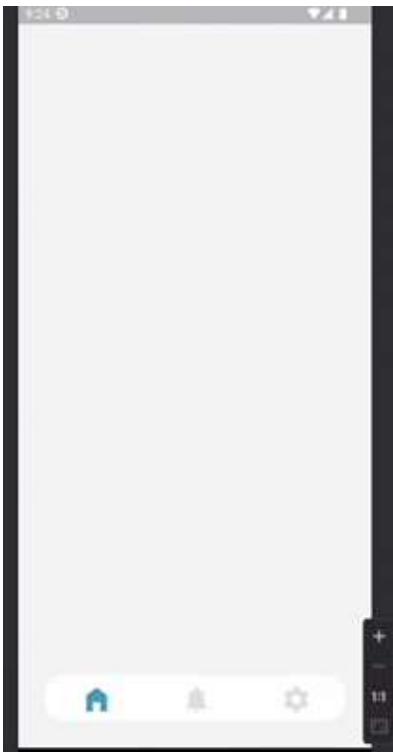
  @override
  State<BottomBarScreen> createState() => _BottomBarScreenState();
}

class _BottomBarScreenState extends State<BottomBarScreen> {
  int pageIndex = 0;

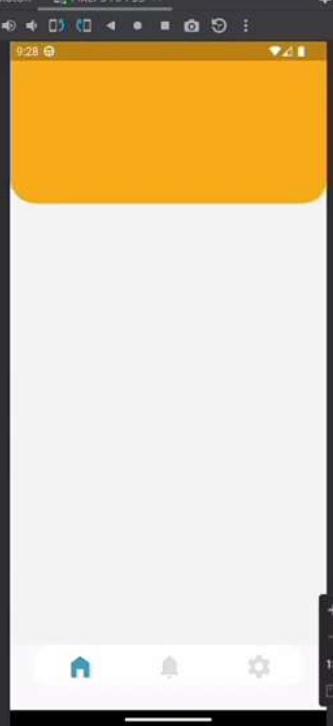
  static final List<Widget> page = [
    const HomeScreen(),
    const NotificationScreen(),
    const ProfileScreen()
  ];

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: Colors.grey[100],
      body: page[pageIndex],
      bottomNavigationBar: Container(
        height: 50,
        margin: const EdgeInsets.only(left: 30, bottom: 30, right: 30),
        decoration: BoxDecoration(
          color: whiteColor,
          borderRadius: BorderRadius.circular(20),
        ), // BoxDecoration
        child: Row(
          mainAxisAlignment: MainAxisAlignment.spaceAround,
          children: [
            IconButton(
              onPressed: () { setState(() { pageIndex = 0; }); },
              icon: pageIndex == 0 ?
                Icon(Icons.home_filled, color: blueGreyColor, size: 35,) :
                Icon(Icons.home_filled, color: grey300, size: 35,), // IconButton
            ),
            IconButton(
              onPressed: () { setState(() { pageIndex = 1; }); },
              icon: pageIndex == 1 ?
                Icon(Icons.notifications_rounded, color: blueGreyColor, size: 35,) :
                Icon(Icons.notifications_rounded, color: grey300, size: 35,), // IconButton
            ),
            IconButton(
              onPressed: () { setState(() { pageIndex = 2; }); },
              icon: pageIndex == 2 ?
                Icon(Icons.settings, color: blueGreyColor, size: 35,) :
                Icon(Icons.settings, color: grey300, size: 35,), // IconButton
            ),
          ],
        ),
      ),
    );
  }
}
```

Output



➤ Creating homescreen



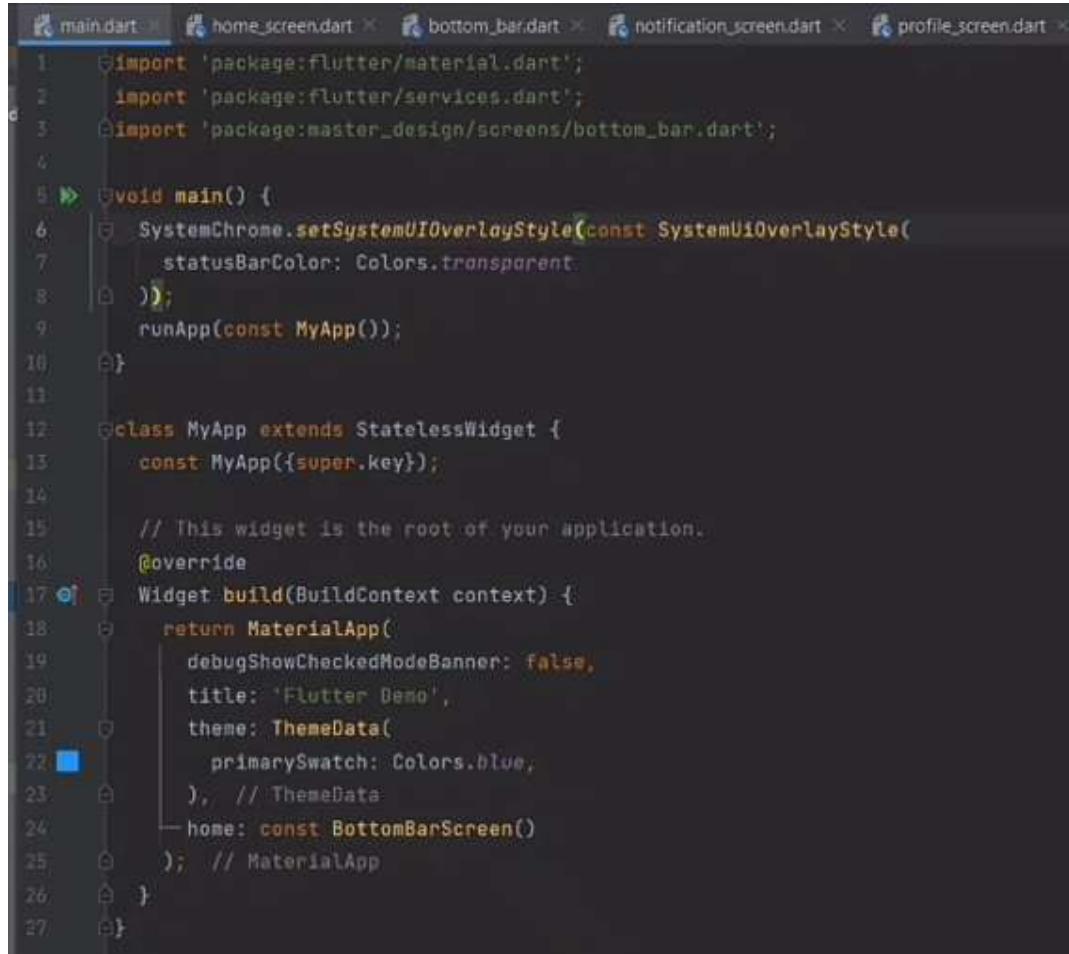
The screenshot shows the Android Studio interface with the code editor on the left and the emulator window on the right. The code editor displays the `home_screen.dart` file, which contains the following Dart code:

```
import '../app_style.dart';

class HomeScreen extends StatelessWidget {
  const HomeScreen({Key? key}) : super(key: key);

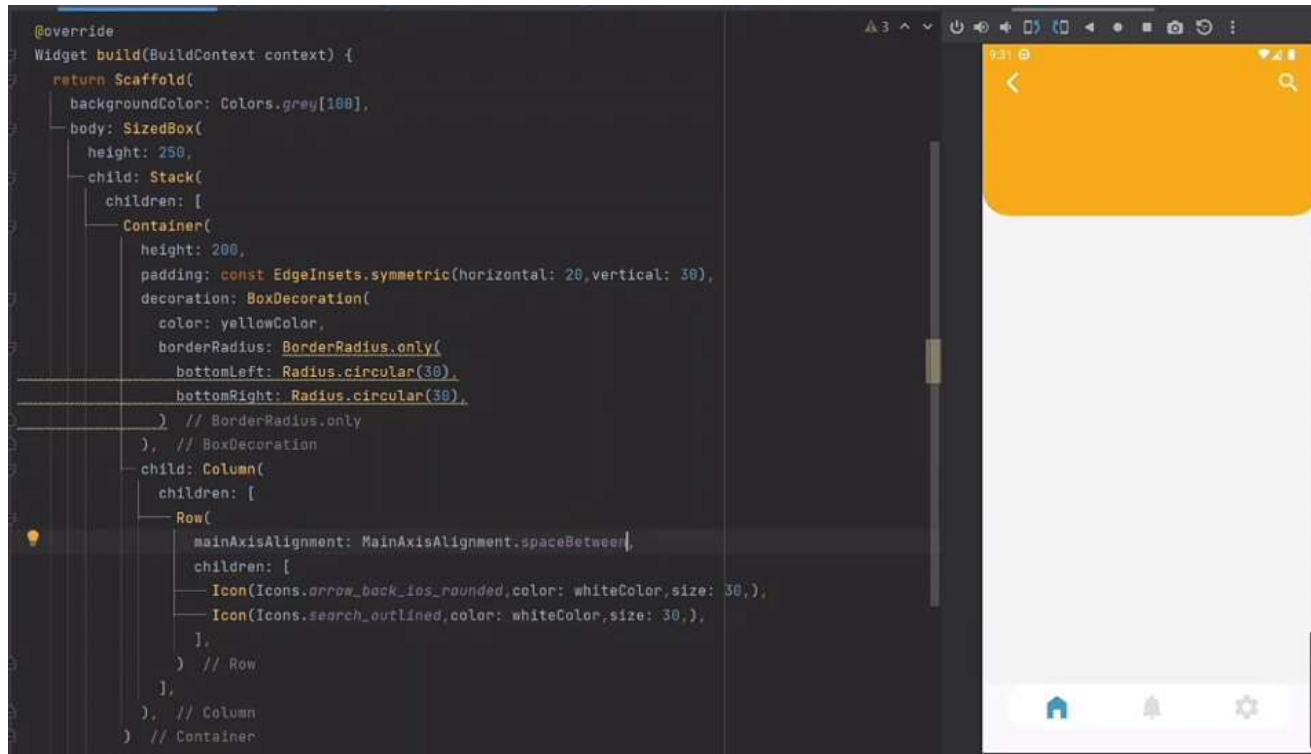
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: Colors.grey[100],
      body: SizedBox(
        height: 250,
        child: Stack(
          children: [
            Container(
              height: 200,
              padding: const EdgeInsets.symmetric(horizontal: 20, vertical: 30),
              decoration: BoxDecoration(
                color: yellowColor,
                borderRadius: BorderRadius.only(
                  bottomLeft: Radius.circular(30),
                  bottomRight: Radius.circular(30),
                ),
              ),
            ),
          ],
        ),
      ),
    );
  }
}
```

Creating transparent statusbar



```
1 import 'package:flutter/material.dart';
2 import 'package:flutter/services.dart';
3 import 'package:master_design/screens/bottom_bar.dart';
4
5 void main() {
6     SystemChrome.setSystemUIOverlayStyle(const SystemUiOverlayStyle(
7         statusBarColor: Colors.transparent,
8     ));
9     runApp(const MyApp());
10 }
11
12 class MyApp extends StatelessWidget {
13     const MyApp({super.key});
14
15     // This widget is the root of your application.
16     @override
17     Widget build(BuildContext context) {
18         return MaterialApp(
19             debugShowCheckedModeBanner: false,
20             title: 'Flutter Demo',
21             theme: ThemeData(
22                 primarySwatch: Colors.blue,
23             ), // ThemeData
24             home: const BottomBarScreen()
25         ); // MaterialApp
26     }
27 }
```

Added icons as well using stack



```
1 @override
2 Widget build(BuildContext context) {
3     return Scaffold(
4         backgroundColor: Colors.grey[100],
5         body: SizedBox(
6             height: 250,
7             child: Stack(
8                 children: [
9                     Container(
10                         height: 200,
11                         padding: const EdgeInsets.symmetric(horizontal: 20, vertical: 30),
12                         decoration: BoxDecoration(
13                             color: yellowColor,
14                             borderRadius: BorderRadius.only(
15                                 bottomLeft: Radius.circular(30),
16                                 bottomRight: Radius.circular(30),
17                             ), // BorderRadius.only
18                         ), // BoxDecoration
19                         child: Column(
20                             children: [
21                                 Row(
22                                     mainAxisAlignment: MainAxisAlignment.spaceBetween,
23                                     children: [
24                                         Icon(Icons.arrow_back_ios_rounded, color: whiteColor, size: 30, ),
25                                         Icon(Icons.search_outlined, color: whiteColor, size: 30, ),
26                                     ],
27                                 ), // Row
28                             ],
29                         ), // Column
30                     ) // Container
31     );
32 }
```

Added text

The screenshot shows the Android Studio interface with the code editor on the left and an emulator preview on the right.

Code Editor (main.dart):

```
33 ht: 250,
34 d: Stack(
35   children: [
36     Container(
37       height: 280,
38       padding: const EdgeInsets.symmetric(horizontal: 20, vertical: 30),
39       decoration: BoxDecoration(
40         color: yellowColor,
41         borderRadius: BorderRadius.only(
42           bottomLeft: Radius.circular(30),
43           bottomRight: Radius.circular(30),
44         ) // BorderRadius.only
45       ), // BoxDecoration
46       child: Column(
47         children: [
48           Row(...), // Row
49           Text("Lorem", style: Style.headLineStyle5.copyWith(fontSize: 30, fontWeight: FontWeight.w400),),
50         ],
51       ), // Column
52     ), // Container
53   ), // Stack
54   SizedBox(),
55   Scaffold(
56     body: Container(
57       color: Colors.white,
58       child: Column(
59         children: [
60           Container(
61             height: 100,
```

Emulator Preview:

The emulator shows a yellow rounded rectangular box containing the text "Lorem". The top-left and bottom-right corners are rounded, while the top-right and bottom-left corners are straight. The text is centered within the box.

Spacer and alignment end

Positioned the circle over container

The screenshot shows the Android Studio IDE with the code for `home_screen.dart` open. The code defines a `Stack` widget with a `Positioned` child. This positioned child contains two `Container` widgets: one with a white background and circular shape, and another with a light blue background and circular shape. The blue container has a `NetworkImage` as its `image`. The `main.dart` tab is also visible in the top bar.

```
14     child: Stack(
15       children: [
16         Container(...), // Container
17
18         Positioned(
19           left: -10, bottom: 0,
20           child: Container(
21             height: 100, width: 100,
22             decoration: BoxDecoration(
23               color: whiteColor,
24               shape: BoxShape.circle
25             ), // BoxDecoration
26             child: Container(
27               height: 68, width: 68,
28               decoration: BoxDecoration(
29                 shape: BoxShape.circle,
30                 image: DecorationImage(
31                   fit: BoxFit.fill,
32                   image: NetworkImage("https://img.freepik.com/free-photo/background_53876-32169.jpg")
33                 ), // BoxDecoration
34               ), // Container
35             ), // Container
36           ), // Positioned
37         ],
38       ), // Stack
39     ), // SizedBox
40   ), // Scaffold
41 }
```

Center

The screenshot shows the Android Studio interface with the code editor on the left and the emulator on the right. The code editor displays the main.dart file, which contains Dart code for a UI component. The emulator shows a yellow screen with a circular icon and placeholder text 'Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. Ut enim ad minim veniam, quis nostrud exerci tatione'.

```
14 child: Stack(
15   children: [
16     Container(...), // Container
17
18     Positioned(
19       left: -10, bottom: 0,
20       child: Container(
21         height: 100, width: 100,
22         decoration: BoxDecoration(
23           color: whiteColor,
24           shape: BoxShape.circle),
25       ), // BoxDecoration
26       child: Center(
27         child: Container(
28           height: 60, width: 60,
29           decoration: BoxDecoration(
30             shape: BoxShape.circle,
31             image: DecorationImage(
32               fit: BoxFit.fill,
33               image: NetworkImage("https://img.freepik.com/free-photo/background_53876-32169.jpg"),
34             ),
35           ), // BoxDecoration
36         ), // Center
37       ), // Container
38     ) // Positioned
39   ],
40 ),
41 ), // Stack
42 ), // SizedBox
43 // Scaffold
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
73
74
75
76
77
78
```

Align text

The screenshot shows the `main.dart` file open in a code editor, specifically the `home_screen.dart` tab. The code defines a `HomeScreen` widget that extends `StatelessWidget`. It overrides the `build` method to return a `Scaffold` with a grey background. The `body` is a `Column` containing a `Profile` icon and an `Align` widget. Inside the `Align`, there is a `Padding` with a right padding of 20, a `Column` with two `Text` elements, and a `Column` with two `Text` elements. The application's UI on the right shows a yellow header with the word "Lorem". Below it is a white card with a blue circular profile icon, the text "Lorem ipsum", and a smaller "Lorem ipsum" below it.

```
const HomeScreen({Key? key}) : super(key: key);

@override
Widget build(BuildContext context) {
  return Scaffold(
    backgroundColor: Colors.grey[100],
    body: Column(
      children: [
        const Profile(),
        Align(
          alignment: Alignment.centerRight,
          child: Padding(
            padding: const EdgeInsets.only(right: 20),
            child: Column(
              crossAxisAlignment: CrossAxisAlignment.end,
              children: [
                Text("Lorem ipsum", style: Style.headLineStyle5.copyWith(fontWeight: FontWeight.bold, color: Colors.black)),
                Text("Lorem ipsum", style: Style.headLineStyle5.copyWith(color: Colors.grey[700], fontWeight: FontWeight.normal)),
              ],
            ),
          ),
        ),
      ],
    );
}
```

Adding row for orange color

The screenshot shows the same `main.dart` file with an additional `Row` widget added to the `body` of the `Scaffold`. The `Row` contains a single `Container` with an orange background, a height of 120, a width of 100, rounded corners, and a decoration matching the orange color. The application's UI on the right now includes an orange square placeholder where the previous text content was located.

```
class HomeScreen extends StatelessWidget {
  const HomeScreen({Key? key}) : super(key: key);

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: Colors.grey[100],
      body: Column(
        children: [
          const Profile(),
          Align(...), // Align

          Row(
            children: [
              Container(
                margin: EdgeInsets.symmetric(horizontal: 20, vertical: 10),
                height: 120, width: 100,
                decoration: BoxDecoration(
                  color: orangeColor,
                  borderRadius: BorderRadius.circular(20)
                ), // BoxDecoration
              ) // Container
            ],
          ), // Row
        ],
      );
}
```

Adding icon in orange

```
1 children: [
2   const Profile(),
3   Align(...), // Align
4
5   Row(
6     children: [
7       Container(
8         margin: EdgeInsets.symmetric(horizontal: 20, vertical: 10),
9         height: 120, width: 100,
10        decoration: BoxDecoration(
11          color: orangeColor,
12          borderRadius: BorderRadius.circular(20)
13        ), // BoxDecoration
14        child: Center(
15          child: Container(
16            height: 30, width: 50,
17            decoration: BoxDecoration(
18              border: Border.all(color: whiteColor),
19              borderRadius: BorderRadius.circular(5)
20            ), // BoxDecoration
21            child: Center(child: Icon(Icons.circle_outlined, color: whiteColor, size: 10,),),
22          ), // Container
23        ) // Center
24      ), // Container
25    ],
26  ], // Row
27 ],
28 ) // Column
```



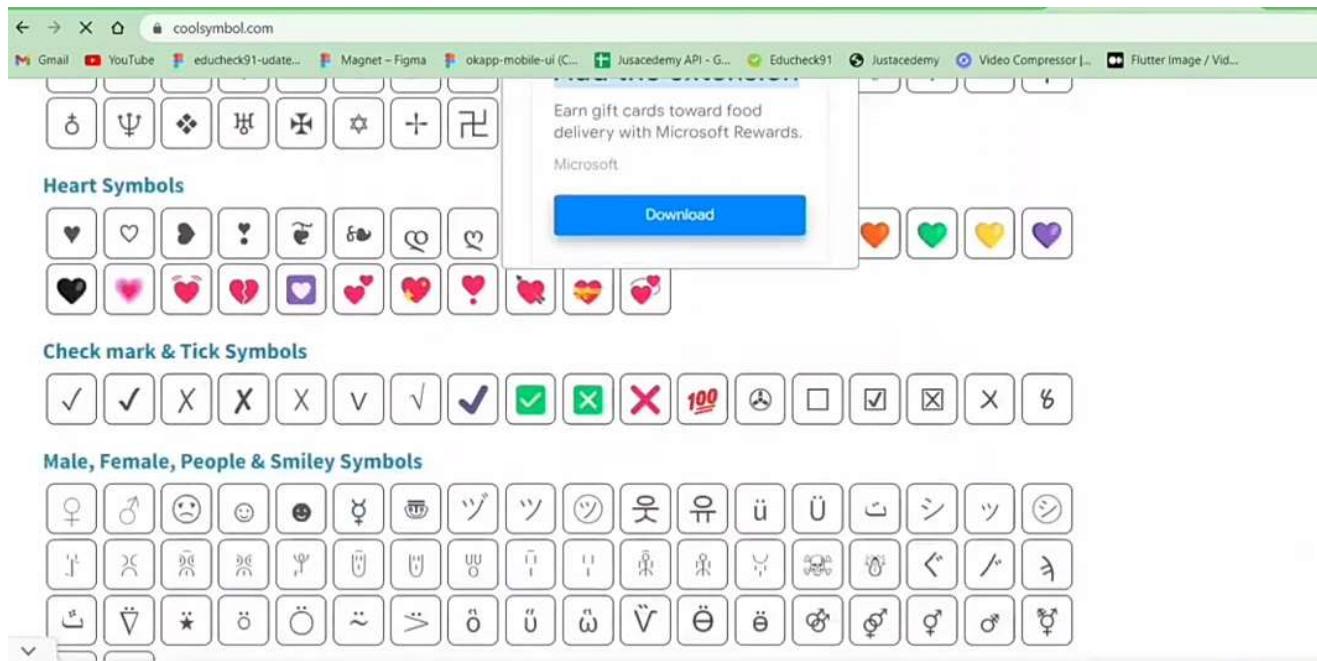
Using expanded for right side content

```
11 Scaffold(
12   backgroundColor: Colors.grey[100],
13   body: Column(
14     children: [
15       const Profile(),
16       Align(...), // Align
17
18       Row(
19         children: [
20           Container(...), // Container
21
22           Expanded(
23             child: Container(
24               margin: const EdgeInsets.only(top: 20, bottom: 10, right: 20),
25               padding: EdgeInsets.all(20),
26               decoration: BoxDecoration(
27                 color: whiteColor,
28                 borderRadius: BorderRadius.circular(20)
29               ), // BoxDecoration
30               child: Column(
31                 children: [
32                   Text("Lorem ipsum", style: Style.headLineStyle3.copyWith(fontWeight: FontWeight.w300),),
33                   SizedBox(height: 10),
34                   Text("Lorem ipsum dolor sit amet", style: Style.headLineStyle1),
35                   Text(""),
36                 ],
37               ), // Column
38             ), // Container
39           ), // Expanded
40         ],
41       ), // Row
42     ],
43   ),
44 
```

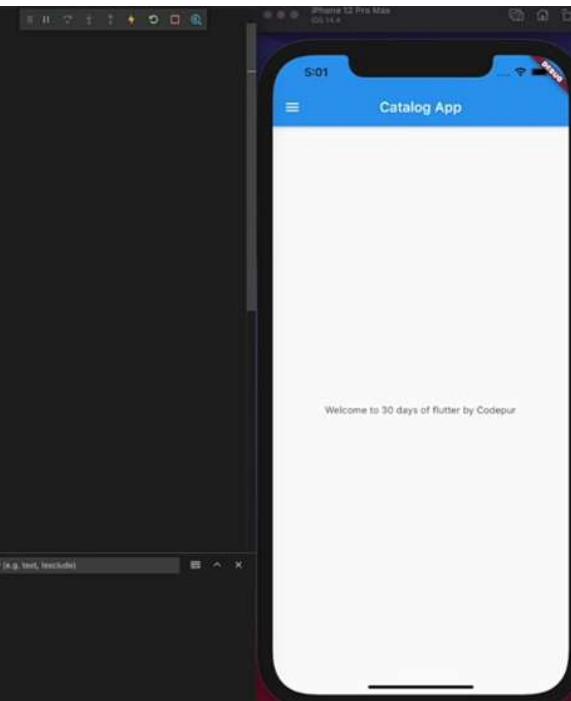


Positioned for placing the icon

```
18 children: [
19   Container(...), // Container
20
21   Positioned(
22     bottom: 20, left: -10,
23     child: Container(
24       height: 180, width: 360,
25       decoration: BoxDecoration(
26         color: orangeColor,
27         borderRadius: BorderRadius.circular(40)
28       ), // BoxDecoration
29       child: Column(
30         children: [
31           CircleAvatar(
32             radius: 30,
33             backgroundImage: NetworkImage("https://cdn-icons-png.flaticon.com/512/3135/3135715.png"),
34           ), // CircleAvatar
35           Text("L"),
36         ], // Column
37       ), // Container, Positioned
38     ],
39   ), // Stack
40   SizedBox(),
41   Container(),
42   Column(
43   ), // Column
44 ] // Scaffold
```



- Theme color and side drawer



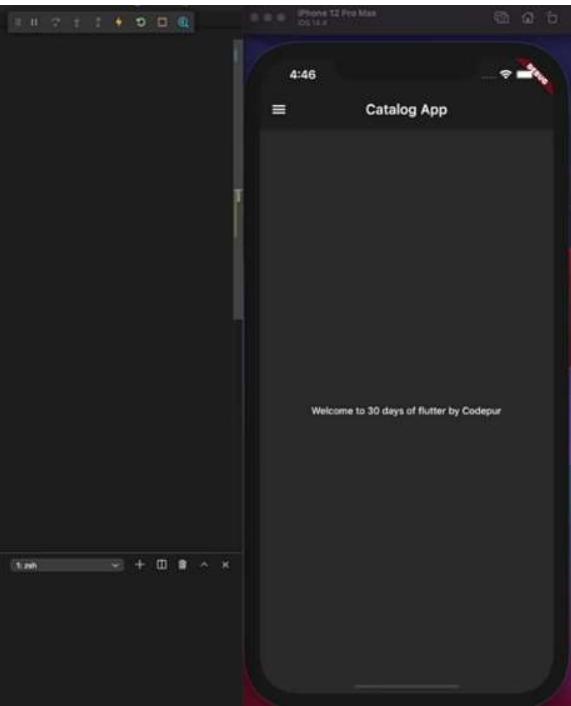
A screenshot of a Flutter application running on an iPhone 12 Pro Max. The app has a light blue header bar with the title "Catalog App". The main content area is white and displays the text "Welcome to 30 days of flutter by Codepur". The code editor shows the `home_page.dart` file with the following code:

```
1 import 'package:flutter/material.dart';
2
3 class HomePage extends StatelessWidget {
4   final int days = 30;
5   final String name = "Codepur";
6   @override
7   Widget build(BuildContext context) {
8     return Scaffold(
9       appBar: AppBar(
10         title: Text("Catalog App"),
11       ), // AppBar
12       body: Center(
13         child: Container(
14           child: Text("Welcome to \$days days of flutter by \$name"),
15         ), // Container
16       ), // Center
17       drawer: Drawer(),
18     ); // Scaffold
19   }
20 }
```

The terminal below the code editor shows the following log output:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Restarted application in 417ms.
Reloaded 2 of 537 libraries in 531ms.
Reloaded 2 of 537 libraries in 423ms.
Reloaded 2 of 537 libraries in 523ms.
Reloaded 2 of 537 libraries in 330ms.
```

Dark theme



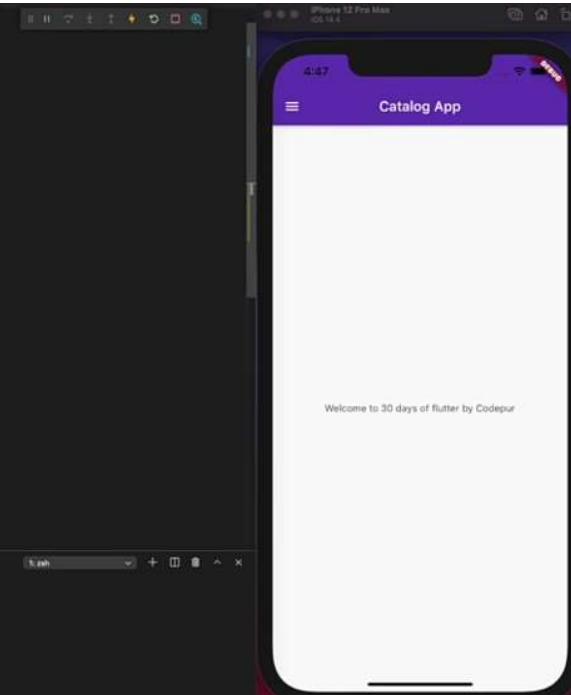
A screenshot of a Flutter application running on an iPhone 12 Pro Max. The app has a dark header bar with the title "Catalog App". The main content area is white and displays the text "Welcome to 30 days of flutter by Codepur". The code editor shows the `main.dart` file with the following code:

```
2
3 import 'pages/home_page.dart';
4
5 void main() {
6   runApp(MyApp());
7 }
8
9 class MyApp extends StatelessWidget {
10   @override
11   Widget build(BuildContext context) {
12     return MaterialApp(
13       home: HomePage(),
14       themeMode: ThemeMode.dark,
15       darkTheme: ThemeData(
16         brightness: Brightness.dark,
17       ), // ThemeData
18     ); // MaterialApp
19   }
20 }
```

The terminal below the code editor shows the following log output:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
flutter_catalog ✘ []
```

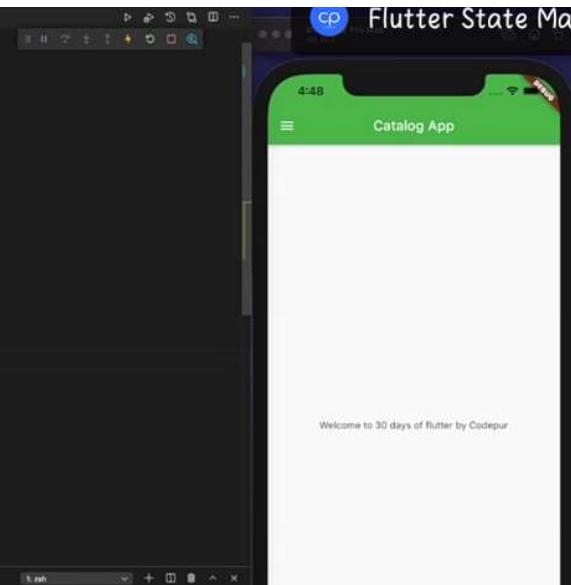
Light theme



The screenshot shows the code for a Flutter application named 'Catalog App'. The main.dart file defines the application's entry point. It imports the necessary packages and defines the main function to run the MyApp widget. The MyApp class extends StatelessWidget and overrides the build method to return a MaterialApp. The MaterialApp configuration includes a home page (HomePage), a light theme mode, a primary swatch color of deepPurple, a dark theme with a green primary swatch, and a brightness level of dark.

```
1 import 'package:flutter/material.dart';
2
3 import 'pages/home_page.dart';
4
5 void main() {
6   runApp(MyApp());
7 }
8
9 class MyApp extends StatelessWidget {
10   @override
11   Widget build(BuildContext context) {
12     return MaterialApp(
13       home: HomePage(),
14       themeMode: ThemeMode.light,
15       theme: ThemeData(primarySwatch: Colors.deepPurple),
16       darkTheme: ThemeData(
17         primarySwatch: Colors.green,
18       ), // ThemeData
19     ); // MaterialApp
20   }
21 }
22
```

Swatch color



The screenshot shows the code for the same 'Catalog App' application, but with a green theme. The primarySwatch color in the ThemeData is now set to Colors.green, while the darkTheme's primarySwatch remains Colors.deepPurple. The rest of the code is identical to the previous screenshot.

```
1 import 'package:flutter/material.dart';
2
3 import 'pages/home_page.dart';
4
5 void main() {
6   runApp(MyApp());
7 }
8
9 class MyApp extends StatelessWidget {
10   @override
11   Widget build(BuildContext context) {
12     return MaterialApp(
13       home: HomePage(),
14       themeMode: ThemeMode.dark,
15       theme: ThemeData(primarySwatch: Colors.deepPurple),
16       darkTheme: ThemeData(
17         primarySwatch: Colors.green,
18       ), // ThemeData
19     ); // MaterialApp
20   }
21 }
22
```

- Routing in flutter

```
main.dart x home_page.dart pubspec.yaml
lib/main.dart MyApp build Run | Debug
5 void main() {
6   runApp(MyApp());
7 }
8
9 class MyApp extends StatelessWidget {
10   @override
11   Widget build(BuildContext context) {
12     return MaterialApp(
13       themeMode: ThemeMode.dark,
14       theme: ThemeData(primarySwatch: Colors.deepPurple),
15       darkTheme: ThemeData(
16         brightness: Brightness.dark,
17       ), // ThemeData
18       routes: {
19         "/": (context) => HomePage(),
20       },
21     ); // MaterialApp
22   }
23 }
24
```

And

```
main.dart x MyApp build
Run | Debug
6 void main() {
7   runApp(MyApp());
8 }
9
10 class MyApp extends StatelessWidget {
11   @override
12   Widget build(BuildContext context) {
13     return MaterialApp(
14       themeMode: ThemeMode.dark,
15       theme: ThemeData(primarySwatch: Colors.deepPurple),
16       darkTheme: ThemeData(
17         brightness: Brightness.dark,
18       ), // ThemeData
19       routes: {
20         "/": (context) => LoginPage(),
21         "/home": (context) => HomePage(),
22         "/login": (context) => LoginPage(),
23       },
24     ); // MaterialApp
25   }
26 }
27
```

The screenshot shows the Flutter development environment. On the left, the code editor displays the main.dart file with the following code. On the right, an iPhone 12 Pro Max simulator is running, showing a dark-themed login screen with the text "Login Page".

```
main.dart x MyApp build
Run | Debug
6 void main() {
7   runApp(MyApp());
8 }
9
10 class MyApp extends StatelessWidget {
11   @override
12   Widget build(BuildContext context) {
13     return MaterialApp(
14       themeMode: ThemeMode.dark,
15       theme: ThemeData(primarySwatch: Colors.deepPurple),
16       darkTheme: ThemeData(
17         brightness: Brightness.dark,
18       ), // ThemeData
19       routes: {
20         "/": (context) => LoginPage(),
21         "/home": (context) => HomePage(),
22         "/login": (context) => LoginPage(),
23       },
24     ); // MaterialApp
25   }
26 }
27
```

Light theme

A screenshot of an iPhone 12 Pro Max simulator displaying a white screen with the text "Login Page" centered in blue font.

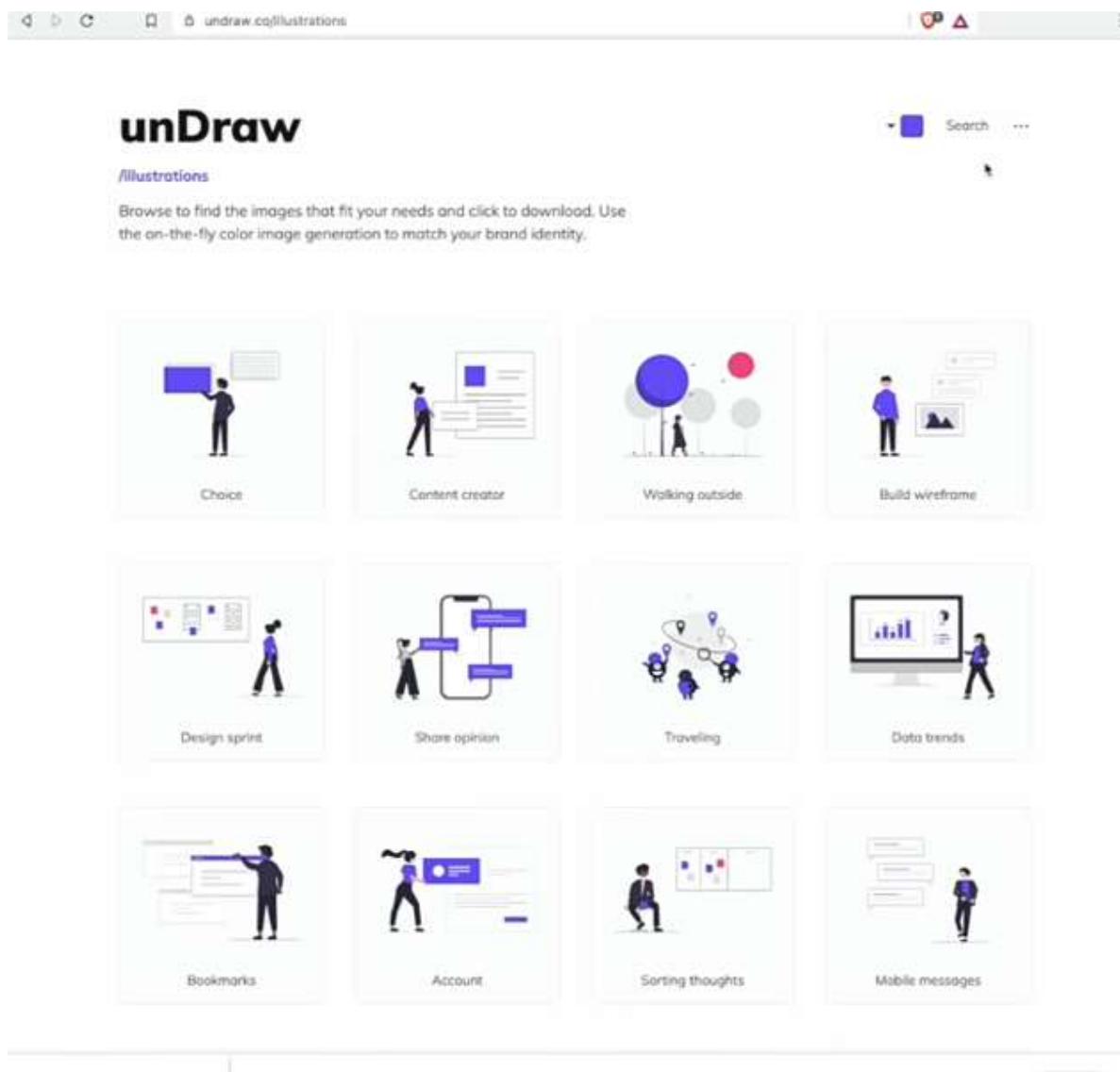
```
lib> main.dart > MyApp > build
5
6 void main() {
7   runApp(MyApp());
8 }
9
10 class MyApp extends StatelessWidget {
11   @override
12   Widget build(BuildContext context) {
13     return MaterialApp(
14       themeMode: ThemeMode.light,
15       theme: ThemeData(primarySwatch: Colors.deepPurple),
16       darkTheme: ThemeData(
17         brightness: Brightness.dark,
18       ), // ThemeData
19       routes: [
20         "/": (context) => new LoginPage(),
21         "/home": (context) => HomePage(),
22         "/login": (context) => LoginPage()
23       ],
24     ); // MaterialApp
25   }
26 }
```

Login screen

A screenshot of an iPhone 12 Pro Max simulator displaying a white screen with the text "Login Page" centered in blue font.

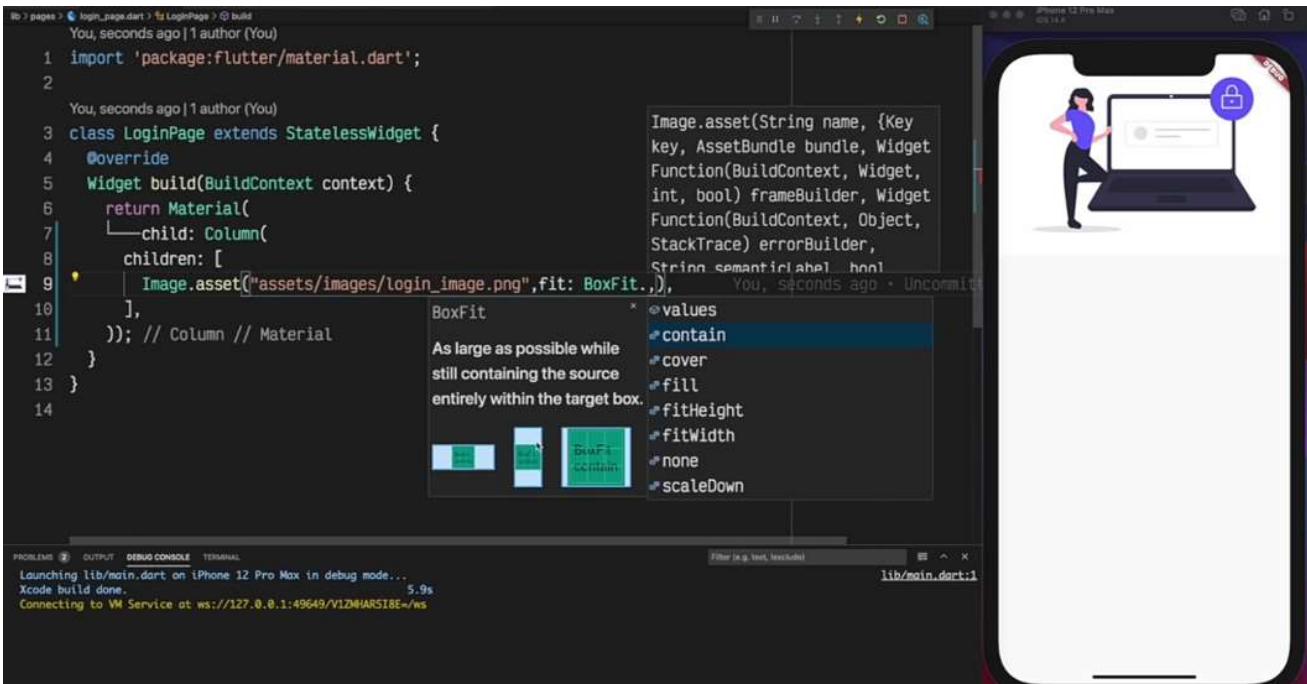
```
lib> pages > LoginPage.dart > LoginPage > build
1 import 'package:flutter/material.dart';
2
3 class LoginPage extends StatelessWidget {
4   @override
5   Widget build(BuildContext context) {
6     return Material(
7       child: Center(
8         child: Text(
9           "Login Page",
10           style: TextStyle(
11             fontSize: 20,
12             color: Colors.blue,
13             fontWeight: FontWeight.bold,
14           ), // TextStyle
15           textScaleFactor: 2.0,
16         ), // Text
17       ), // Center
18     ); // Material
19   }
20 }
```

Undraw



Adding assets

```
pubspec.yaml
1 flutter:
2   # The following line ensures that the Material Icons font is
3   # included with your application, so that you can use the icons in
4   # the material Icons class.
5   uses-material-design: true
6
7   # To add assets to your application, add an assets section, like this:
8   assets:
9     - assets/images/ You, seconds ago • Uncommitted changes
10    # - images/a_dot_ham.jpeg
11
```



Adding fonts

[google_fonts 1.1.2](#)

Published Jan 25, 2021 • material.io

FLUTTER ANDROID iOS WEB

1.61K

Readme Changelog Example Installing Versions Scores

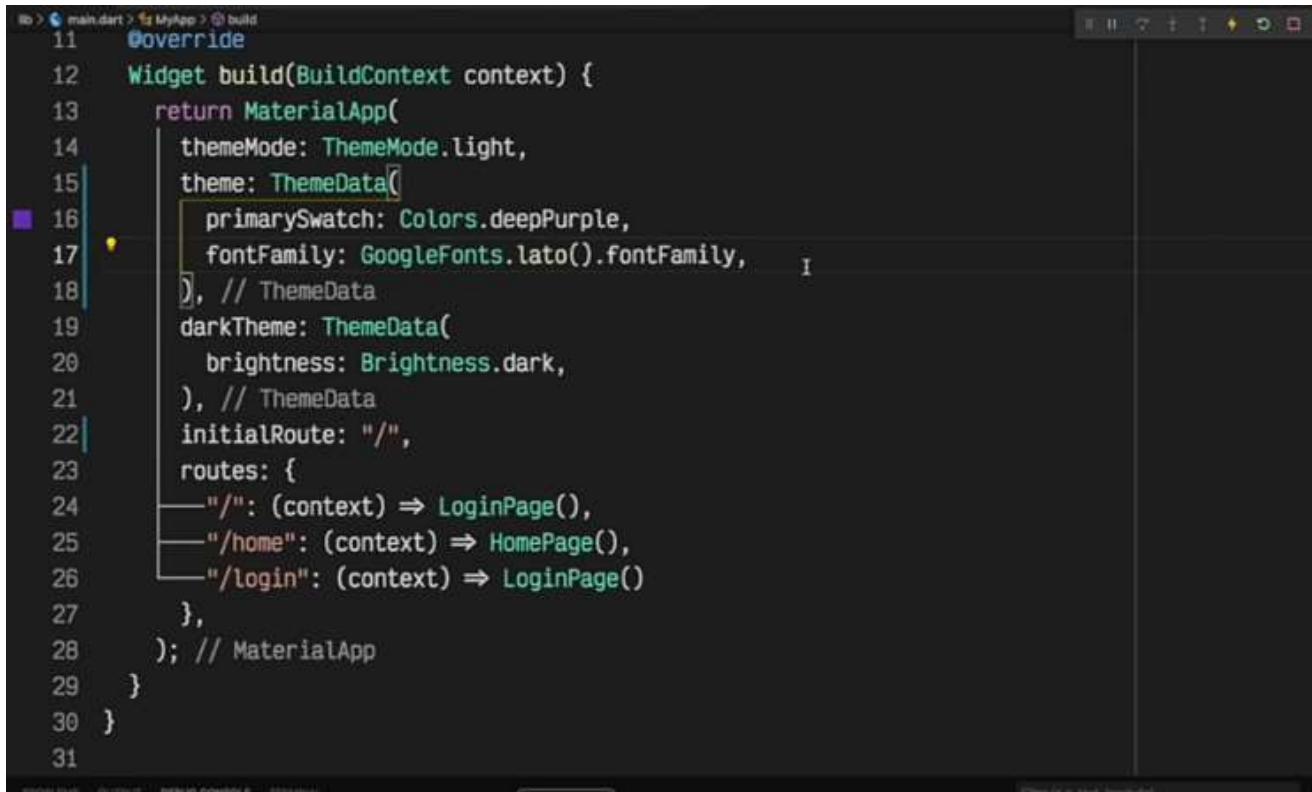
google_fonts

```

pubspec.yaml
17 # https://developer.apple.com/library/archive/documentation/General/Reference/InfoPlistKeyReference/Articles/StandardKeys.html#//apple_ref/doc/uid/TP40009251-SW1
18 version: 1.0.0+1
19
20 environment:
21   sdk: ">=2.7.0 <3.0.0"
22
23 dependencies:
24   flutter:
25     sdk: flutter
26
27   # The following adds the Cupertino Icons font to your application.
28   # Use with the CupertinoIcons class for iOS style icons.
29   cupertino_icons: ^1.0.1
30   google_fonts: ^1.1.2      You, seconds ago | Uncommitted changes
31
32 dev_dependencies:
33   flutter_test:
34     sdk: flutter
35   integration_test:
36     sdk: flutter

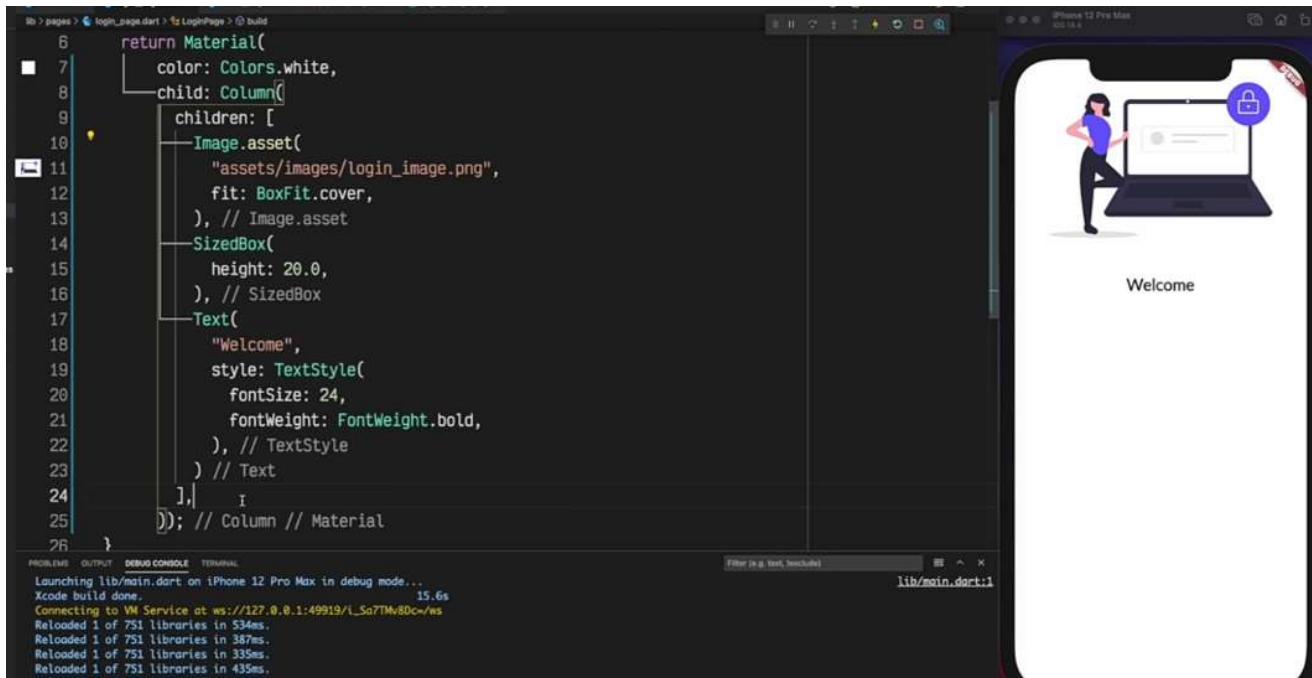
```

Setting fonts family globally to use in app



```
lib/main.dart
11 @override
12 Widget build(BuildContext context) {
13   return MaterialApp(
14     themeMode: ThemeMode.light,
15     theme: ThemeData(
16       primarySwatch: Colors.deepPurple,
17       fontFamily: GoogleFonts.lato().fontFamily,    I
18     ), // ThemeData
19     darkTheme: ThemeData(
20       brightness: Brightness.dark,
21     ), // ThemeData
22     initialRoute: "/",
23     routes: {
24       "/": (context) => LoginPage(),
25       "/home": (context) => HomePage(),
26       "/login": (context) => LoginPage()
27     },
28   ); // MaterialApp
29 }
30 }
31 
```

Designing



```
lib/pages/login_page.dart
6   return Material(
7     color: Colors.white,
8     child: Column(
9       children: [
10         Image.asset(
11           "assets/images/login_image.png",
12           fit: BoxFit.cover,
13         ), // Image.asset
14         SizedBox(
15           height: 20.0,
16         ), // SizedBox
17         Text(
18           "Welcome",
19           style: TextStyle(
20             fontSize: 24,
21             fontWeight: FontWeight.bold,
22           ), // TextStyle
23         ) // Text
24       ],
25     ), // Column // Material
26   }
27 
```

iPhone 12 Pro Max
Welcome

Launching lib/main.dart on iPhone 12 Pro Max in debug mode...
Xcode build done. 15.6s
Connecting to VM Service at ws://127.0.0.1:49919/L_5a7TMv8Dc-/ws
Reloaded 1 of 751 libraries in 534ms.
Reloaded 1 of 751 libraries in 387ms.
Reloaded 1 of 751 libraries in 335ms.
Reloaded 1 of 751 libraries in 435ms.
Reloaded 1 of 751 libraries in 335ms.

Adding inputs

```
lib>pages>login_page.dart>LoginPage>@build
24     SizedBox(
25       height: 20.0,
26     ), // SizedBox
27     TextFormField(
28       decoration: InputDecoration(
29         hintText: "Enter username",
30         labelText: "Username",
31       ), // InputDecoration
32     ), // TextFormField
33     TextFormField(
34       decoration: InputDecoration(
35         hintText: "Enter password",
36         labelText: "Password",
37       ), // InputDecoration
38     ), // TextFormField
39   ],
40 ); // Column // Material
41 }
42 }
43
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

Connecting to VM Service at ws://127.0.0.1:49919/L_So7TMv8Dc-/ws

Reloaded 1 of 751 libraries in 534ms.

Reloaded 1 of 751 libraries in 387ms.

Reloaded 1 of 751 libraries in 335ms.

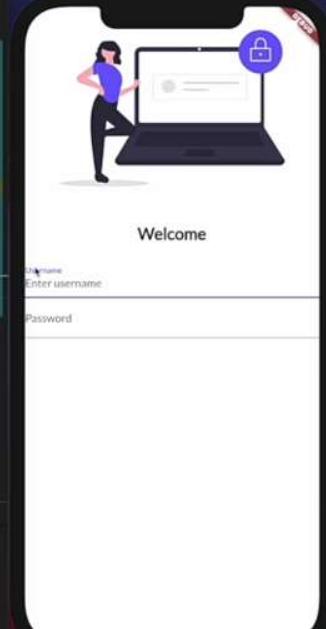
Reloaded 1 of 751 libraries in 435ms.

Reloaded 2 of 751 libraries in 383ms.

Reloaded 2 of 751 libraries in 474ms.

Reloaded 2 of 751 libraries in 371ms.

Reloaded 2 of 751 libraries in 569ms.



Padding to input fields

```
lib>pages>login_page.dart>LoginPage>@build
19       style: TextStyle(
20         fontSize: 24,
21         fontWeight: FontWeight.bold,
22       ), // TextStyle
23     ), // Text
24     SizedBox(
25       height: 20.0,
26     ), // SizedBox
27     Padding(
28       padding:
29         const EdgeInsets.symmetric(vertical: 16.0, horizontal: 32.0),
30       child: Column(
31         children: [
32           TextFormField(
33             decoration: InputDecoration(
34               hintText: "Enter username",
35               labelText: "Username",
36             ), // InputDecoration
37           ), // TextFormField
38           TextFormField(
39             decoration: InputDecoration(
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

Connecting to VM Service at ws://127.0.0.1:49919/L_So7TMv8Dc-/ws

Reloaded 1 of 751 libraries in 534ms.

Reloaded 1 of 751 libraries in 387ms.

Reloaded 1 of 751 libraries in 335ms.

Reloaded 1 of 751 libraries in 435ms.

Reloaded 2 of 751 libraries in 383ms.

Reloaded 2 of 751 libraries in 474ms.

Reloaded 2 of 751 libraries in 371ms.

Reloaded 2 of 751 libraries in 569ms.

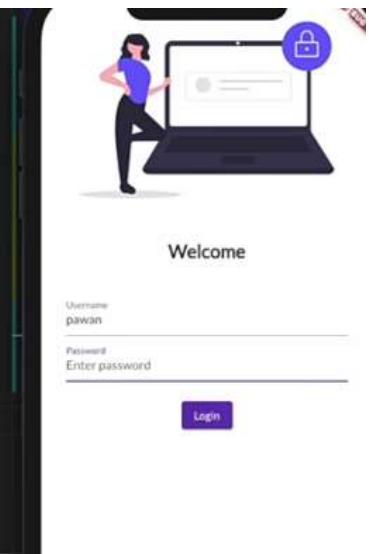


Securing password text

```
lib>pages>login_page.dart>LoginPage>@build
1     TextFormField(
2       obscureText: true,      You, seconds ago + Uncommitted changes
3       decoration: InputDecoration(
4         hintText: "Enter password",
5         labelText: "Password",
6       ), // InputDecoration
7     ), // TextFormField
8   ],
9   ), // Column
10 ) // Padding
11 ];
```



Adding login button

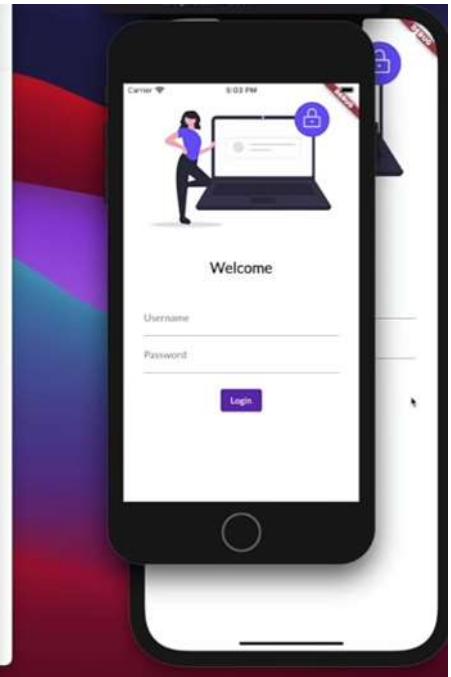


The screenshot shows a Flutter application running in an IDE. On the left, the code for a login page is displayed:

```
1 class LoginPage extends StatelessWidget {
2   @override
3   Widget build(BuildContext context) {
4     return Material(
5       color: Colors.white,
6       child: SingleChildScrollView()
7         child: Column(
8           children: [
9             Image.asset(
10               "assets/images/login_image.png",
11               fit: BoxFit.cover,
12             ),
13             SizedBox(
14               height: 20.0,
15             ),
16             Text(
17               "Welcome",
18               style: TextStyle(
19                 ...),
20             ),
21             TextFormField(
22               obscureText: true,
23               decoration: InputDecoration(
24                 hintText: "Enter password",
25                 labelText: "Password",
26               ),
27             ),
28             ElevatedButton(
29               child: Text("Login"),
30               style: TextButton.styleFrom(),
31               onPressed: () {
32                 print("Hi Codepur");
33               },
34             ),
35           ],
36         ),
37       ),
38     );
39   }
40 }
```

The right side of the screenshot shows the resulting mobile application interface. It features a background image of a person standing next to a laptop with a lock icon. The text "Welcome" is at the top. Below it is a form with a "Username" field containing "pawan" and a "Password" field with placeholder text "Enter password". A purple "Login" button is at the bottom right.

Adding singleChildScrollView to allow scrolling in small devices

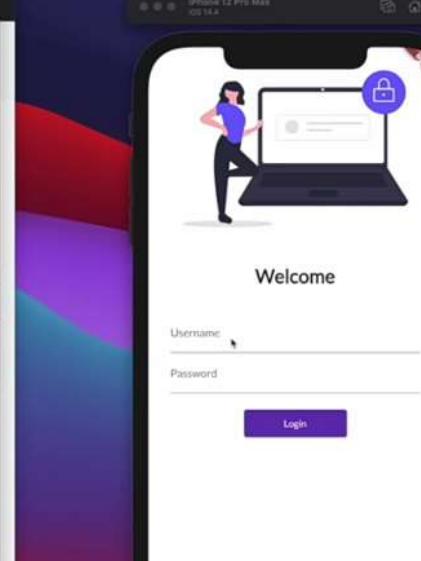


The screenshot shows a Flutter application running in an IDE. On the left, the code for the login page includes a `SingleChildScrollView` widget:

```
1 class LoginPage extends StatelessWidget {
2   @override
3   Widget build(BuildContext context) {
4     return Material(
5       color: Colors.white,
6       child: SingleChildScrollView()
7         child: Column(
8           children: [
9             Image.asset(
10               "assets/images/login_image.png",
11               fit: BoxFit.cover,
12             ),
13             SizedBox(
14               height: 20.0,
15             ),
16             Text(
17               "Welcome",
18               style: TextStyle(
19                 ...),
20             ),
21             TextFormField(
22               obscureText: true,
23               decoration: InputDecoration(
24                 hintText: "Enter password",
25                 labelText: "Password",
26               ),
27             ),
28             ElevatedButton(
29               child: Text("Login"),
30               style: TextButton.styleFrom(),
31               onPressed: () {
32                 print("Hi Codepur");
33               },
34             ),
35           ],
36         ),
37       ),
38     );
39   }
40 }
```

The right side of the screenshot shows the mobile application on a smartphone. The screen displays the same "Welcome" message and form fields as before, but the entire content is scrollable, allowing users to see the full form even if the device screen is smaller than the content area.

Giving min size to login

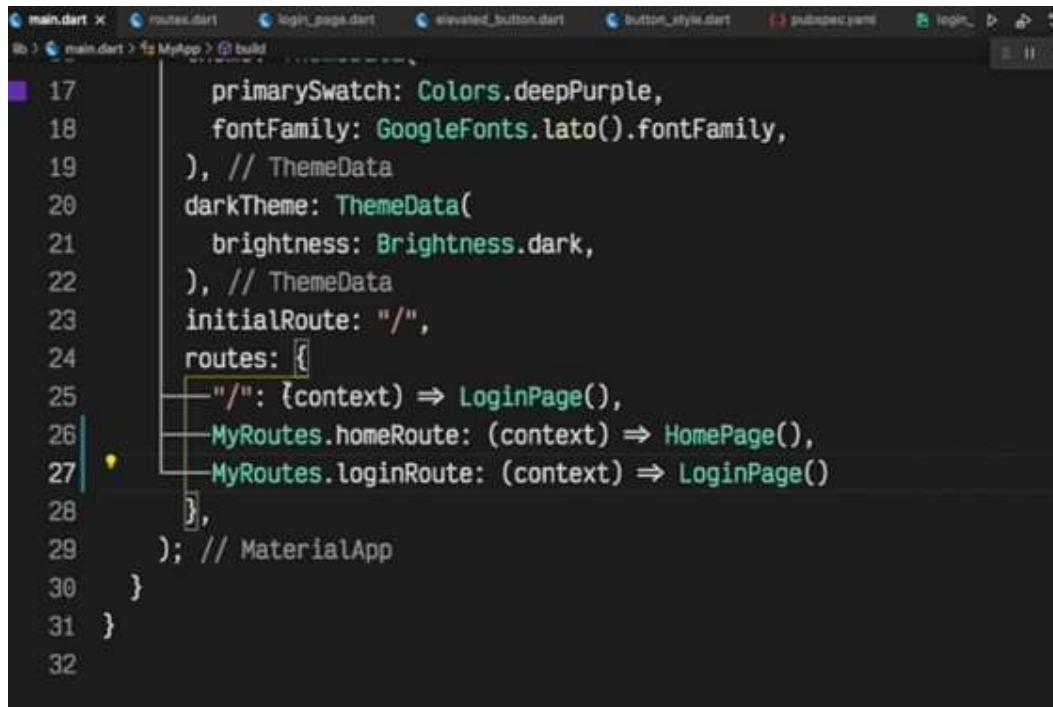


```
41     decoration: InputDecoration(42         hintText: "Enter password",43         labelText: "Password",44     ), // InputDecoration45     ), // TextFormField46     SizedBox(47         height: 20.0,48     ), // SizedBox49     ElevatedButton(50         child: Text("Login"),51         style: TextButton.styleFrom(minimumSize: Size(150, 40)),52         onPressed: () {53             print("Hi Codepur");54         },55     ), // ElevatedButton56     ],57 ), // Column58 ), // Padding59 ],60 ), // Column61 ); // SingleChildScrollView // Material
```

Handling routes from class

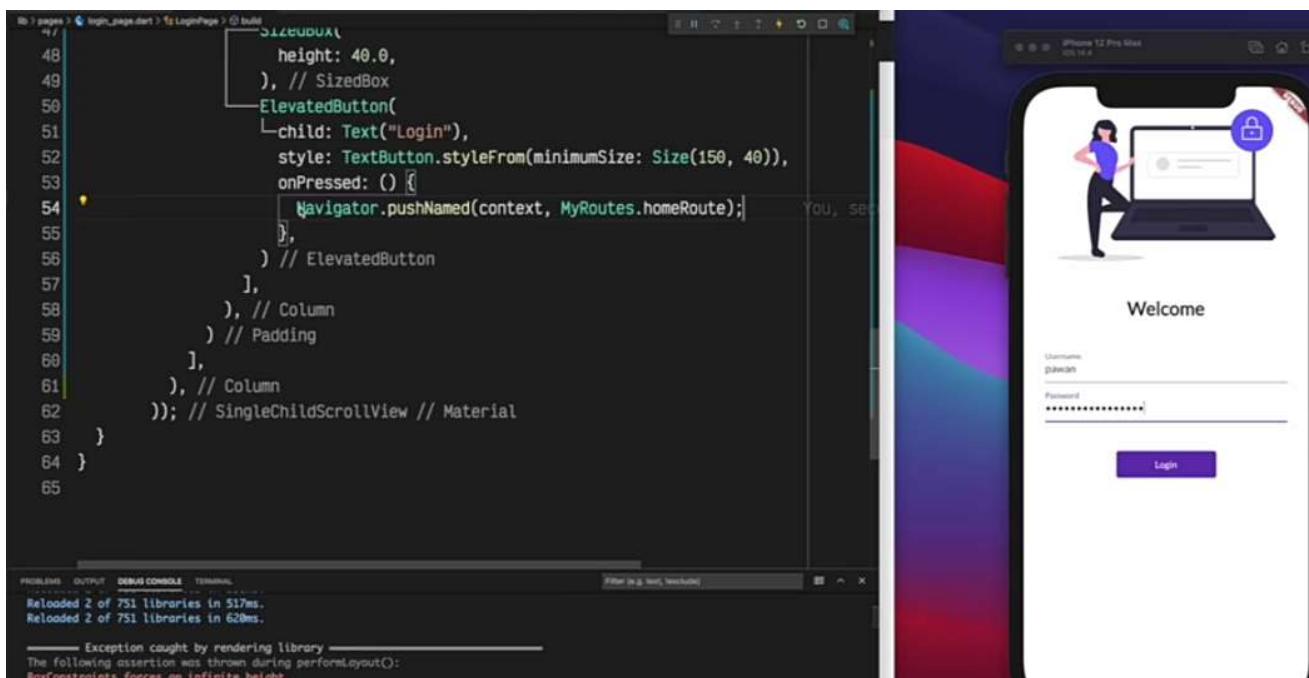
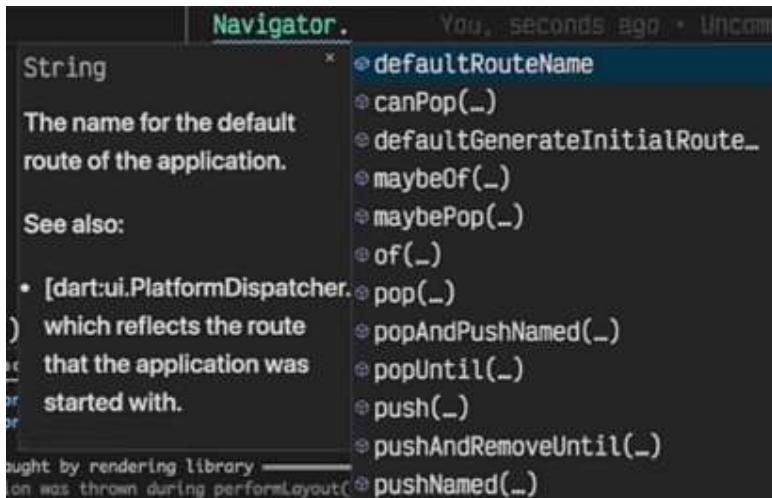


```
lib> villa> routes.dart> MyRoutes> homeRoute1 class MyRoutes {2     static String loginRoute = "/login";3     static String HomeRoute = "/home";4 }5
```



```
main.dart X routes.dart login_page.dart elevated_button.dart button_style.dart subspecify... login_> build17     primarySwatch: Colors.deepPurple,18     fontFamily: GoogleFonts.lato().fontFamily,19     ), // ThemeData20     darkTheme: ThemeData(21         brightness: Brightness.dark,22     ), // ThemeData23     initialRoute: "/",24     routes: [25         "/": (context) => LoginPage(),26         MyRoutes.homeRoute: (context) => HomePage(),27         MyRoutes.loginRoute: (context) => LoginPage()28     ],29     ); // MaterialApp30 }31 }32 }
```

- Navigation to different screen

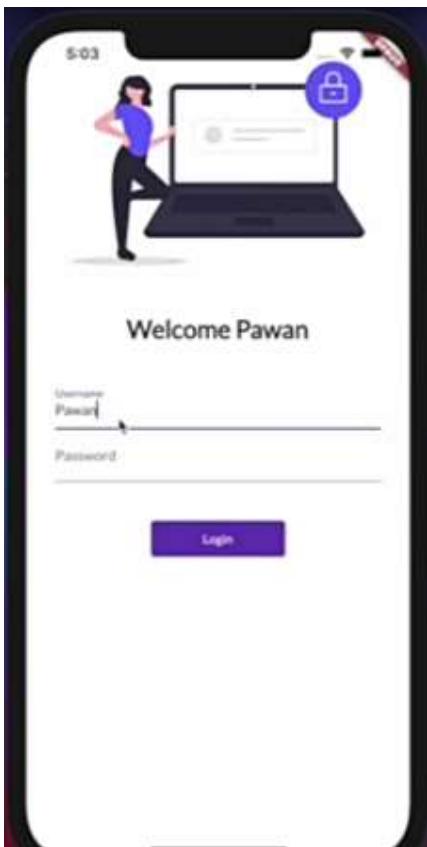


Login styling

```
main.dart routes.dart login_page.dart <_LoginPageState.dart button_style.dart pubspec.yaml login_image.png
lib> pages > login_page.dart > _LoginPageState.dart <_LoginPageState.dart
  You, seconds ago | 1 author (You)
3
  You, seconds ago | 1 author (You)
4  class LoginPage extends StatefulWidget {
5    @override
6    _LoginPageState createState() => _LoginPageState();
7  }
8
  You, seconds ago | 1 author (You)
9  class _LoginPageState extends State<LoginPage> {
10  String name = ""; You, seconds ago + Uncommitted changes
11
12  @override
13  Widget build(BuildContext context) {
14    return Material(
15      color: Colors.white,
16      child: SingleChildScrollView(
17        child: Column(
18          children: [
19            Image.asset(
20              "assets/images/login_image.png",
21              fit: BoxFit.cover,
22            ), // Image.asset
23            SizedBox(
24              height: 20.0,
25            ), // SizedBox
26            Text(
27              "Welcome",
28              style: TextStyle(
29                color: Colors.white,
30                fontSize: 24,
31                fontWeight: FontWeight.bold,
32              ), // TextStyle
33            ),
34            SizedBox(
35              height: 20.0,
36            ), // SizedBox
37            Padding(
38              padding: const EdgeInsets.symmetric(
39                vertical: 16.0, horizontal: 32.0), // EdgeInsets.symmetric
40              child: Column(
41                children: [
42                  TextFormField(
43                    decoration: InputDecoration(
44                      hintText: "Email or Phone Number",
45                      hintStyle: TextStyle(
46                        color: Colors.grey,
47                      ), // TextStyle
48                    ),
49                  ),
50                  SizedBox(
51                    height: 12.0,
52                  ), // SizedBox
53                  TextFormField(
54                    decoration: InputDecoration(
55                      hintText: "Password",
56                      hintStyle: TextStyle(
57                        color: Colors.grey,
58                      ), // TextStyle
59                    ),
60                  ),
61                  SizedBox(
62                    height: 12.0,
63                  ), // SizedBox
64                  Container(
65                    width: double.infinity,
66                    height: 45.0,
67                    child: TextButton(
68                      onPressed: () {
69                        Navigator.pushNamed(context, "/home");
70                      },
71                      style: TextButton.styleFrom(
72                        backgroundColor: Colors.purple,
73                        shape: RoundedRectangleBorder(
74                          borderRadius: BorderRadius.circular(10.0),
75                        ),
76                        padding: EdgeInsets.all(10.0),
77                      ),
78                      child: Text(
79                        "Log In",
80                        style: TextStyle(
81                          color: Colors.white,
82                          fontWeight: FontWeight.bold,
83                        ), // TextStyle
84                      ),
85                    ),
86                  ),
87                ],
88              ),
89            ),
90          ],
91        ),
92      ),
93    );
94  }
95}
```

```
39     child: Column(
40       children: [
41         TextFormField(
42           decoration: InputDecoration(
43             hintText: "Enter username",
44             labelText: "Username",
45           ), // InputDecoration
46           onChanged: (value) {
47             name = value;
48             setState(() {
49               You, seconds ago • Uncommitted changes
50             });
51           },
52         ), // TextFormField
53         TextFormField(
54           obscureText: true,
55           decoration: InputDecoration(
56             hintText: "Enter password",
57             labelText: "Password",
58           ), // InputDecoration
59         ), // TextFormField
```

Output



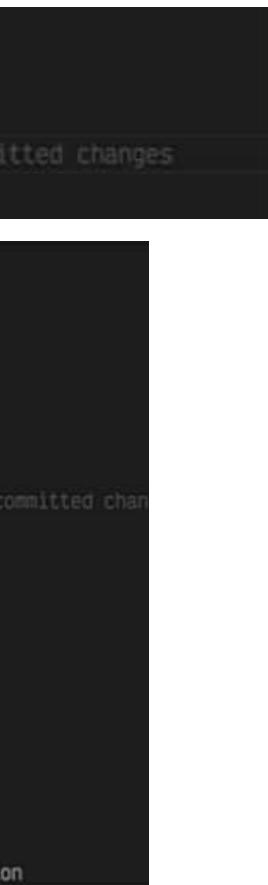
Styling login button



```
InkWell(
  onTap: () {
    Navigator.pushNamed(context, MyRoutes.homeRoute);
  },
  child: Container(
    width: 150,
    height: 50,
    alignment: Alignment.center,
    child: Text(
      "Login",
      style: TextStyle(
        color: Colors.white,
        fontWeight: FontWeight.bold,
        fontSize: 18), // TextStyle
    ), // Text
    decoration: BoxDecoration(

```

Styling the button when pressed



```
You, seconds ago | 1 author (You)
9 class _LoginPageState extends State<LoginPage> {
10   String name = "";
11   bool changeButton = false;      You, seconds ago • Uncommitted changes
12 }
```

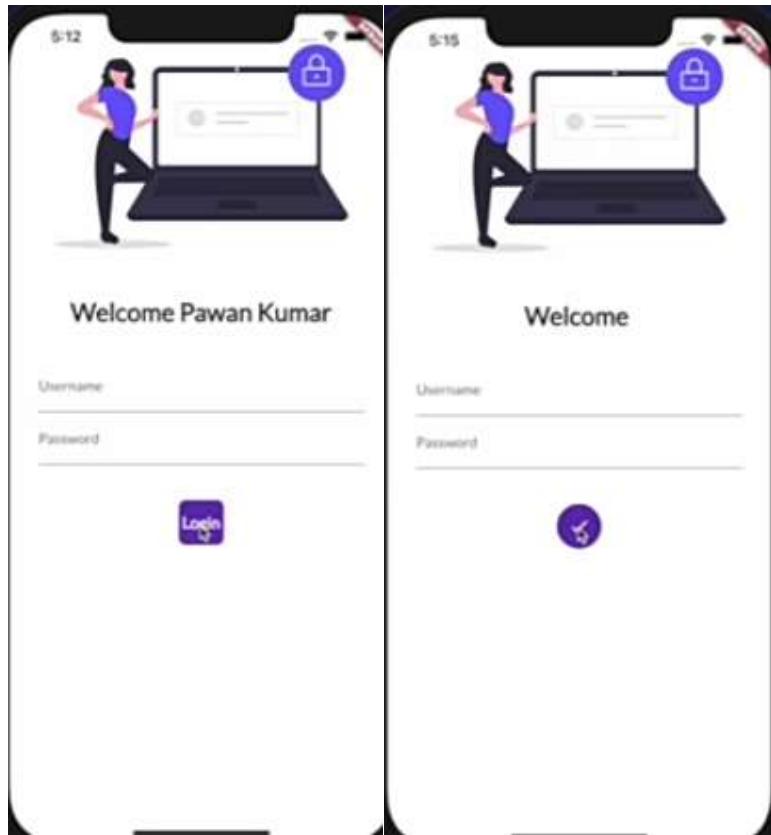
```

  onTap: () {
    setState(() {
      changeButton = true;
    });
    // Navigator.pushNamed(context, MyRoutes.homeRoute);
  },
  child: AnimatedContainer(
    duration: Duration(seconds: 1),
    width: changeButton ? 50 : 150,      You, seconds ago • Uncommitted chan
    height: 50,
    alignment: Alignment.center,
    child: Text(
      "Login",
      style: TextStyle(
        color: Colors.white,
        fontWeight: FontWeight.bold,
        fontSize: 18), // TextStyle
    ), // Text
    decoration: BoxDecoration(
      color: Colors.deepPurple,
      borderRadius: BorderRadius.circular(8)), // BoxDecoration

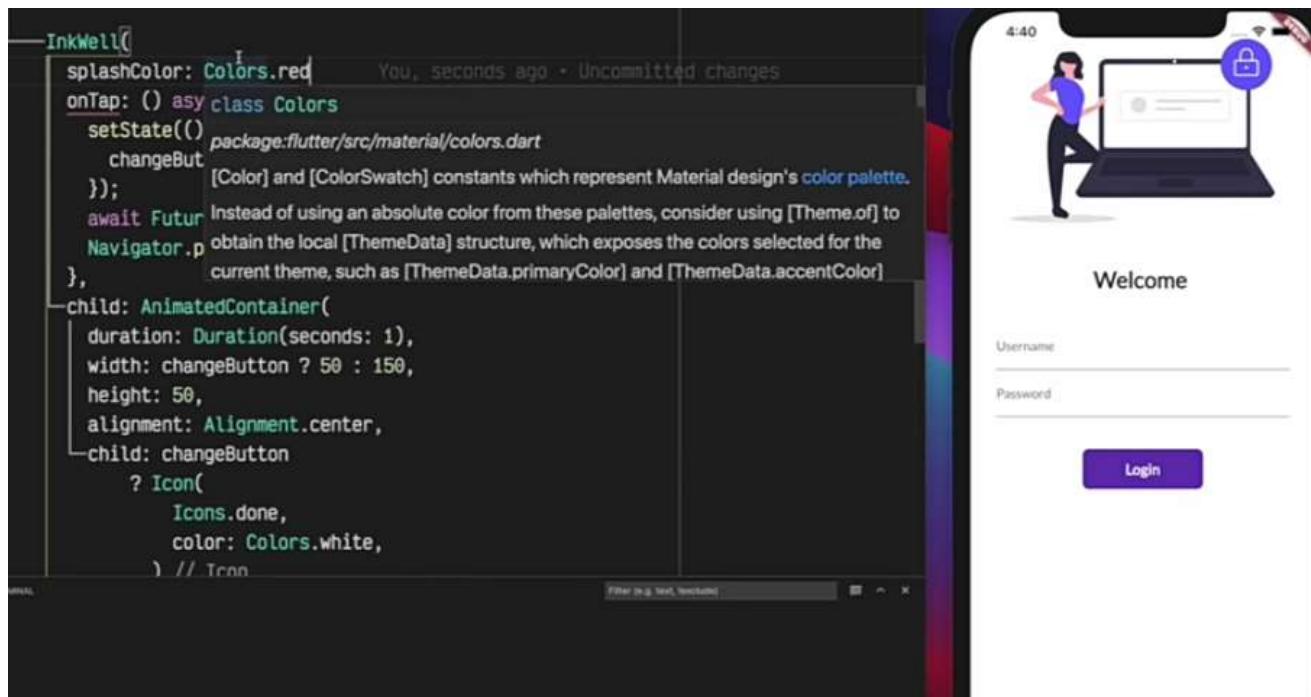
```

```
└ child: AnimatedContainer(
  duration: Duration(seconds: 1),
  width: changeButton ? 50 : 50,
  height: 50,
  alignment: Alignment.center,
  child: changeButton
    ? Icon(
        Icons.done,
        color: Colors.white,
      ) // Icon
    : Text(
        "Login",
        style: TextStyle(
          color: Colors.white,
          fontWeight: FontWeight.bold,
          fontSize: 18), // TextStyle
      ), // Text
  decoration: BoxDecoration(
    color: Colors.deepPurple,
    borderRadius:
      BorderRadius.circular(changeButton ? 50 : 8),
  ),
)
```

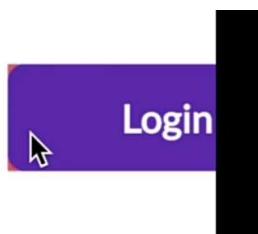
Output



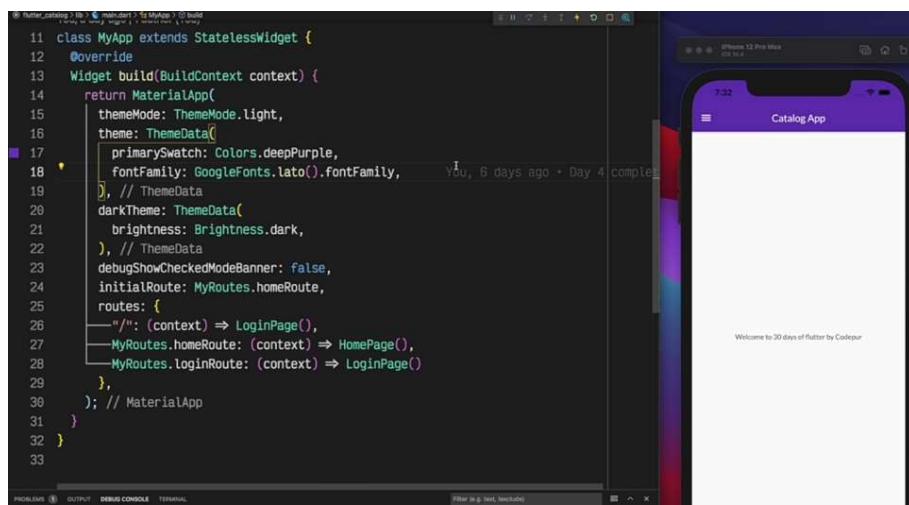
Adding ripple or splash effect



Output



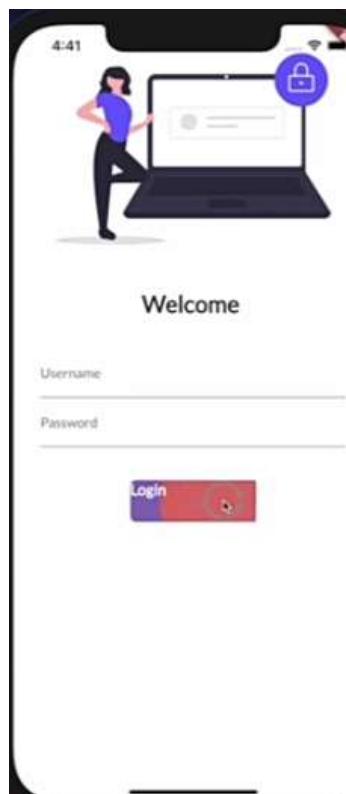
Hiding debug banner



To make splash on front side using Ink instead of AnimatedContainer

```
InkWell(  
    splashColor: Colors.red,  
    onTap: () async {  
        setState(() {  
            changeButton = true;  
        });  
        await Future.delayed(Duration(seconds: 1));  
        Navigator.pushNamed(context, MyRoutes.homeRoute);  
    },  
    child: Ink(  
        // duration: Duration(seconds: 1),  
        width: changeButton ? 50 : 150,  
        height: 50,  
        // alignment: Alignment.center, You, seconds ago + Uncommit  
        child: changeButton  
            ? Icon(  
                Icons.done,  
                color: Colors.white,  
            ) // Icon  
            : Text(  
                You, seconds ago + Uncommit  
            )  
    )  
)
```

Output



Using Material instead of inkwell for ripple effect inside animated



Handling validation for input with UI/UX using FormState

```
class _LoginPageState extends State<LoginPage> {
  String name = "";
  bool changeButton = false;

  final _formKey = GlobalKey<FormState>();

  @override
  Widget build(BuildContext context) {
    return Material(
```

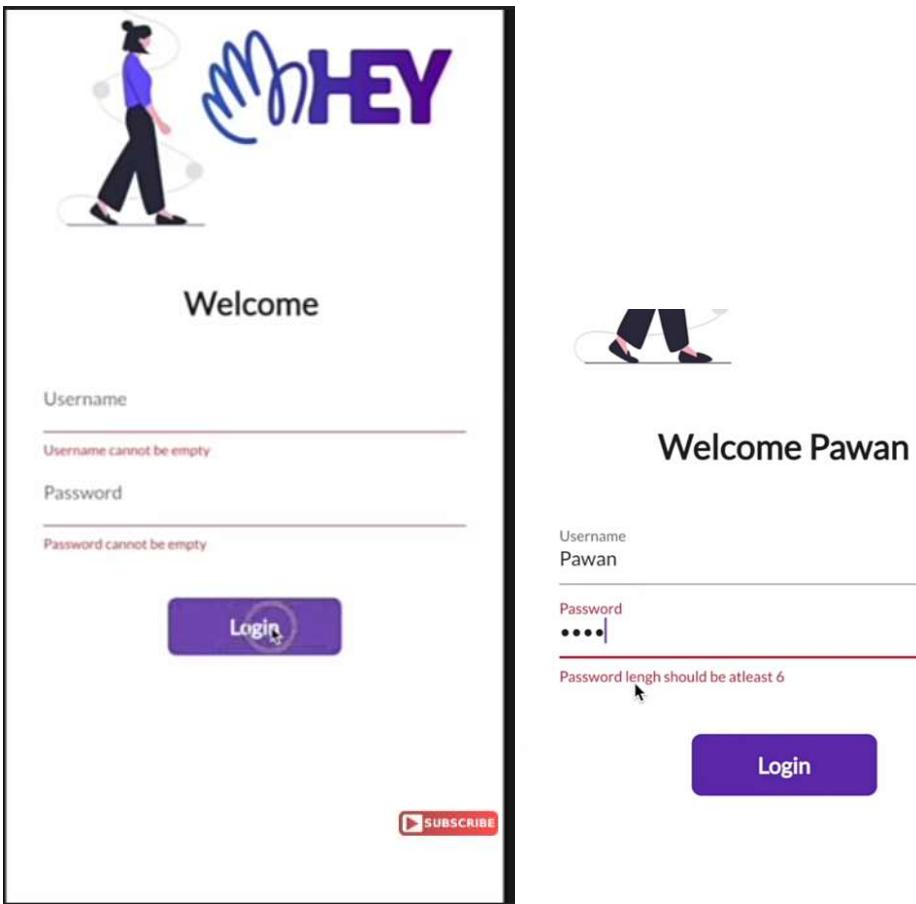
```
Padding(
  padding: const EdgeInsets.symmetric(
    vertical: 16.0, horizontal: 32.0), // EdgeInsets.symmetric
  child: Column(
    children: [
      TextFormField(
        decoration: InputDecoration(
          hintText: "Enter username",
          labelText: "Username",
        ), // InputDecoration
        validator: (value) {
          if (value.isEmpty) {
            return "Username cannot be empty";
          }
          return null;
        }, // Validator
        onChanged: (value) {
          name = value;
          setState(() {});
        },
      ), // TextFormField
    ],
  ),
),
```

```
    TextFormField(
      obscureText: true,
      decoration: InputDecoration(
        hintText: "Enter password",
        labelText: "Password",
      ), // InputDecoration
      validator: (value) {
        if (value.isEmpty) {
          return "Password cannot be empty";
        } else if (value.length < 6) {
          return "Password length should be atleast 6";
        }
        return null;
      },
    ), // TextFormField
    SizedBox(
      height: 40.0,
    ), // SizedBox
    Material(
      color: Colors.deepPurple,
      borderRadius:
        BorderRadius.circular(changeButton ? 50 : 8),
```

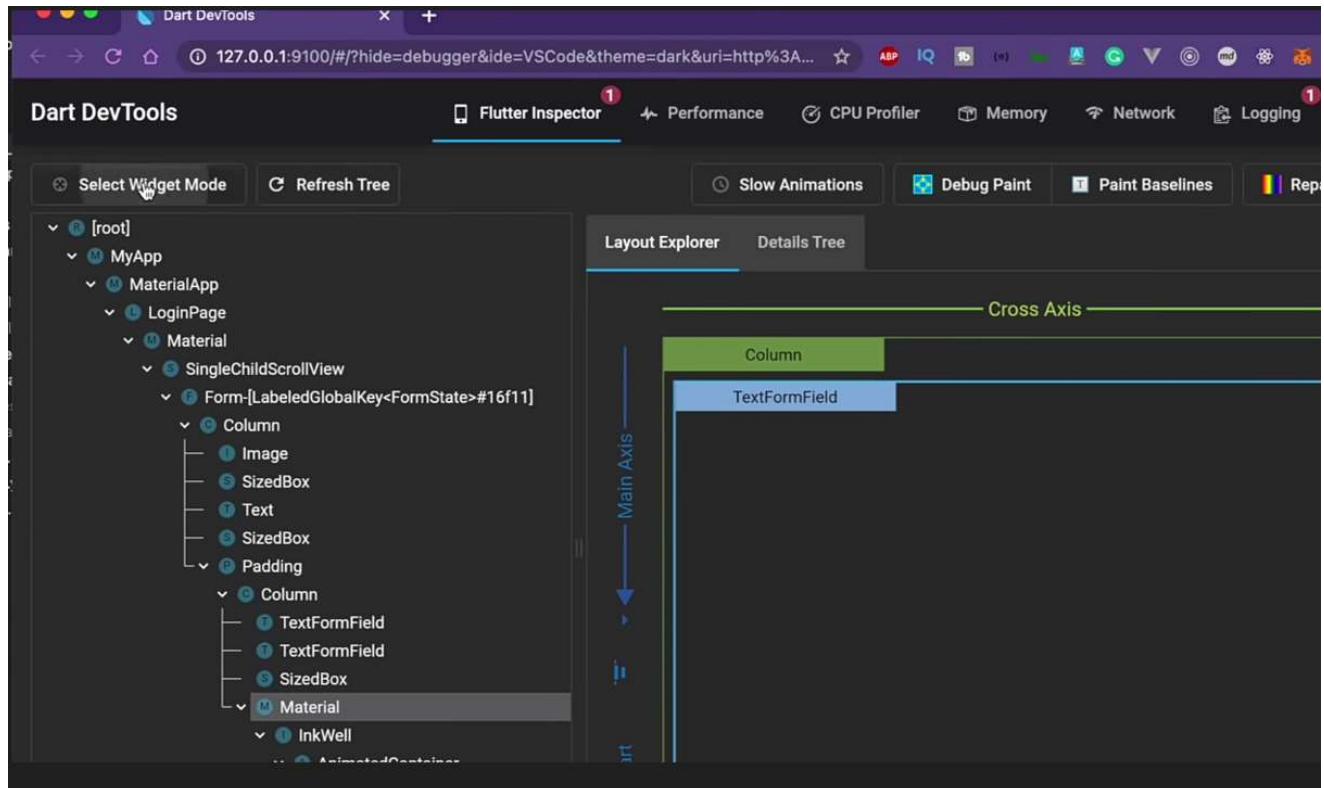
And

```
lib> pages > login_page.dart > _LoginPageState > moveToHome
11   bool changeButton = false;
12
13   final _formKey = GlobalKey<FormState>();
14
15   moveToHome(BuildContext context) async {
16     if (_formKey.currentState.validate()) {
17       setState(() {
18         changeButton = true;
19       });
20       await Future.delayed(Duration(seconds: 1));
21       await Navigator.pushNamed(context, MyRoutes.homeRoute);
22       setState(() {
23         changeButton = false;
24       });
25     }
26   }
27
28   @override
29   Widget build(BuildContext context) {
30     return Material(
```

Output



Flutter debugger



• Designing drawer

The screenshot shows an IDE interface with two main sections. On the left, the code for `HomePage` is displayed:

```
You, seconds ago | 1 author (You)
2
3 class HomePage extends StatelessWidget {
4   final int days = 30;
5   final String name = "Codepur";
6   @override
7   Widget build(BuildContext context) {
8     return Scaffold(
9       appBar: AppBar(
10         title: Text("Catalog App"),
11       ), // AppBar
12       body: Center(
13         child: Container(
14           child: Text("Welcome to $days days of flutter by $name"),
15         ), // Container
16       ), // Center
17       drawer: MyDrawer(), You, seconds ago • Uncommitted changes
18     ); // Scaffold
19   }
20 }
```

A code completion dropdown is open at the line `drawer: MyDrawer()`, showing options like `Import library 'package:flutter_catalog/widgets/drawer.dart'`, `Create class 'MyDrawer'`, `Create method 'MyDrawer'`, `Create function 'MyDrawer'`, and `Extract Local Variable`.

On the right, the mobile application preview shows a purple header bar with the text "Catalog App". Below it, the main content area displays the text "Welcome to 30 days of flutter by Codepur".

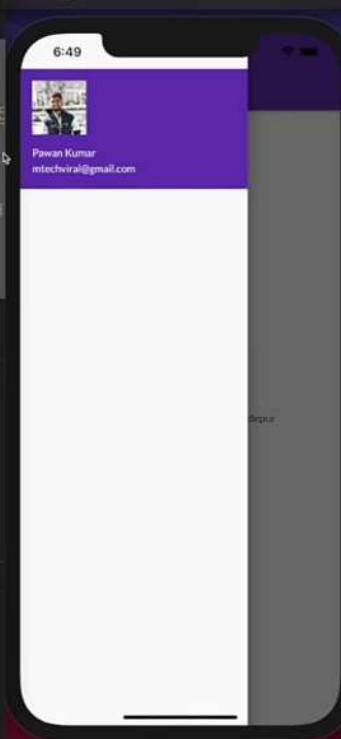
And

The screenshot shows an IDE interface with two main sections. On the left, the code for `MyDrawer` is displayed:

```
flutter_catalog > lib > widgets > drawer.dart > MyDrawer > build
1 import 'package:flutter/material.dart';
2
3 class MyDrawer extends StatelessWidget {
4   @override
5   Widget build(BuildContext context) {
6     return Drawer(
7       child: ListView(
8         children: [
9           DrawerHeader(
10             padding: EdgeInsets.zero,
11             child: UserAccountsDrawerHeader(
12               margin: EdgeInsets.zero,
13               accountName: Text("Pawan Kumar"),
14               accountEmail: Text("mtechviral@gmail.com"),
15             ), // UserAccountsDrawerHeader
16           ), // DrawerHeader
17         ],
18       ), // ListView
19     ); // Drawer
20   }
21 }
```

On the right, the mobile application preview shows a purple header bar with the text "6:47". Below it, the main content area displays the user information "Pawan Kumar" and "mtechviral@gmail.com".

Profile image



```
class MyDrawer extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    final imageUrl =
        "https://avatars.githubusercontent.com/u/12619420?s=460&u=26db98cbde1dd34c7c67b85c24050";
    return Drawer(
      child: ListView(
        children: [
          DrawerHeader(
            padding: EdgeInsets.zero,
            child: UserAccountsDrawerHeader(
              margin: EdgeInsets.zero,
              accountName: Text("Pawan Kumar"),
              accountEmail: Text("mtechviral@gmail.com"),
              currentAccountPicture: Image.network(imageUrl),
            ), // UserAccountsDrawerHeader
          ), // DrawerHeader
        ],
      ), // ListView
    ); // Drawer
  }
}
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

Reloaded 1 of 752 libraries in 528ms.
Reloaded 0 libraries in 285ms.
Reloaded 3 of 753 libraries in 415ms.
Reloaded 3 of 753 libraries in 513ms.
Reloaded 3 of 753 libraries in 675ms.
Reloaded 3 of 753 libraries in 658ms.
Reloaded 3 of 753 libraries in 605ms.
Reloaded 3 of 753 libraries in 445ms.
Reloaded 3 of 753 libraries in 564ms.
Reloaded 3 of 753 libraries in 510ms.
Reloaded 3 of 753 libraries in 610ms.

circularAvatar

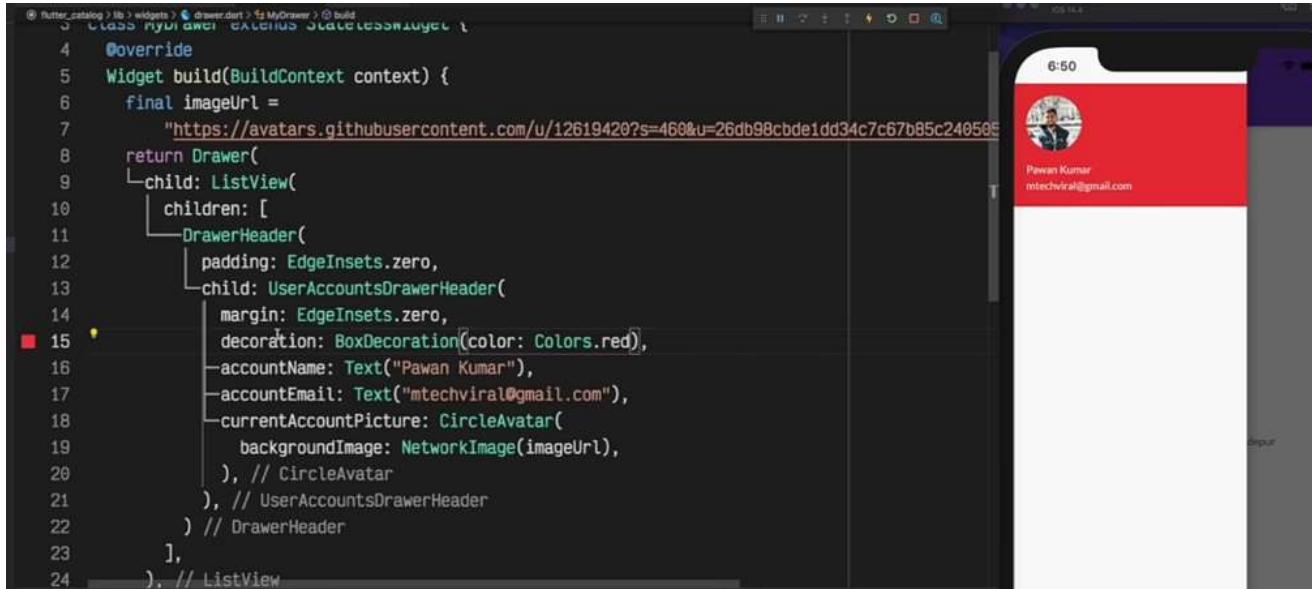


```
class MyDrawer extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    final imageUrl =
        "https://avatars.githubusercontent.com/u/12619420?s=460&u=26db98cbde1dd34c7c67b85c24050";
    return Drawer(
      child: ListView(
        children: [
          DrawerHeader(
            padding: EdgeInsets.zero,
            child: UserAccountsDrawerHeader(
              margin: EdgeInsets.zero,
              accountName: Text("Pawan Kumar"),
              accountEmail: Text("mtechviral@gmail.com"),
              currentAccountPicture: CircleAvatar(
                backgroundImage: NetworkImage(imageUrl),
              ), // CircleAvatar
            ), // UserAccountsDrawerHeader
          ), // DrawerHeader
        ],
      ), // ListView
    ); // Drawer
  }
}
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

Reloaded 1 of 752 libraries in 528ms.
Reloaded 0 libraries in 285ms.
Reloaded 3 of 753 libraries in 415ms.
Reloaded 3 of 753 libraries in 513ms.
Reloaded 3 of 753 libraries in 675ms.
Reloaded 3 of 753 libraries in 658ms.
Reloaded 3 of 753 libraries in 605ms.
Reloaded 3 of 753 libraries in 445ms.
Reloaded 3 of 753 libraries in 564ms.
Reloaded 3 of 753 libraries in 510ms.
Reloaded 3 of 753 libraries in 610ms.

Changing color of drawerheader

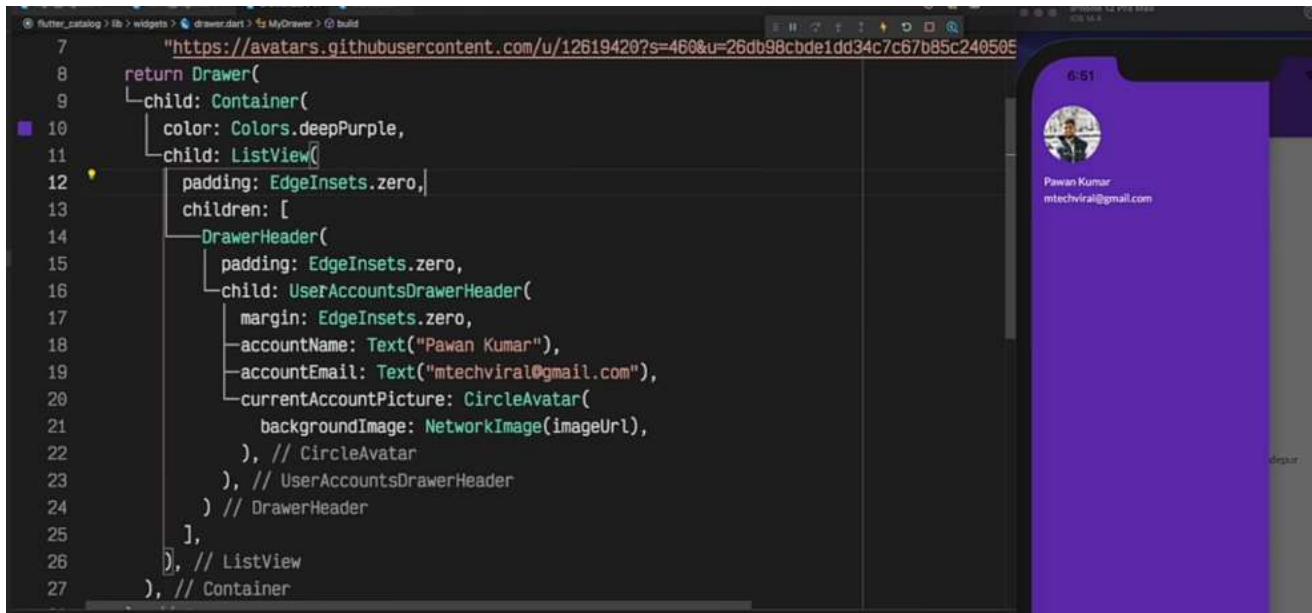


A screenshot of a Flutter development environment. On the left, the code for a drawer header is displayed:

```
4  @override
5  Widget build(BuildContext context) {
6    final imageUrl =
7      "https://avatars.githubusercontent.com/u/12619420?s=460&u=26db98cbde1dd34c7c67b85c240505";
8    return Drawer(
9      child: ListView(
10        children: [
11          DrawerHeader(
12            padding: EdgeInsets.zero,
13            child: UserAccountsDrawerHeader(
14              margin: EdgeInsets.zero,
15              decoration: BoxDecoration(color: Colors.red),
16              accountName: Text("Pawan Kumar"),
17              accountEmail: Text("mtechviral@gmail.com"),
18              currentAccountPicture: CircleAvatar(
19                backgroundImage: NetworkImage(imageUrl),
20              ), // CircleAvatar
21            ), // UserAccountsDrawerHeader
22          ), // DrawerHeader
23        ],
24      ). // ListView
25    ), // Container
26  ). // Drawer
27}
```

The right side of the screen shows a mobile application running on an iPhone X simulator. The top navigation bar is red, featuring a circular profile picture, the name "Pawan Kumar", and the email "mtechviral@gmail.com". The time on the phone is 6:50.

Giving color to whole drawer

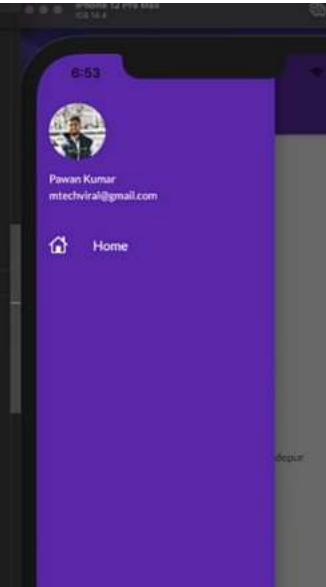


A screenshot of a Flutter development environment. On the left, the code for a drawer with a purple background is displayed:

```
7      "https://avatars.githubusercontent.com/u/12619420?s=460&u=26db98cbde1dd34c7c67b85c240505";
8    return Drawer(
9      child: Container(
10        color: Colors.deepPurple,
11        child: ListView(
12          padding: EdgeInsets.zero,
13          children: [
14            DrawerHeader(
15              padding: EdgeInsets.zero,
16              child: UserAccountsDrawerHeader(
17                margin: EdgeInsets.zero,
18                accountName: Text("Pawan Kumar"),
19                accountEmail: Text("mtechviral@gmail.com"),
20                currentAccountPicture: CircleAvatar(
21                  backgroundImage: NetworkImage(imageUrl),
22                ), // CircleAvatar
23              ), // UserAccountsDrawerHeader
24            ), // DrawerHeader
25          ],
26        ). // ListView
27      ). // Container
28    ). // Drawer
29}
```

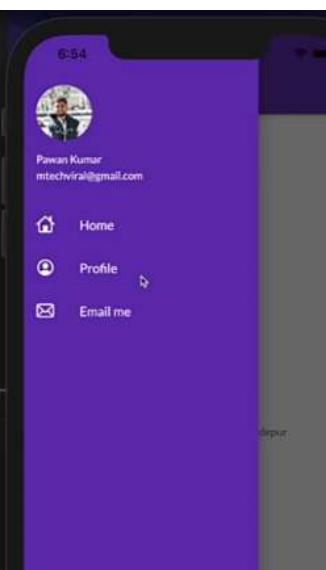
The right side of the screen shows a mobile application running on an iPhone X simulator. The entire drawer has a solid purple background. The top navigation bar is white, featuring a circular profile picture, the name "Pawan Kumar", and the email "mtechviral@gmail.com". The time on the phone is 6:51.

Adding items of sidebar



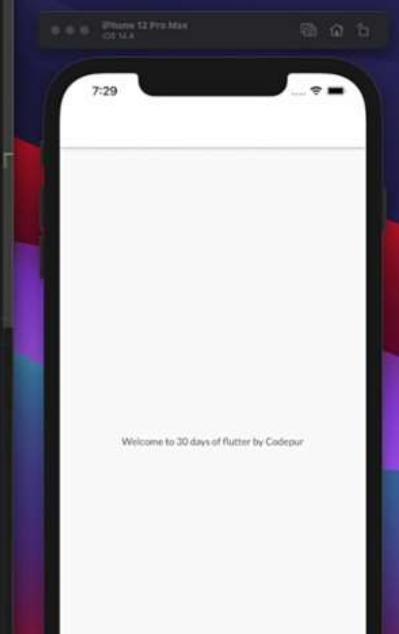
```
23     ), // CircleAvatar
24   ), // UserAccountsDrawerHeader
25   ), // DrawerHeader
26   ListTile(
27     leading: Icon(
28       CupertinoIcons.home,
29       color: Colors.white,
30     ), // Icon
31     title: Text(
32       "Home",
33       textScaleFactor: 1.2,
34       style: TextStyle(
35         color: Colors.white,
36       ), // TextStyle
37     ), // Text
38   ) // ListTile
39 ],
40 ), // ListView
41 ), // Container
42 ); // Drawer
43 }
```

More items



```
48   ), // TextStyle
49   ), // Text
50   ), // ListTile
51   ListTile(
52     leading: Icon(
53       CupertinoIcons.mail,
54       color: Colors.white,
55     ), // Icon
56     title: Text(
57       "Email me",
58       textScaleFactor: 1.2,
59       style: TextStyle(
60         color: Colors.white,
61       ), // TextStyle
62     ), // Text
63   ) // ListTile
64 ],
65 ],
66 ), // ListView
67 ), // Container
68 ); // Drawer
```

Designing header of home



```
You, seconds ago | 1 author (You)
1 import 'package:flutter/material.dart';
2 import 'package:flutter_catalog/widgets/drawer.dart';
3
4 You, seconds ago | 1 author (You)
5 class HomePage extends StatelessWidget {
6   final int days = 30;
7   final String name = "Codepur";
8   @override
9   Widget build(BuildContext context) {
10     return Scaffold(
11       appBar: AppBar(
12         backgroundColor: Colors.white,
13         title: Text("Catalog App"),
14       ),
15       body: Center(
16         child: Container(
17           child: Text("Welcome to $days days of flutter by $name"),
18         ),
19       ),
20       drawer: MyDrawer(),
21     );
22   }
23 }
```

Elevation 0



```
home_page.dart X main.dart drawer.dart catalog.json
flutter_catalog lib pages home_page.dart > HomePage > build
You, seconds ago | 1 author (You)
1 import 'package:flutter/material.dart';
2 import 'package:flutter_catalog/widgets/drawer.dart';
3
4 You, seconds ago | 1 author (You)
5 class HomePage extends StatelessWidget {
6   final int days = 30;
7   final String name = "Codepur";
8   @override
9   Widget build(BuildContext context) {
10     return Scaffold(
11       appBar: AppBar(
12         backgroundColor: Colors.white,
13         elevation: 0.0,
14         iconTheme: IconThemeData(color: Colors.black),
15         title: Text(
16           "Catalog App",
17           style: TextStyle(color: Colors.black),
18         ),
19       ),
20     );
21   }
22 }
```

```
11 class MyApp extends StatelessWidget {
12   @override
13   Widget build(BuildContext context) {
14     return MaterialApp(
15       themeMode: ThemeMode.light,
16       theme: ThemeData(
17         primarySwatch: Colors.deepPurple,
18         fontFamily: GoogleFonts.lato().fontFamily,
19         appBarTheme: AppBarTheme(
20           color: Colors.white,
21           elevation: 0.0,
22           iconTheme: IconThemeData(color: Colors.black),
23           textTheme: Theme.of(context).textTheme,
24         ), // AppBarTheme // ThemeData
25         darkTheme: ThemeData([
26           brightness: Brightness.dark,
27         ], // ThemeData
28         debugShowCheckedModeBanner: false,
29         initialRoute: MyRoutes.homeRoute,
30         routes: {
31           "/": (context) => LoginPage(),
32           MyRoutes.homeRoute: (context) => HomePage(),
33           MyRoutes.loginRoute: (context) => LoginPage()
34         },
35       );
36     }
37   }
38 }
```



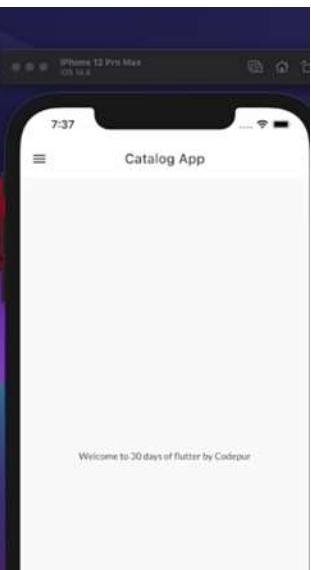
Making separate class for ThemeData

```
1 import 'package:flutter/material.dart';
2 import 'package:google_fonts/google_fonts.dart';
3
4 class MyTheme {
5   static ThemeData lightTheme(BuildContext context) => ThemeData(
6     primarySwatch: Colors.deepPurple,
7     fontFamily: GoogleFonts.lato().fontFamily,
8     appBarTheme: AppBarTheme(
9       color: Colors.white,
10      elevation: 0.0,
11      iconTheme: IconThemeData(color: Colors.black),
12      textTheme: Theme.of(context).textTheme,
13    ); // AppBarTheme // ThemeData
14
15   static ThemeData darkTheme(BuildContext context) => ThemeData([
16     brightness: Brightness.dark,
17   ],
18 );
19 }
```



And

```
1 You, seconds ago | 1 author (You)
2 class MyApp extends StatelessWidget {
3   @override
4   Widget build(BuildContext context) {
5     return MaterialApp(
6       themeMode: ThemeMode.light,
7       theme: MyTheme.lightTheme(context), // You, seconds ago + Uncommitted changes
8       darkTheme: ThemeData(
9         brightness: Brightness.dark,
10       ), // ThemeData
11       debugShowCheckedModeBanner: false,
12       initialRoute: MyRoutes.homeRoute,
13       routes: {
14         "/": (context) => LoginPage(),
15         MyRoutes.homeRoute: (context) => HomePage(),
16         MyRoutes.loginRoute: (context) => LoginPage()
17       },
18     ); // MaterialApp
19   }
20 }
```



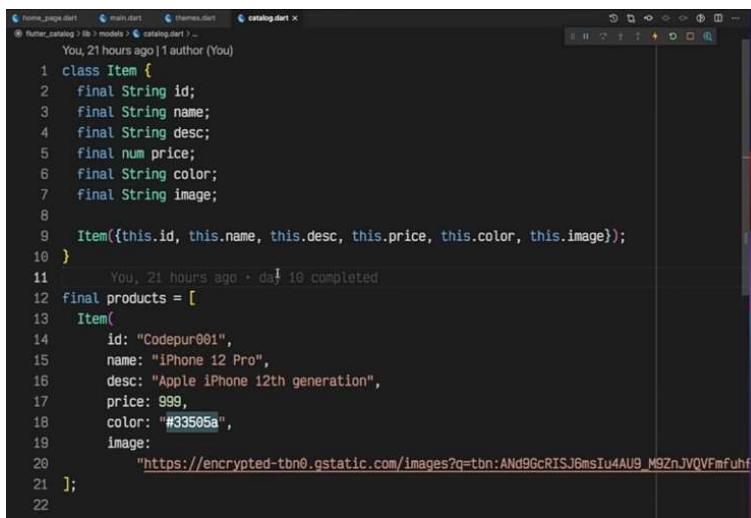
- Using static json data to display

```

1 {
2   "products": [
3     {
4       "id": "Codepur001",
5       "name": "iPhone 12 Pro",
6       "desc": "Apple iPhone 12th generation",
7       "price": "999",
8       "color": "#33505a",
9       "image": "https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcRISJ6msIu4AU9_M9ZnJVQVFmfuhf"
10    },
11    {
12      "id": "Codepur002",
13      "name": "Pixel 5",
14      "desc": "Google Pixel phone 5th generation",
15      "price": "699",
16      "color": "#00ac51",
17      "image": "https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcSd0JhwLvm_uL"
18    },
19    {
20      "id": "Codepur003",
21      "name": "Mi Macbook Air",
22      "desc": "Apple Macbook air with apple silicon",
23      "price": "1099",
24      "color": "#e0bfae",

```

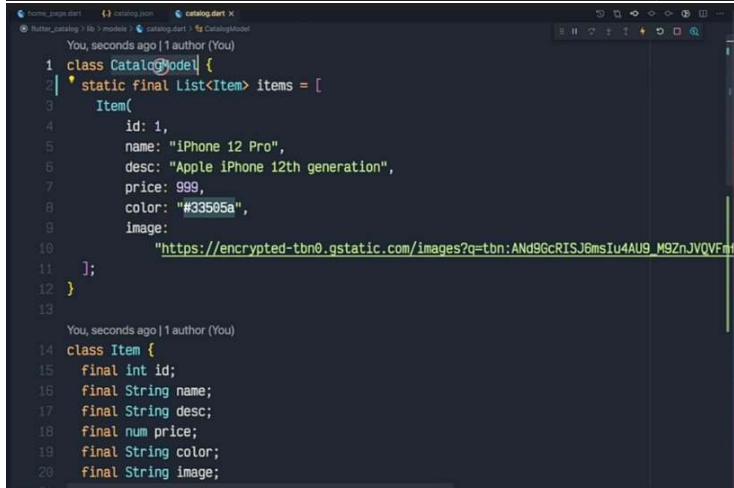
catalog.dart



```

You, 21 hours ago | 1 author (You)
1 class Item {
2   final String id;
3   final String name;
4   final String desc;
5   final num price;
6   final String color;
7   final String image;
8
9   Item({this.id, this.name, this.desc, this.price, this.color, this.image});
10 }
11 You, 21 hours ago + da 10 completed
12 final products = [
13   Item(
14     id: "Codepur001",
15     name: "iPhone 12 Pro",
16     desc: "Apple iPhone 12th generation",
17     price: 999,
18     color: "#33505a",
19     image:
20       "https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcRISJ6msIu4AU9_M9ZnJVQVFmfuhf"
21   );
22 ]

```



```

You, seconds ago | 1 author (You)
1 class CatalogModel {
2   static final List<Item> items = [
3     Item(
4       id: 1,
5       name: "iPhone 12 Pro",
6       desc: "Apple iPhone 12th generation",
7       price: 999,
8       color: "#33505a",
9       image:
10        "https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcRISJ6msIu4AU9_M9ZnJVQVFmfuhf"
11     );
12   ];
13
14 class Item {
15   final int id;
16   final String name;
17   final String desc;
18   final num price;
19   final String color;
20   final String image;

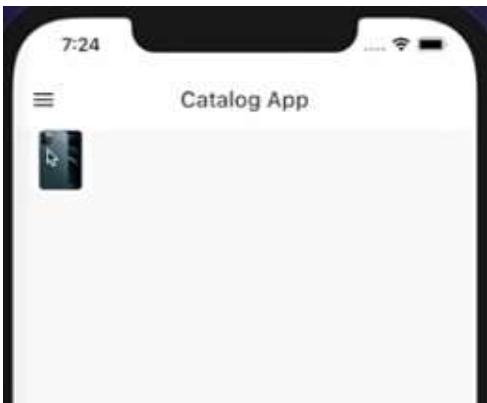
```

```
flutter_catalog>lib>widgets>item_widget.dart>ItemWidget>build
1 import 'package:flutter/material.dart';
2 import 'package:flutter_catalog/models/catalog.dart';
3
4 class ItemWidget extends StatelessWidget {
5   final Item item;
6
7   const ItemWidget({Key key, @required this.item})
8     : assert(item != null),
9       super(key: key);
10
11  @override
12  Widget build(BuildContext context) {
13    return ListTile(
14      leading: Image.network(item.image),
15    ); // ListTile
16  }
17}
18
```

Using

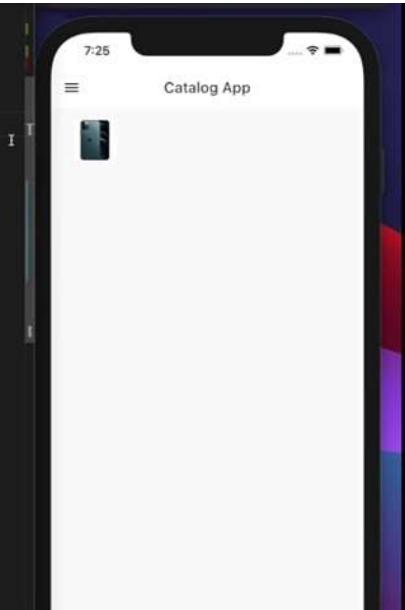
```
home_page.dart X item_widget.dart
flutter_catalog>lib>pages>home_page.dart>HomePage>build
6 class HomePage extends StatelessWidget {
7   final int days = 30;
8   final String name = "Codepur";
9   @override
10  Widget build(BuildContext context) {
11    return Scaffold(
12      appBar: AppBar(
13        title: Text("Catalog App"),
14      ), // AppBar
15      body: ListView.builder(
16        itemCount: CatalogModel.items.length, ✓
17        itemBuilder: (context, index) { ✓
18          return ItemWidget(
19            item: CatalogModel.items[index], You, seconds ago + Un
20          ); // ItemWidget
21        },
22      ), // ListView.builder
23      drawer: MyDrawer(),
24    ); // Scaffold
25  }
26}
```

Output



Giving padding around image

```
6 class HomePage extends StatelessWidget {  
7     final int days = 30;  
8     final String name = "Codepur";  
9     @override  
10    Widget build(BuildContext context) { You, 10 days ago + Day 2 completed  
11        return Scaffold(  
12            appBar: AppBar(  
13                title: Text("Catalog App"),  
14            ), // AppBar  
15            body: Padding(  
16                padding: const EdgeInsets.all(16.0),  
17                child: ListView.builder(  
18                    itemCount: CatalogModel.items.length,  
19                    itemBuilder: (context, index) {  
20                        return ItemWidget(  
21                            item: CatalogModel.items[index],  
22                        ); // ItemWidget  
23                    },  
24                ), // ListView.builder  
25            ), // Padding  
26            drawer: MyDrawer(),  
27        ); // Scaffold  
28    }  
29}
```

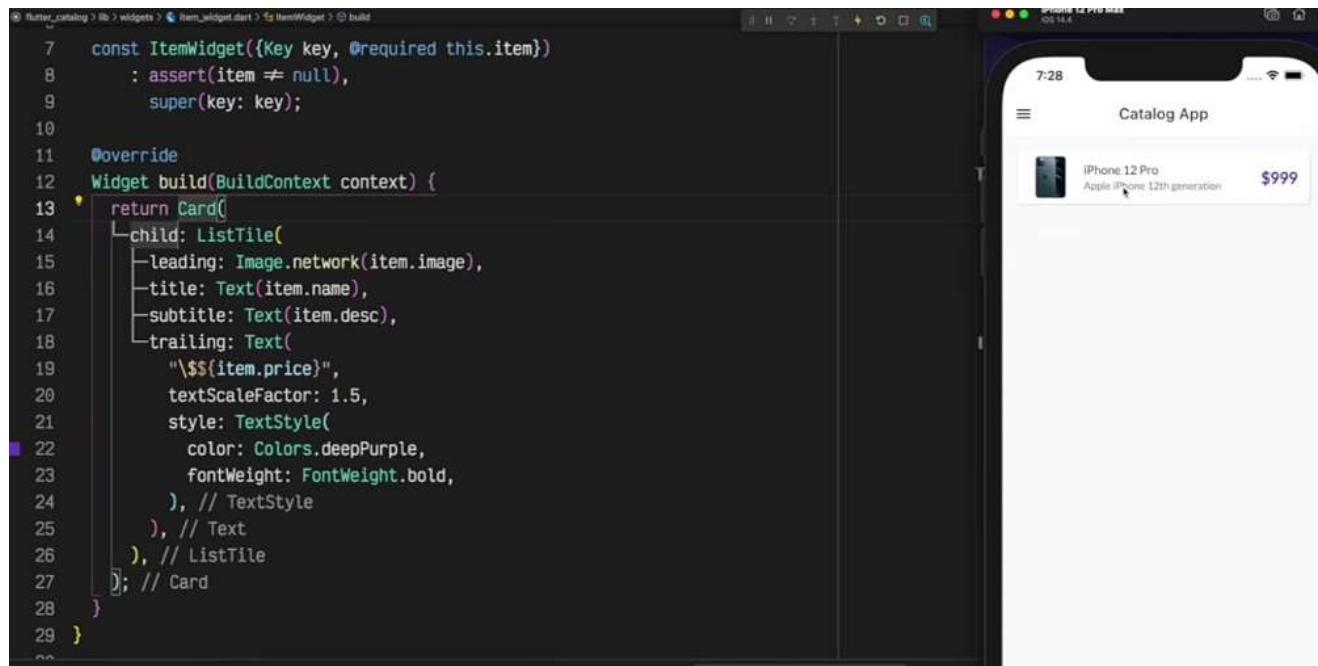


Updating ItemWidget and adding leading, title and trailing

The screenshot shows a development environment with two main panes. On the left is a code editor for a Dart file named `item_widget.dart`. The code defines a class `ItemWidget` that extends `ListTile`. It includes properties for item image, name, description, and price, and styles for the title and subtitle. On the right is a mobile application preview titled "Catalog App" running on an iPhone 12 Pro Max. The app displays a single item card for an iPhone 12 Pro, showing its name, description, and price (\$999).

```
7 const ItemWidget({Key key, @required this.item})
8     : assert(item != null),
9     super(key: key);
10
11 @override
12 Widget build(BuildContext context) {
13     return ListTile(
14         leading: Image.network(item.image),
15         title: Text(item.name),
16         subtitle: Text(item.desc),
17         trailing: Text(
18             "\$${item.price}",
19             textScaleFactor: 1.5,
20             style: TextStyle(
21                 color: Colors.deepPurple,
22                 fontWeight: FontWeight.bold,
23             ), // TextStyle
24         ), // Text
25     ); // ListTile
26 }
27 }
28
```

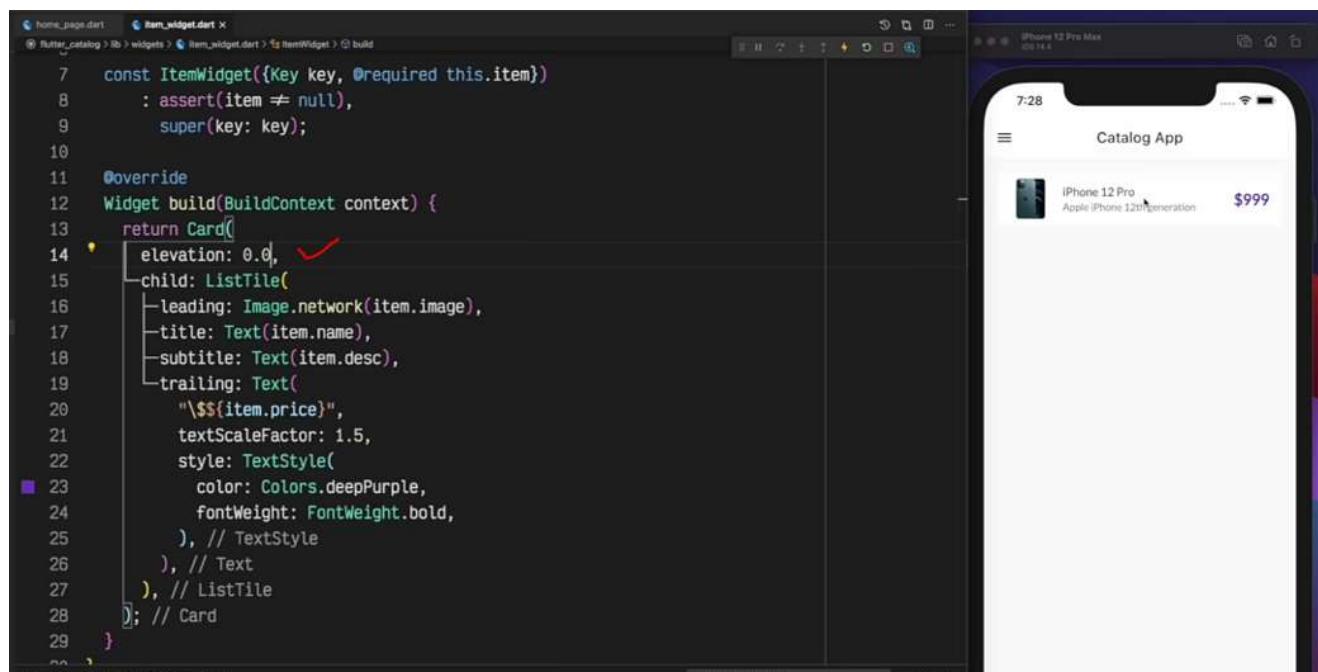
Wrapping in Card to get elevation like effect



The screenshot shows the code for the `ItemWidget` class in a Flutter project. The `build` method returns a `Card` widget containing a `ListTile`. The `ListTile` has a leading image, a title, a subtitle, and a trailing text element displaying the item's price. The code is annotated with comments explaining the structure.

```
7 const ItemWidget({Key key, @required this.item})
8     : assert(item != null),
9     super(key: key);
10
11 @override
12 Widget build(BuildContext context) {
13     return Card(
14         child: ListTile(
15             leading: Image.network(item.image),
16             title: Text(item.name),
17             subtitle: Text(item.desc),
18             trailing: Text(
19                 "\$${item.price}",
20                 textScaleFactor: 1.5,
21                 style: TextStyle(
22                     color: Colors.deepPurple,
23                     fontWeight: FontWeight.bold,
24                 ), // TextStyle
25             ), // Text
26         ), // ListTile
27     ); // Card
28 }
29 }
```

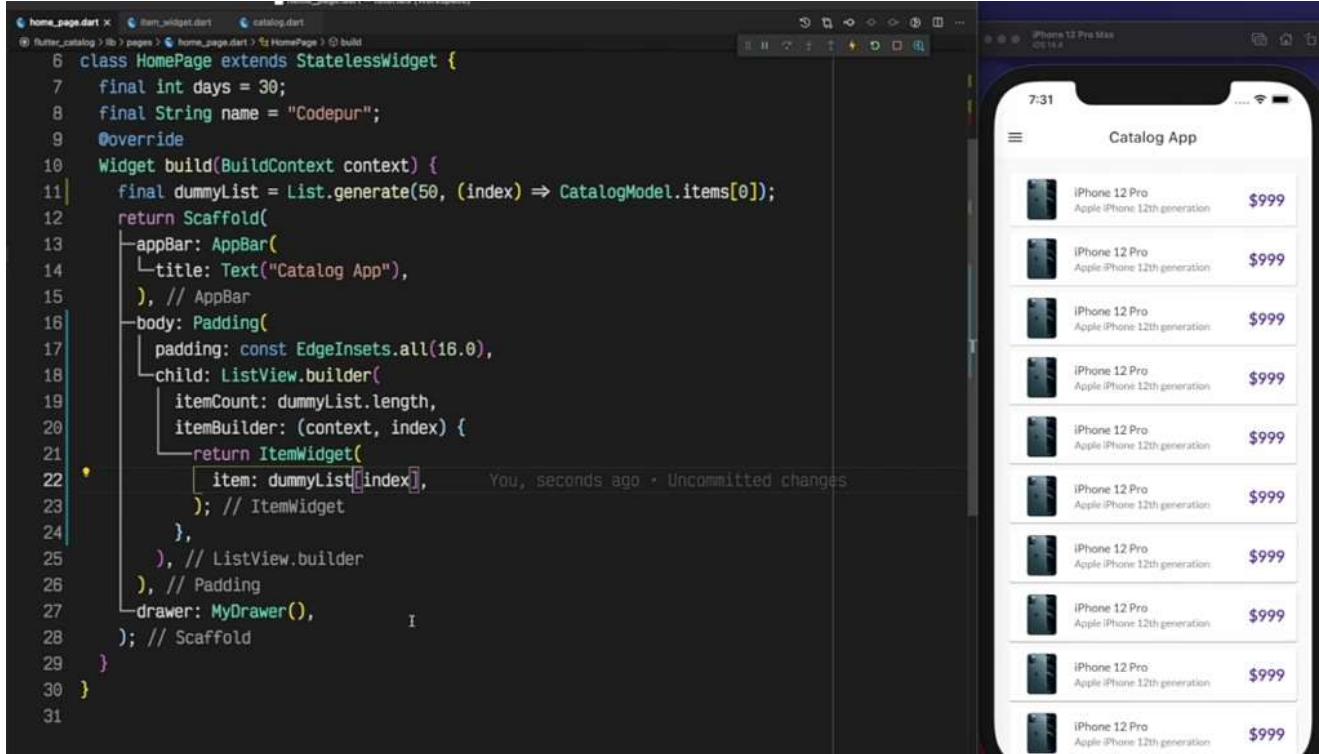
We can set elevation zero to remove the shadow effect



The screenshot shows the same `ItemWidget` code as the previous one, but with the `elevation` property explicitly set to 0.0 on the `Card` widget. A red checkmark is placed next to the `elevation` line to indicate that the change has been made.

```
7 const ItemWidget({Key key, @required this.item})
8     : assert(item != null),
9     super(key: key);
10
11 @override
12 Widget build(BuildContext context) {
13     return Card(
14         elevation: 0.0, ✓
15         child: ListTile(
16             leading: Image.network(item.image),
17             title: Text(item.name),
18             subtitle: Text(item.desc),
19             trailing: Text(
20                 "\$${item.price}",
21                 textScaleFactor: 1.5,
22                 style: TextStyle(
23                     color: Colors.deepPurple,
24                     fontWeight: FontWeight.bold,
25                 ), // TextStyle
26             ), // Text
27         ), // ListTile
28     ); // Card
29 }
```

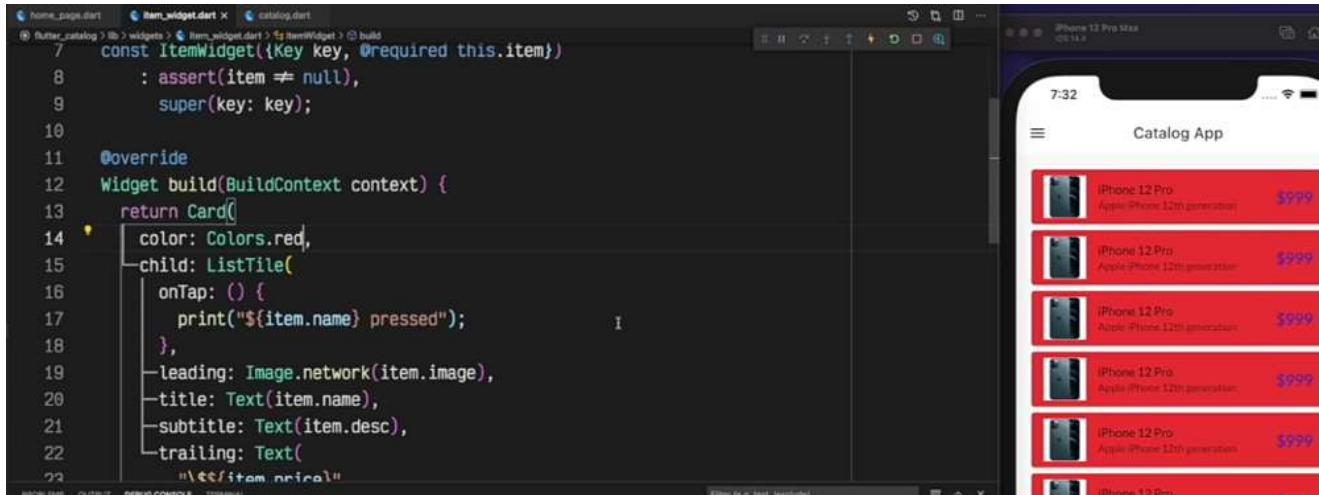
Adding dummy items



The screenshot shows a Flutter project structure with files: home_page.dart, item_widget.dart, and catalog.dart. The code in home_page.dart defines a HomePage class that generates a list of 50 dummy items using List.generate. The generated items are instances of CatalogModel.items[0]. The itemBuilder for each item in the ListView.builder returns an ItemWidget. The ItemWidget uses a Card widget with a red color. The ListTile inside the Card displays the item's name, description, and price. The app is running on an iPhone 12 Pro Max simulator, showing a list of 10 identical items, each with a red background.

```
6 class HomePage extends StatelessWidget {
7   final int days = 30;
8   final String name = "Codepur";
9   @override
10  Widget build(BuildContext context) {
11    final dummyList = List.generate(50, (index) => CatalogModel.items[0]);
12    return Scaffold(
13      appBar: AppBar(
14        title: Text("Catalog App"),
15      ), // AppBar
16      body: Padding(
17        padding: const EdgeInsets.all(16.0),
18        child: ListView.builder(
19          itemCount: dummyList.length,
20          itemBuilder: (context, index) {
21            return ItemWidget(
22              item: dummyList[index],
23            ); // ItemWidget
24          },
25        ), // ListView.builder
26      ), // Padding
27      drawer: MyDrawer(),
28    ); // Scaffold
29  }
30}
31
```

Giving color to list card



The screenshot shows the same project structure and code as the previous screenshot, but the color of the cards has been changed. In the ItemWidget constructor, the color of the Card is set to Colors.red. The app is running on an iPhone 12 Pro Max simulator, showing a list of 10 items where each card now has a red background.

```
7 const ItemWidget({Key key, @required this.item})
8   : assert(item != null),
9   super(key: key);
10
11 @override
12 Widget build(BuildContext context) {
13   return Card(
14     color: Colors.red,
15     child: ListTile(
16       onTap: () {
17         print("${item.name} pressed");
18       },
19       leading: Image.network(item.image),
20       title: Text(item.name),
21       subtitle: Text(item.desc),
22       trailing: Text(
23         "\$${item.price}"
24       ),
25     ),
26   );
27 }
```

Giving circular stadium like shape to card



```
flutter_catalog> lib>widgets> item_widget.dart > ItemWidget > build
7     const ItemWidget({Key key, @required this.item})
8         : assert(item != null),
9         super(key: key);
10
11    @override
12    Widget build(BuildContext context) {
13        return Card(
14            shape: StadiumBorder(),
15            child: ListTile(
16                onTap: () {
17                    print("${item.name} pressed");
18                },
19                leading: Image.network(item.image),
20                title: Text(item.name),
21                subtitle: Text(item.desc),
22                trailing: Text(
23                    "\$${item.price}");
24    }
25}
```

Adding file as accessible in pubspec

```
⑧ flutter_catalog 39 pubspec.yaml
39 # following page: https://dart.dev/tools/pub/pubspec
40
41 # The following section is specific to Flutter.
42 flutter:
43   # The following line ensures that the Material Icons font is
44   # included with your application, so that you can use the icons in
45   # the material Icons class.
46   uses-material-design: true
47
48   # To add assets to your application, add an assets section, like this:
49 assets:
50   - assets/images/
51   - assets/files/      You, seconds ago • Uncommitted changes
52   #   - images/a_dot_ham.jpeg
53
```

Loading data

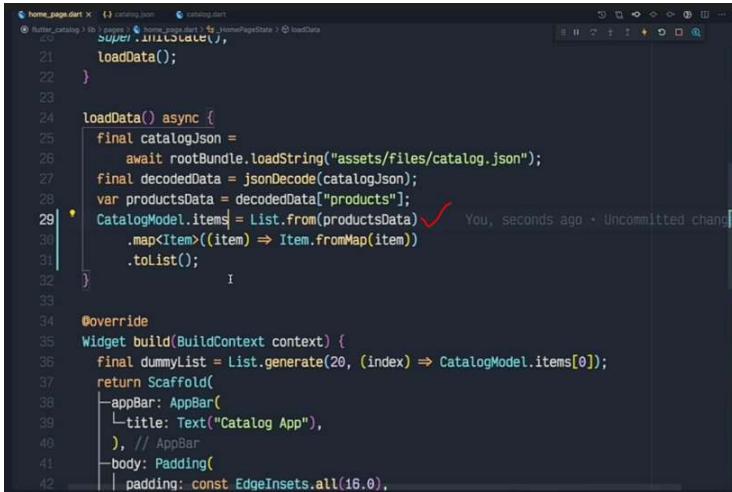
```
⑧ flutter_catalog > lib > pages > home_page.dart > _HomePageState > loadData
13   final int days = 30;
14
15   final String name = "Codepur";
16
17   @override
18   void initState() {
19     super.initState();
20     loadData();
21   }
22
23   loadData() async [
24     var catalogJson = await rootBundle.loadString("assets/files/catalog.json");
25     print(catalogJson);      You, seconds ago + Uncommitted changes
26   ]
27
28   @override
29   Widget build(BuildContext context) {
30     final dummyList = List.generate(20, (index) => CatalogModel.items[0]);
31     return Scaffold(
32       appBar: AppBar(
33         title: Text("Catalog App"),
34       ), // AppBar
35   
```

Config

```
⑧ flutter_catalog > assets > files > catalog.json > products > () > #id
You, 3 days ago | 1 author (You)
1  {
2   "products": [
3     {
4       "id": 1,      You, 3 days ago + day13 completed
5       "name": "iPhone 12 Pro",
6       "desc": "Apple iPhone 12th generation",
7       "price": 999,
8       "color": "#33505a",
9       "image": "https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcRISJ6msIu4AU9_Mk
10      },
11      {
12        "id": 2,
13        "name": "Pixel 5",
14        "desc": "Google Pixel phone 5th generation",
15        "price": 699,
16        "color": "#00ac51",
17        "image": "https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcSd0JhwLvm_uLDLc-
18      },
19      {
20        "id": 3,
```

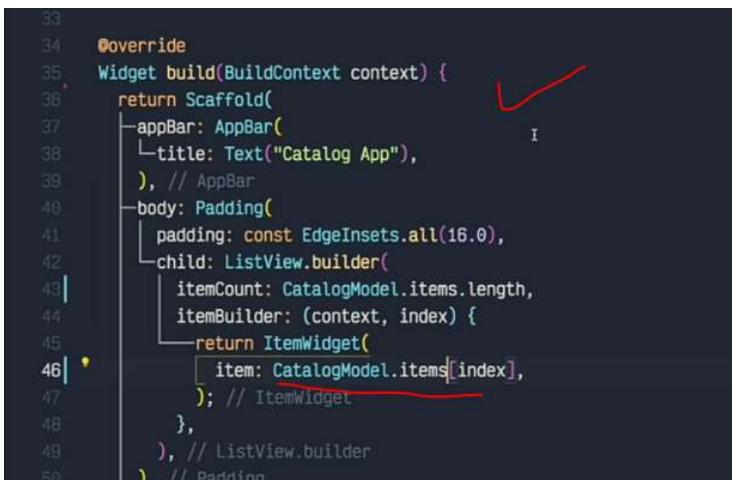
```
flutter_catalog > lib > models > catalog.dart > CatalogModel
You, seconds ago | 1 author (You)
1 class CatalogModel {
2     static List<Item> items = [      You, seconds ago + Uncommitted changes
3         Item(
4             id: 1,
5             name: "iPhone 12 Pro",
6             desc: "Apple iPhone 12th generation",
7             price: 999,
8             color: "#33505a",
9             image:
10                "https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcRISJ6msIu4AU9_M9ZnJVQVFm
11            ];
12        }
13
flutter_catalog > lib > models > catalog.dart > Item > fromMap
12 }
13
You, seconds ago | 1 author (You)
14 class Item {
15     final int id;
16     final String name;
17     final String desc;
18     final num price;
19     final String color;
20     final String image;
21
22     Item({this.id, this.name, this.desc, this.price, this.color, this.image});
23
24     factory Item.fromMap(Map<String, dynamic> map) {
25         Item(
26             id: map["id"]
27             name: map["name"]
28             desc: map["desc"]
29             price: map["price"]
30             color: map["color"]
31             image: map["image"]      You, seconds ago + Uncommitted changes
32         );
33     }
34
35     toMap() => {
36         "id": id,
37         "name": name,
38         "desc": desc,
39         "price": price,
40         "color": color,
41         "image": image      You, seconds ago + Uncommitted changes
42     };
43 }
44 }
```

Setting the loaded data to catalogModel.items



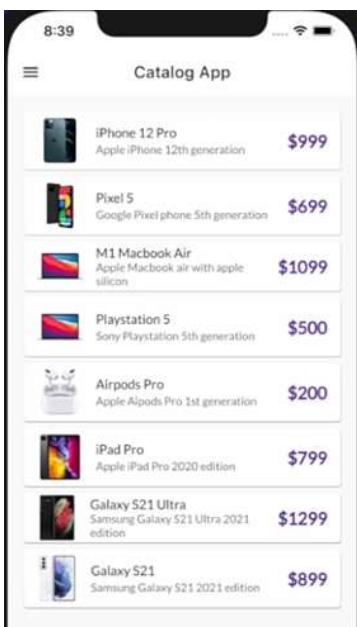
```
home_page.dart X catalog.json catalog.dart
21   loadData();
22 }
23
24 loadData() async {
25   final catalogJson =
26     await rootBundle.loadString("assets/files/catalog.json");
27   final decodedData = jsonDecode(catalogJson);
28   var productsData = decodedData["products"];
29   CatalogModel.items = List.from(productsData) ✓ You, seconds ago • Uncommitted change
30     .map<Item>((item) => Item.fromMap(item))
31     .toList();
32 }
33
34 @override
35 Widget build(BuildContext context) {
36   final dummyList = List.generate(20, (index) => CatalogModel.items[0]);
37   return Scaffold(
38     appBar: AppBar(
39       title: Text("Catalog App"),
40     ), // AppBar
41     body: Padding(
42       padding: const EdgeInsets.all(16.0),
43     ),
44   );
45 }
```

Removed dummylist

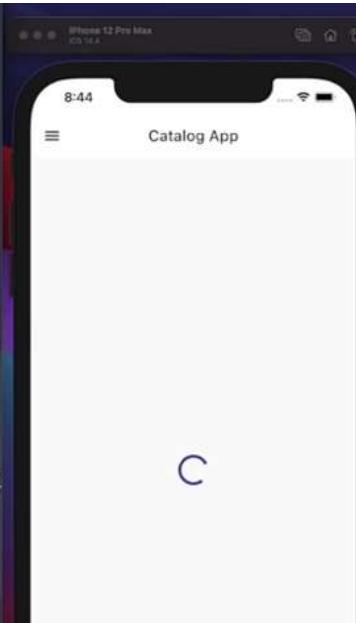


```
33
34 @override
35 Widget build(BuildContext context) {
36   return Scaffold(
37     appBar: AppBar(
38       title: Text("Catalog App"),
39     ), // AppBar
40     body: Padding(
41       padding: const EdgeInsets.all(16.0),
42       child: ListView.builder(
43         itemCount: CatalogModel.items.length,
44         itemBuilder: (context, index) {
45           return ItemWidget(
46             item: CatalogModel.items[index],
47           ); // ItemWidget
48         },
49       ), // ListView.builder
50     ), // Padding
51   );
52 }
```

Output



Adding loader when data is loading



A screenshot of an iPhone 12 Pro Max displaying the "Catalog App". The screen shows a title bar with the text "Catalog App" and a circular progress indicator in the center of the main content area, indicating that data is still loading.

```
38 appBar: AppBar(
39   title: Text("Catalog App"),
40 ), // AppBar
41 body: Padding(
42   padding: const EdgeInsets.all(16.0),
43   child: (CatalogModel.items != null && CatalogModel.items.isNotEmpty)
44     ? ListView.builder(
45       itemCount: CatalogModel.items.length,
46       itemBuilder: (context, index) {
47         return ItemWidget(
48           item: CatalogModel.items[index],
49         );
50       },
51     ) // ListView.builder
52     : Center(
53       child: CircularProgressIndicator(),
54     ); // Center
55   ), // Padding
56   drawer: MyDrawer(),
57 ); // Scaffold
58 }
59 
```

➤ Converting listView.builder to GridView

With 2 counts in row



A screenshot of an iPhone 12 Pro Max displaying the "Catalog App". The screen shows a grid view of various items, including multiple iPhone models and a pair of AirPods, arranged in two columns per row.

```
36 void
37   build(BuildContext context) {
38   return Scaffold(
39     appBar: AppBar(
40       title: Text("Catalog App"),
41     ), // AppBar
42     body: Padding(
43       padding: const EdgeInsets.all(16.0),
44       child: (CatalogModel.items != null && CatalogModel.items.isNotEmpty)
45         ? GridView.builder(
46           gridDelegate: SliverGridDelegateWithFixedCrossAxisCount(
47             crossAxisCount: 2,
48           ), // SliverGridDelegateWithFixedCrossAxisCount
49           itemBuilder: (context, index) {
50             final item = CatalogModel.items[index];
51             return Card(
52               shape: RoundedRectangleBorder(
53                 borderRadius: BorderRadius.circular(15), // RoundedRectangleBorder
54               child: GridTile(child: Image.network(item.image)); // Card
55             ),
56             itemCount: CatalogModel.items.length,
57           ) // GridView.builder
58           : Center(
59             child: CircularProgressIndicator(),
60           ); // Center
61   ), // Padding
62   drawer: MyDrawer(),
63 ); // Scaffold
64 
```

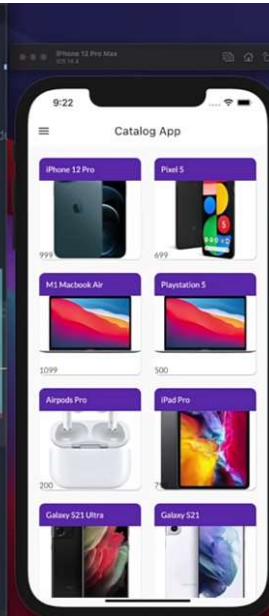
Increasing count in row

```
42 body: Padding(  
43   padding: const EdgeInsets.all(16.0),  
44   child: (CatalogModel.items != null && CatalogModel.items.isNotEmpty)  
45     ? GridView.builder(  
46       gridDelegate: SliverGridDelegateWithFixedCrossAxisCount(  
47         crossAxisCount: 3,           You, seconds ago + Uncommitted changes  
48         mainAxisSpacing: 16,  
49         crossAxisSpacing: 16,  
50       ), // SliverGridDelegateWithFixedCrossAxisCount  
51       itemBuilder: (context, index) {  
52         final item = CatalogModel.items[index];  
53         return Card(  
54           clipBehavior: Clip.antiAlias,  
55           shape: RoundedRectangleBorder(  
56             borderRadius: BorderRadius.circular(10)), // RoundedRectangleBorder  
57           child: GridTile(  
58             header: Text(item.name),  
59             child: Image.network(item.image),  
60             footer: Text(  
61               item.price.toString(),  
62             ), // Text  
63           ); // GridTile // Card  
64         },  
65         itemCount: CatalogModel.items.length,  
66       ) // GridView.builder  
67     : Center(  
68       child: CircularProgressIndicator(),  
69     ), // Padding
```

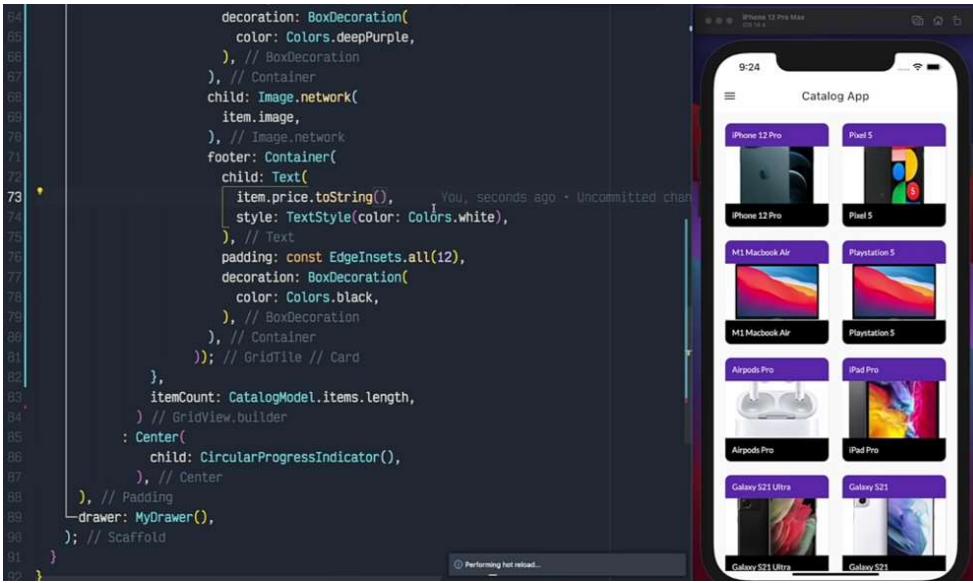


Adding header & footer in gridView

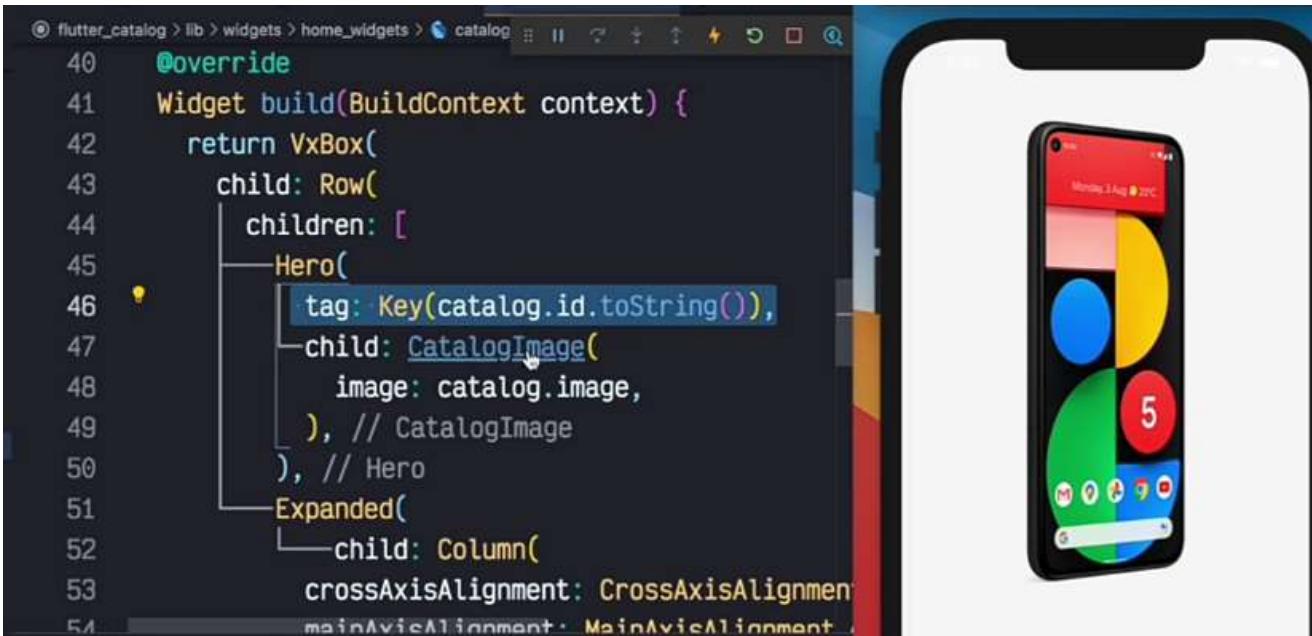
```
41 body: Padding(  
42   padding: const EdgeInsets.all(16.0),  
43   child: (CatalogModel.items != null && CatalogModel.items.isNotEmpty)  
44     ? GridView.builder(  
45       gridDelegate: SliverGridDelegateWithFixedCrossAxisCount(  
46         crossAxisCount: 2,           You, seconds ago + Uncommitted changes  
47         mainAxisSpacing: 16,  
48         crossAxisSpacing: 16,  
49       ), // SliverGridDelegateWithFixedCrossAxisCount  
50       itemBuilder: (context, index) {  
51         final item = CatalogModel.items[index];  
52         return Card(  
53           clipBehavior: Clip.antiAlias,  
54           shape: RoundedRectangleBorder(  
55             color: Colors.deepPurple),  
56             borderRadius: BorderRadius.circular(10)), // RoundedRectangleBorder  
57           child: GridTile(  
58             header: Container(  
59               child: Text(  
60                 item.name,  
61                 style: TextStyle(color: Colors.white),  
62               ), // Text  
63               padding: const EdgeInsets.all(12),  
64               decoration: BoxDecoration(  
65                 color: Colors.deepPurple),  
66               ), // BoxDecoration  
67             ), // Container  
68             child: Image.network(item.image),  
69             footer: Text(  
70               item.price.toString(),  
71             ), // Text  
72           ); // GridTile // Card  
73         },  
74         itemCount: CatalogModel.items.length,  
75       ) // GridView.builder  
76     : Center(  
77       child: CircularProgressIndicator(),  
78     ), // Center  
79   ), // Padding  
80   drawer: MyDrawer(),  
81 
```

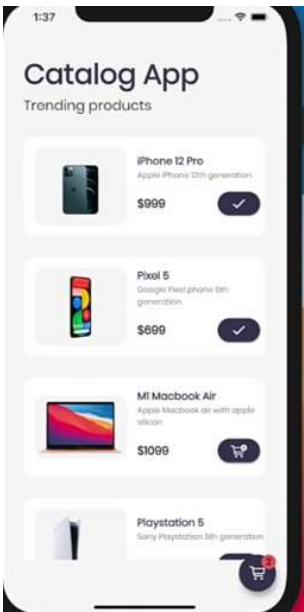


And

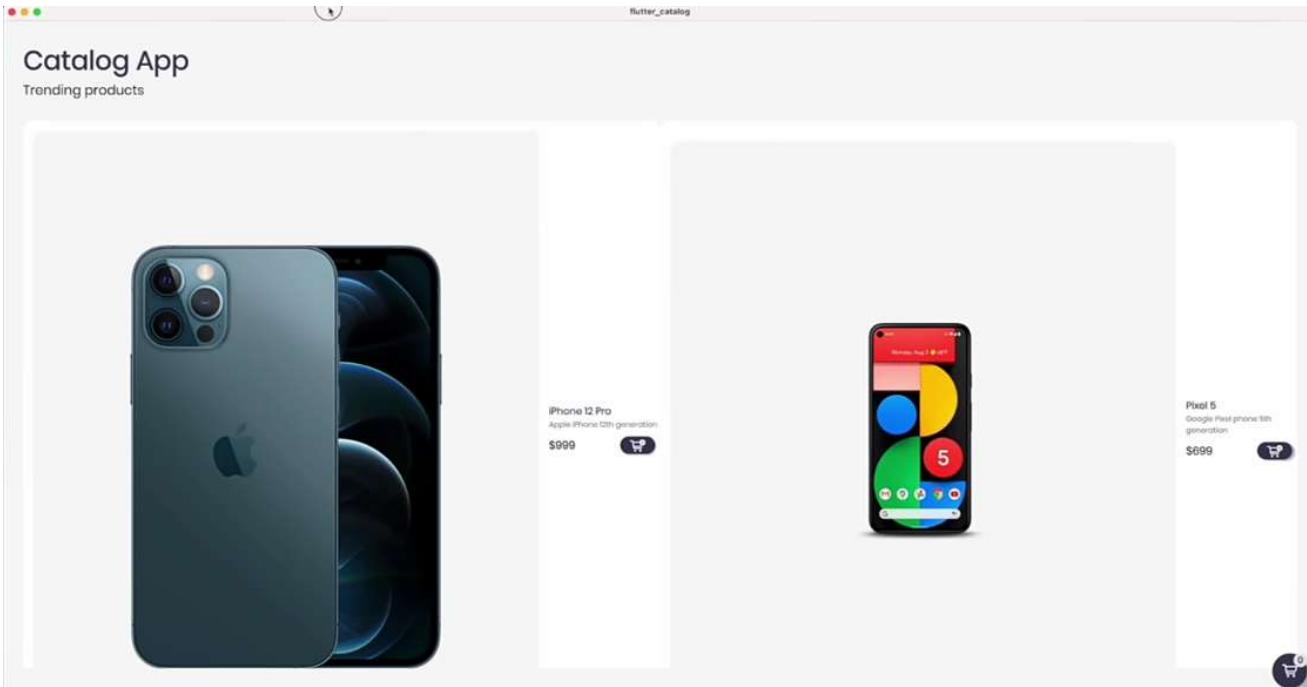


Hero to display full screen image





We can run this flutter app in macos, browser,ios and android as well



- Useful Widgets

1. Floating draggable widget

floating_draggable_widget 2.1.5 

Published 6 months ago 

SDK FLUTTER PLATFORM ANDROID IOS LINUX MACOS WEB WINDOWS

19 79

Readme Changelog Example Installing Versions Scores

[example/lib/main.dart](#)

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

void main() {
    runApp(const MyApp());
}

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

    @override
    Widget build(BuildContext context) {
        return const MaterialApp(
            title: 'Floating Draggable Widget',
            home: MyHomePage(),
        );
    }
}

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

```
Widget build(BuildContext context) {
  return FloatingDraggableWidget(
    floatingWidget: FloatingActionButton(
      onPressed: () {},
      child: const Icon(Icons.add, size: 50),
    ),
    floatingWidgetHeight: 90,
    floatingWidgetWidth: 90,
    dx: 200,
    dy: 300,
    deleteWidgetDecoration: const BoxDecoration(
      gradient: LinearGradient(
        colors: [Colors.white12, Colors.grey],
        begin: Alignment.topCenter,
        end: Alignment.bottomCenter,
        stops: [.0, 1],
      ),
      borderRadius: BorderRadius.only(
        topLeft: Radius.circular(50),
        topRight: Radius.circular(50),
      ),
    ),
    deleteWidget: Container(
      decoration: BoxDecoration(
        shape: BoxShape.circle,
        border: Border.all(width: 2, color: Colors.black87),
      ),
      child: const Icon(Icons.close, color: Colors.black87),
    ),
    onDeleteWidget: () {
      debugPrint('Widget deleted');
    },
    mainScreenWidget: Scaffold(
      appBar: AppBar(
        title: const Text('Floating Animated Widget'),
      ),
      body: Center(
        child: Column(
          mainAxisAlignment: MainAxisAlignment.center,
          children: <Widget>[
            Text(
              'Floating Animated Widget',
              style: Theme.of(context).textTheme.headline6,
            ),
          ],
        ),
      ),
    ),
  );
}
```

The screenshot shows the Flutter IDE interface. On the left, the code editor displays the `home_screen.dart` file with the same code as the previous snippet. On the right, the iPhone X simulator window shows the application running. The app has a white background with a purple floating button at the top center. Below it, there is a purple circular container with a white icon and the text "Floating Animated Widget". The status bar at the top of the simulator screen shows the time as 3:16.

```
home_screen.dart
57   dx: 200,
58   dy: 300,
59   deleteWidgetDecoration: const BoxDecoration(
60     gradient: LinearGradient(
61       colors: [Colors.white12, Colors.grey],
62       begin: Alignment.topCenter,
63       end: Alignment.bottomCenter,
64       stops: [.0, 1],
65     ), // LinearGradient
66     borderRadius: BorderRadius.only(
67       topLeft: Radius.circular(50),
68       topRight: Radius.circular(50),
69     ), // BorderRadius.only
70   ), // BoxDecoration
71   deleteWidget: Container(
72     decoration: BoxDecoration(
73       shape: BoxShape.circle,
74       border: Border.all(width: 2, color: Colors.black87),
75     ), // BoxDecoration
76     child: const Icon(Icons.close, color: Colors.black87),
77   ), // Container
78   onDeleteWidget: () {
79     debugPrint('Widget deleted');
80   },
81
82 );
83 // FloatingDraggableWidget
84 }
```

2. Delightful toast

delightful_toast 1.1.0  

Published 7 months ago • swaroopsambhayya.com Dart 3 compatible

SDK FLUTTER PLATFORM ANDROID IOS LINUX MACOS WEB WINDOWS   

Readme Changelog Example Installing Versions Scores

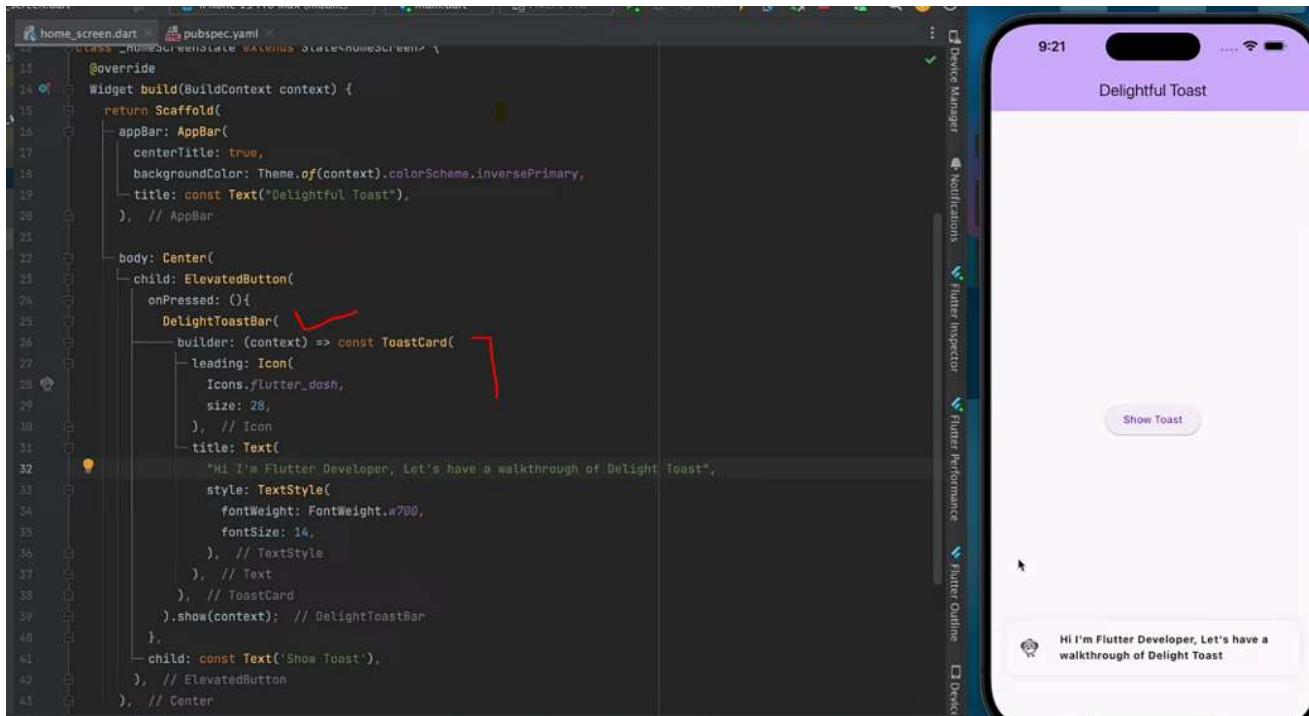
A delightful and very interactive toast or a snackbar to add more delight to your application

 https://github.com/SwaroopSambhayya/delight_toast/assets/31922733/45b77a2e-7a73-44a0-899d-8d2f740097fb

Usage

Use the toastbar whenever a event has been triggered like onPressed/onTapped or in any other situation where the event is enforced

```
ElevatedButton( onPressed:(){  
    DelightToastBar(  
        builder: (context) => const ToastCard(  
            leading: Icon(  
                Icons.flutter_dash,  
                size: 28,  
            ),  
            title: Text(  
                "Hi I'm Dash, Let's have a walkthrough of Delight Toast",  
                style: TextStyle(  
                    fontWeight: FontWeight.w700,  
                    fontSize: 14,  
                ),  
            ),  
        ),  
    ).show(context);  
}, child: const Text("Toast!"))
```



The screenshot shows the Flutter IDE interface. On the left, the code for `home_screen.dart` is displayed, showing the implementation of the `DelightToastBar`. A red arrow points from the code to the `Flutter Inspector` panel on the right, which shows a screenshot of an iPhone X displaying a purple header bar with the text "Delightful Toast". Below the header, there is a white toast card with a black border and rounded corners. The toast contains the text "Hi I'm Flutter Developer, Let's have a walkthrough of Delight Toast". The `Flutter Inspector` panel also shows the device manager, notifications, and other developer tools.

3. Flutter_config_plus (.env)

flutter_config_plus 1.1.1  flutter_config_plus: ^1.1.1 copied to clipboard

Published 13 months ago Dart 3 compatible

SDK FLUTTER PLATFORM ANDROID IOS 10

Readme Changelog Example Installing Versions Scores

Config Variables for your Flutter apps

Basic Usage

Create a new file `.env` in the root of your Flutter app:

```
API_URL=https://myopi.com
FABRIC_ID=abcdefgfh
```

load all environment variables in `main.dart`

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

void main() async {
  WidgetsFlutterBinding.ensureInitialized(); // Required by FlutterConfigPlus
  await FlutterConfigPlus.loadEnvVariables();
}

runApp(MyApp());
}
```

Now you can access your environment variables anywhere in your app.

```
import 'package:flutter_config_plus/flutter_config.dart';
FlutterConfigPlus.get('FABRIC_ID') // returns 'abcdefgfh'
```

Keep in mind this module doesn't obfuscate or encrypt secrets for packaging, so do not store sensitive keys in `.env`. It's basically impossible to prevent users from reverse engineering mobile app secrets, so design your app (and APIs) with that in mind.

4. Carousel_slider

carousel_slider 4.2.1 

Published 10 months ago • @ serenader.me Dart 3 compatible

SDK FLUTTER PLATFORM ANDROID IOS LINUX MACOS WEB WINDOWS 4.6K

Readme Changelog Example Installing Versions Scores

carousell_slider

4674 LINES 120 PUB POINTS 100% POPULARITY

Publisher
@ serenader.me

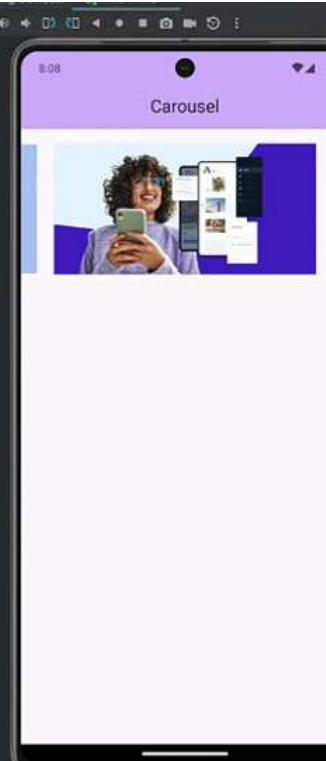
```
import 'package:carousel_slider/carousel_slider.dart';
import 'package:flutter/material.dart';

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

  @override
  State<HomeScreen> createState() => _HomeScreenState();
}

class _HomeScreenState extends State<HomeScreen> {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        centerTitle: true,
        backgroundColor: Theme.of(context).colorScheme.inversePrimary,
        title: const Text("Carousel"),
      ), // AppBar

      body: Column(
        children: [
          SizedBox(
            height: 200, width: double.infinity,
            child: CarouselSlider(
              options: CarouselOptions(
                autoPlay: true,
                enlargeCenterPage: true,
              ), // CarouselOptions
              items: [
                Image.network('https://storage.googleapis.com/website-production/uploads/2023/06/best-mobile-ad-platforms-france.jpg'),
                Image.network('https://about.fb.com/wp-content/uploads/2023/01/Ad-Fairness-Technology-Header.jpg'),
                Image.network('https://storage.googleapis.com/gweb-unshing-pubish-prod/original_images/assets-01_-300dpi.j')
              ], // CarouselSlider
            ), // SizedBox
        ],
      ),
    );
  }
}
```



5. Input form validator and controller

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

  @override
  State<HomeScreen> createState() => _HomeScreenState();
}

class _HomeScreenState extends State<HomeScreen> {

  TextEditingController writehereController = TextEditingController();

  final _formKey = GlobalKey<FormState>();

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        centerTitle: true,
        backgroundColor: Theme.of(context).colorScheme.inversePrimary,
        title: const Text("Form Validation"),
      ), // AppBar
      body: Padding(
        padding: const EdgeInsets.symmetric(horizontal: 30),
        child: Form(
          key: _formKey,
          child: Column(
            mainAxisAlignment: MainAxisAlignment.center,
            children: [
              TextFormField(
                controller: writehereController,
                decoration: const InputDecoration(
                  hintText: "Write here..."
                ), // InputDecoration
                validator: (value){
                  if(value!.isEmpty){
                    return "Write here";
                  }
                },
              ),
              const SizedBox(height: 20),
              ElevatedButton(
                onPressed: (){
                  if(_formKey.currentState!.validate()){
                    print("Validate");
                  }
                },
                child: const Text("Submit")) // ElevatedButton
            ],
          ), // Column
        ), // Padding
      ); // Scaffold
  }
}
```

Output

Form Validation

Write here...

Write here

Submit

6. Encrypted Shared Preference

flutter_secure_storage 9.0.0

Published 4 months ago • ⚭ steenbakker.dev (Dart 3 compatible)
SDK FLUTTER PLATFORM ANDROID IOS LINUX MACOS WEB WINDOWS 3.2K

Readme Changelog Example Installing Versions Scores

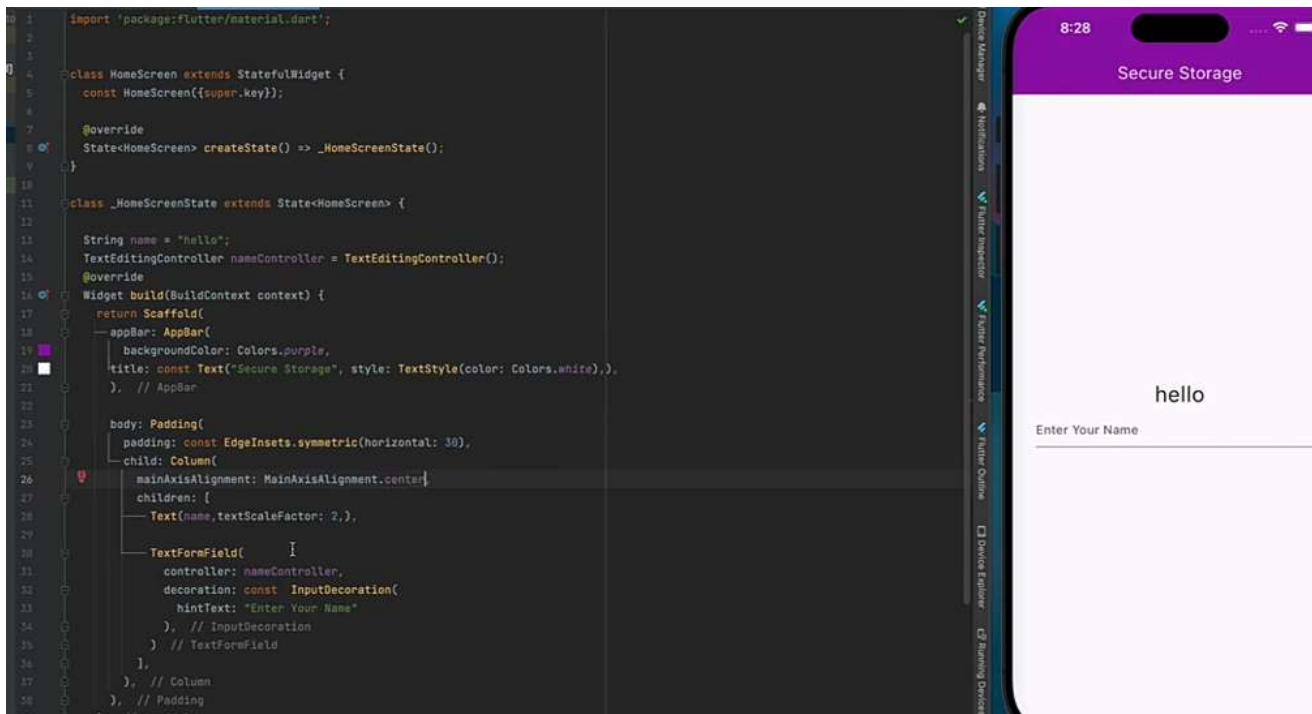
flutter_secure_storage

Note: usage of encryptedSharedPreferences[#]

When using the `encryptedSharedPreferences` parameter on Android, make sure to pass the option to the constructor instead of the function like so:

```
AndroidOptions _getAndroidOptions() => const AndroidOptions(
    encryptedSharedPreferences: true,
);
final storage = FlutterSecureStorage(options: _getAndroidOptions());
```

This will prevent errors due to mixed usage of `encryptedSharedPreferences`. For more info, see this issue.



Updated

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

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

  @override
  State<HomeScreen> createState() => _HomeScreenState();
}

class _HomeScreenState extends State<HomeScreen> {
  String name = "";
  TextEditingController nameController = TextEditingController();

  @override
  void initState() {
    SecureStorage().getName().then((value) {
      setState(() {
        name = value;
      });
    });
    super.initState();
  }
}
```

```
    @override
    Widget build(BuildContext context) {
      return Scaffold(
        appBar: AppBar(
          backgroundColor: Colors.purple,
          title: const Text("Secure Storage", style: TextStyle(color: Colors.white)),
          actions: [
            IconButton(
              onPressed: () {
                SecureStorage().clearSecureStorage();
              },
              icon: const Icon(Icons.delete_forever_rounded, color: Colors.white),
            ),
          ],
        ), // AppBar

        body: Padding(
          padding: const EdgeInsets.symmetric(horizontal: 30),
          child: Column(
            mainAxisAlignment: MainAxisAlignment.center,
            children: [
              Text(name, textScaleFactor: 2.0),

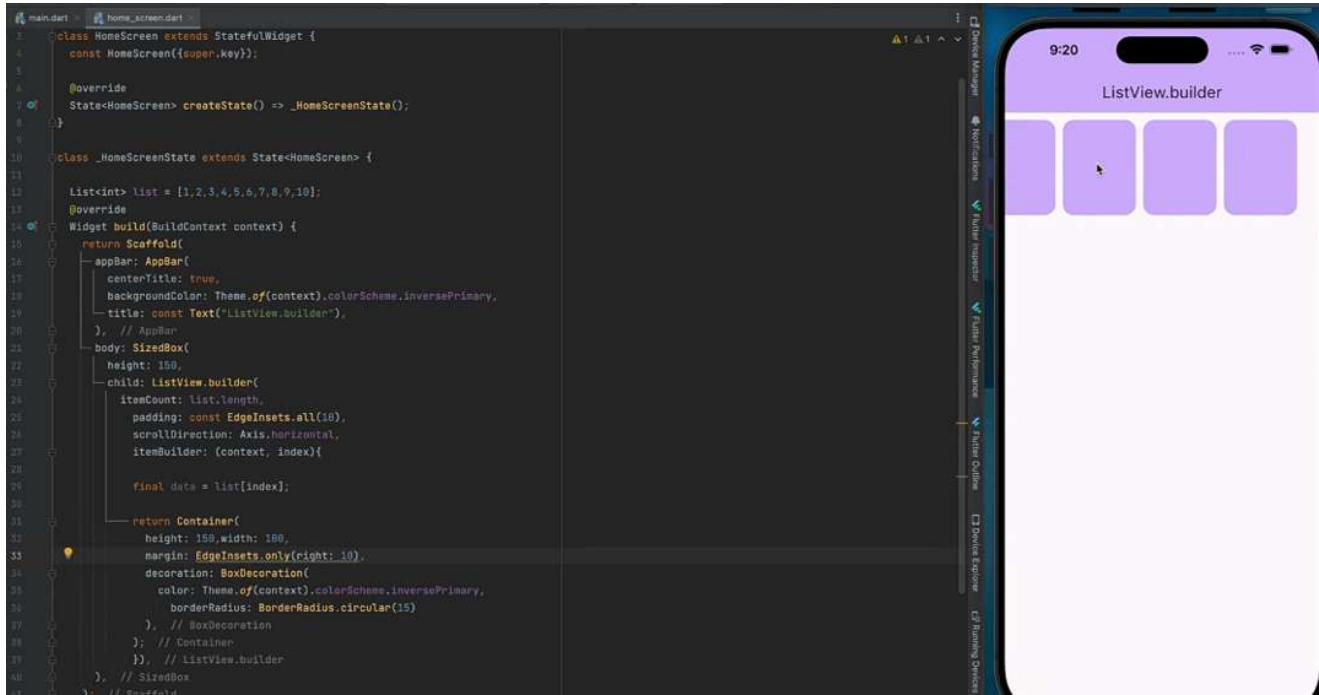
              const SizedBox(height: 20),

              TextFormField(
                controller: nameController,
                decoration: const InputDecoration(
                  hintText: "Enter Your Name"
                ), // InputDecoration
              ), // TextFormField

              const SizedBox(height: 50),
            ],
          ),
        ),
      );
    }
  }
}
```

```
        SizedBox(
            height: 50.width: double.infinity,
            child: ElevatedButton(
                onPressed: () {
                    SecureStorage().setName(nameController.text).then((value)
                        SecureStorage().getName().then((value){
                            setState(() {
                                name = value;
                            });
                        });
                    );
                },
            ),
            child: const Text("Store"), // ElevatedButton
        ); // SizedBox
    ),
),
), // column
), // Padding
); // Scaffold
};
```

7. Horizontal listView.builder



```
main.dart
1 class HomeScreen extends StatefulWidget {
2   const HomeScreen({super.key});
3
4   @override
5   State<HomeScreen> createState() => _HomeScreenState();
6 }
7
8 class _HomeScreenState extends State<HomeScreen> {
9
10   List<int> list = [1,2,3,4,5,6,7,8,9,10];
11
12   @override
13   Widget build(BuildContext context) {
14     return Scaffold(
15       appBar: AppBar(
16         centerTitle: true,
17         backgroundColor: Theme.of(context).colorScheme.inversePrimary,
18         title: const Text("ListView.builder"),
19       ), // AppBar
20       body: SizedBox(
21         height: 150,
22         child: ListView.builder(
23           itemCount: list.length,
24           padding: const EdgeInsets.all(10),
25           scrollDirection: Axis.horizontal,
26           itemBuilder: (context, index){
27
28             final data = list[index];
29
30             return Container(
31               height: 150, width: 100,
32               margin: EdgeInsets.only(right: 10),
33               decoration: BoxDecoration(
34                 color: Theme.of(context).colorScheme.inversePrimary,
35                 borderRadius: BorderRadius.circular(15)
36               ); // BoxDecoration
37             ); // Container
38           }, // ListView.builder
39         ), // SizedBox
40       ); // Scaffold
41   }
42 }
```

Adding animation

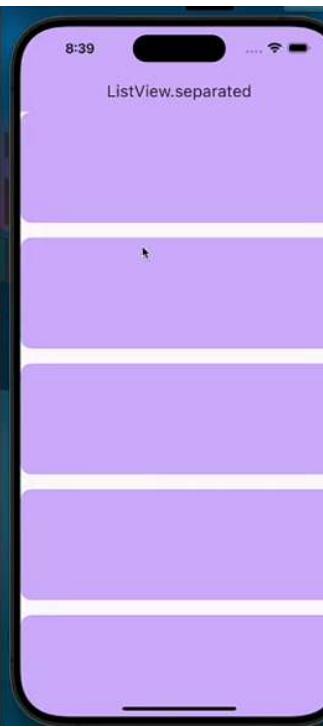


```
body: SizedBox(
  height: 150,
  child: ListView.builder(
    itemCount: list.length,
    padding: const EdgeInsets.all(10),
    [parent]
    physics: NeverScrollableScrollPhysics(),
    scrollDirection: Axis.horizontal,
    itemBuilder: (context, index){

      final data = list[index];
```

8. ListView.separated

a) Giving space as separator



A screenshot of an iPhone X simulator running Flutter. The screen is purple and displays five white rectangular boxes arranged vertically. Each box has a height of 150 pixels and a circular border radius of 15 pixels. The background color of the list items is the inverse primary color of the theme. The title "ListView.separated" is displayed at the top of the screen.

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

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

  @override
  State<HomeScreen> createState() => _HomeScreenState();
}

class _HomeScreenState extends State<HomeScreen> {

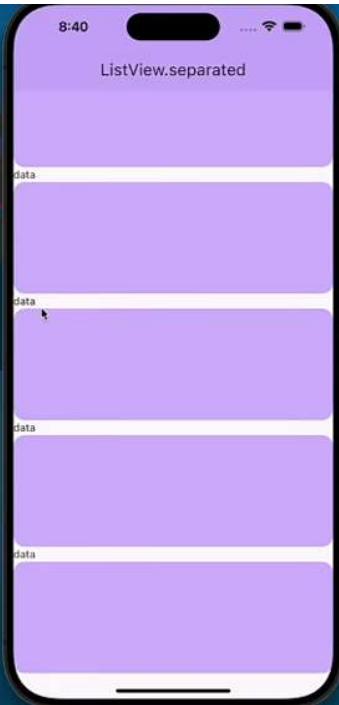
  List<String> list = ["1", "2", "3", "4", "5"];

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        centerTitle: true,
        backgroundColor: Theme.of(context).colorScheme.inversePrimary,
        title: const Text("ListView.separated"),
      ), // AppBar

      body: ListView.separated(
        itemCount: list.length,
        scrollDirection: Axis.vertical,
        separatorBuilder: (context, index)>> const SizedBox(height: 10,)||
        itemBuilder: (context, index){
          return Container(
            height: 150,
            decoration: BoxDecoration(
              color: Theme.of(context).colorScheme.inversePrimary,
              borderRadius: BorderRadius.circular(15)
            ), // BoxDecoration
          ); // Container
        }, // ListView.separated
      ); // Scaffold
    }
}

```

b) Giving text as separator



A screenshot of an iPhone X simulator running Flutter. The screen is purple and displays five white rectangular boxes arranged vertically. Each box contains the word "data" and has a height of 150 pixels and a circular border radius of 15 pixels. The background color of the list items is the inverse primary color of the theme. The title "ListView.separated" is displayed at the top of the screen.

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

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

  @override
  State<HomeScreen> createState() => _HomeScreenState();
}

class _HomeScreenState extends State<HomeScreen> {

  List<String> list = ["1", "2", "3", "4", "5"];

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        centerTitle: true,
        backgroundColor: Theme.of(context).colorScheme.inversePrimary,
        title: const Text("ListView.separated"),
      ), // AppBar

      body: ListView.separated(
        itemCount: list.length,
        scrollDirection: Axis.vertical,
        separatorBuilder: (context, index)>> Text("data"),
        itemBuilder: (context, index){
          return Container(
            height: 150,
            decoration: BoxDecoration(
              color: Theme.of(context).colorScheme.inversePrimary,
              borderRadius: BorderRadius.circular(15)
            ), // BoxDecoration
          ); // Container
        }, // ListView.separated
      ); // Scaffold
    }
}

```

Giving padding

A screenshot of a Flutter development environment. On the left is the code editor with the file `home_screen.dart` open, showing the implementation of a vertical `ListView.separated`. On the right is a mobile device emulator showing the resulting app interface. The title bar of the emulator says "ListView.separated". The screen displays five purple rectangular items arranged vertically.

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

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

  @override
  State<HomeScreen> createState() => _HomeScreenState();
}

class _HomeScreenState extends State<HomeScreen> {
  List<String> list = ["1", "2", "3", "4", "5"];

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        centerTitle: true,
        backgroundColor: Theme.of(context).colorScheme.inversePrimary,
        title: const Text("ListView.separated"),
      ), // AppBar

      body: ListView.separated(
        itemCount: list.length,
        padding: EdgeInsets.all(20),
        scrollDirection: Axis.vertical,
        separatorBuilder: (context, index)>> const SizedBox(height: 20),
        itemBuilder: (context, index){
          return Container(
            height: 150,
            decoration: BoxDecoration(
              color: Theme.of(context).colorScheme.inversePrimary,
              borderRadius: BorderRadius.circular(15)
            ), // BoxDecoration
        ); // Container
      ), // ListView.separated
    ); // Scaffold
  }
}
```

Making it horizontal list

A screenshot of a Flutter development environment. On the left is the code editor with the file `home_screen.dart` open, showing the modification to create a horizontal `ListView.separated`. On the right is a mobile device emulator showing the resulting app interface. The title bar of the emulator says "ListView.separated". The screen displays four purple rectangular items arranged horizontally.

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

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

  @override
  State<HomeScreen> createState() => _HomeScreenState();
}

class _HomeScreenState extends State<HomeScreen> {
  List<String> list = ["1", "2", "3", "4", "5"];

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        centerTitle: true,
        backgroundColor: Theme.of(context).colorScheme.inversePrimary,
        title: const Text("ListView.separated"),
      ), // AppBar

      body: SizedBox(
        height: 150,
        child: ListView.separated(
          itemCount: list.length,
          padding: const EdgeInsets.all(20),
          scrollDirection: Axis.horizontal,
          separatorBuilder: (context, index)>> const SizedBox(height: 20,width: 20),
          itemBuilder: (context, index){
            return Container(
              height: 150,width: 100,
              decoration: BoxDecoration(
                color: Theme.of(context).colorScheme.inversePrimary,
                borderRadius: BorderRadius.circular(15)
              ), // BoxDecoration
            ); // Container
          }, // ListView.separated
        ), // SizedBox
    );
  }
}
```

9. GridView.builder

The screenshot shows the Android Studio interface. On the left is the code editor with the file `home_screen.dart` open. The code defines a `Scaffold` with an `AppBar` and a `GridView.builder` body containing 20 items. On the right is the emulator window displaying the app titled "GridViews.builder" on an iPhone 15 Pro Max. The grid has 4 columns and 5 rows of purple squares.

```
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      centerTitle: true,
      backgroundColor: Theme.of(context).colorScheme.inversePrimary,
      title: const Text("GridViews.builder"),
    ), // AppBar

    body: GridView.builder(
      itemCount: 20,
      gridDelegate: SliverGridDelegateWithFixedCrossAxisCount(
        crossAxisCount: 2,
        mainAxisSpacing: 16.0,
        crossAxisSpacing: 16.0, // SliverGridDelegateWithFixedCrossAxisCount
      ),
      itemBuilder: (context, index) {
        return Container(color: Theme.of(context).colorScheme.inversePrimary);
      }, // GridView.builder
    ), // Scaffold
  );
}
```

Giving radius

The screenshot shows the Android Studio interface with the same code as before, but now the `Container` in the `itemBuilder` has a `BoxDecoration` with a `borderRadius: BorderRadius.circular(15)`. A red arrow points to this line. The emulator window shows the app running on an iPhone 15 Pro Max, where each grid item now has rounded corners.

```
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      centerTitle: true,
      backgroundColor: Theme.of(context).colorScheme.inversePrimary,
      title: const Text("GridViews.builder"),
    ), // AppBar

    body: GridView.builder(
      itemCount: 20,
      padding: EdgeInsets.all(15),
      gridDelegate: SliverGridDelegateWithFixedCrossAxisCount(
        crossAxisCount: 2,
        mainAxisSpacing: 16.0,
        crossAxisSpacing: 16.0,
        childAspectRatio: 1/1.5
      ), // SliverGridDelegateWithFixedCrossAxisCount
      itemBuilder: (context, index) {
        return Container(
          decoration: BoxDecoration(
            color: Theme.of(context).colorScheme.inversePrimary,
            borderRadius: BorderRadius.circular(15)
          ), // BoxDecoration
        ); // Container
      }, // GridView.builder
    ), // Scaffold
  );
}
```

10. Connectivity_plus

connectivity_plus 4.0.2

Published 51 days ago • ⚭ fluttercommunity.dev (Dart 3 compatible)

PLATFORM | ANDROID | IOS | LINUX | MACOS | WEB | WINDOWS

2.5K

Readme Changelog Example Installing Versions Scores

Use this package as a library

Depend on it

Run this command:

With Flutter:

```
$ flutter pub add connectivity_plus
```

Making function

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

Future<bool> isNetworkAvailable() async{
  var result = await Connectivity().checkConnectivity();

  return result == ConnectivityResult.none ? false : true;
}
```

A code editor window showing a file named `network_connectivity.dart`. The code imports the `connectivity_plus` package and defines a `isNetworkAvailable` function. A tooltip from the IDE shows the enum `ConnectivityResult` with values: none, mobile, bluetooth, ethernet, other, vpn, and wifi.

Calling

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

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        backgroundColor: Colors.blueGrey,
        title: const Text("Network Connectivity", style: TextStyle(color: Colors.white)),
      ), // AppBar

      body: Center(
        child: ElevatedButton(
          onPressed: () async{
            if(await isNetworkAvailable()){
              print("Network Available");
            }else{
              print("Network Not Available");
            }
          },
          child: const Text("Check Network"), // ElevatedButton
        ), // Center
      ), // Scaffold
    );
  }
}
```

A code editor window showing a file named `home_screen.dart`. It contains a `ElevatedButton` that calls the `isNetworkAvailable` function defined in the previous screenshot. A red arrow points from the `onPressed` block to the `isNetworkAvailable` call.

11. Webview

webview_flutter 4.2.4

Published 23 days ago • ⚡ flutter.dev (Bart 3 compatible)

SDK FLUTTER PLATFORM ANDROID iOS 3.4K

Readme Changelog Example Installing Versions Scores

WebView for Flutter

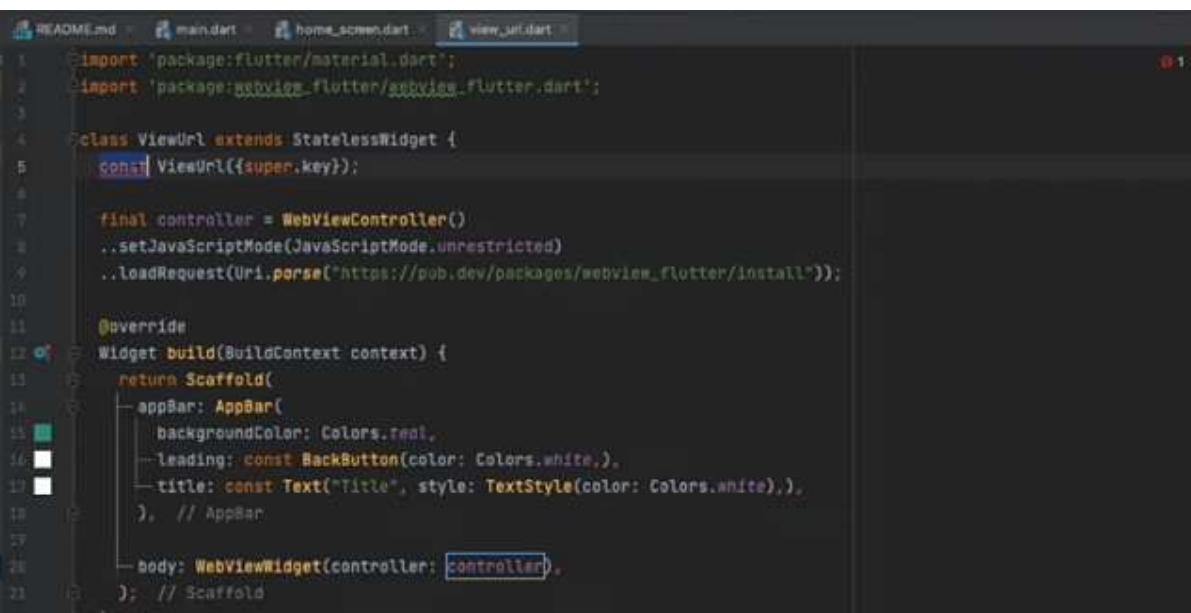
pub v4.2.4

A Flutter plugin that provides a WebView widget.

On iOS the WebView widget is backed by a WKWebView. On Android the WebView widget is backed by a WebView.

Android iOS

Support SDK 19+ or 20+ 11.0+



Function

```
1  README.md  2  main.dart  3  home_screen.dart  4  view_url.dart
5
6  import 'package:flutter/material.dart';
7  import 'package:webview_flutter/webview_flutter.dart';
8
9
10 class ViewUrl extends StatelessWidget {
11   const ViewUrl({super.key});
12
13   final controller = WebViewController()
14     ..setJavaScriptMode(JavaScriptMode.unrestricted)
15     ..loadRequest(Uri.parse("https://pub.dev/packages/webview_flutter/install"));
16
17   @override
18   Widget build(BuildContext context) {
19     return Scaffold(
20       appBar: AppBar(
21         backgroundColor: Colors.red,
22         leading: const BackButton(color: Colors.white,),
23         title: const Text("Title", style: TextStyle(color: Colors.white),),
24       ), // AppBar
25
26       body: WebViewWidget(controller: controller),
27     ); // Scaffold
28 }
```

Output

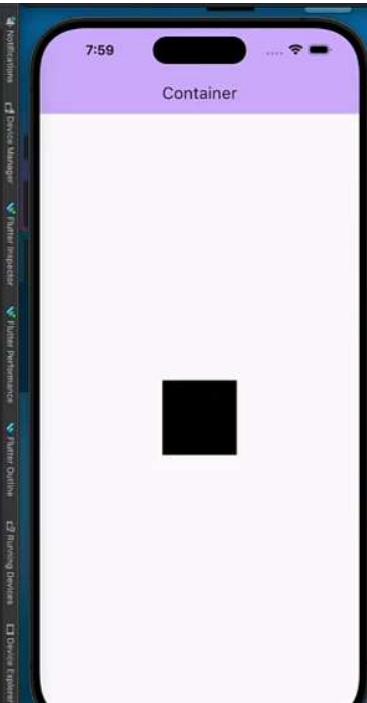
The screenshot shows a development environment with two main panes. On the left, the code editor displays the Dart code for the `HomeScreen` widget:

```
1 import 'package:flutter/material.dart';
2 import 'package:webview_in_flutter_app/view_url.dart';
3
4 class HomeScreen extends StatelessWidget {
5   const HomeScreen({super.key});
6
7   @override
8   Widget build(BuildContext context) {
9     return Scaffold(
10       appBar: AppBar(
11         backgroundColor: Colors.teal,
12         title: const Text("Webview", style: TextStyle(color: Colors.white)),
13       ), // AppBar
14
15       body: Center(
16         child: ElevatedButton(
17           style: ElevatedButton.styleFrom(
18             backgroundColor: Colors.teal,
19           ),
20           onPressed: () {
21             Navigator.push(context, MaterialPageRoute(builder: (context) => ViewUrl()));
22           },
23           child: const Text("Open URL", style: TextStyle(color: Colors.white)), // ElevatedButton
24         ), // Center
25       ), // Scaffold
26     );
27   }
28 }
```

On the right, a mobile device screen displays the `pub.dev` website for the `webview_flutter` package. The device status bar shows "11:06". The browser title is "Title". The page content includes:

- webview_flutter 4.2.4**
- Published 23 days ago • © flutter.dev
- Dart 3 compatible
- SDK | FLUTTER | 3.4K
- PLATFORM | ANDROID | IOS
- Metadata**
- A Flutter plugin that provides a WebView widget on Android and iOS.
- More...
- Readme | Changelog | Example
- Installing** (underlined)
- Versions | Scores
- This site uses cookies from Google to deliver and enhance the quality of its services and to analyze traffic.
- Learn more | OK, got it

12. Container



A screenshot of a Flutter application running on an iPhone. The title bar says "Container". The main content area contains a single black square centered on the screen.

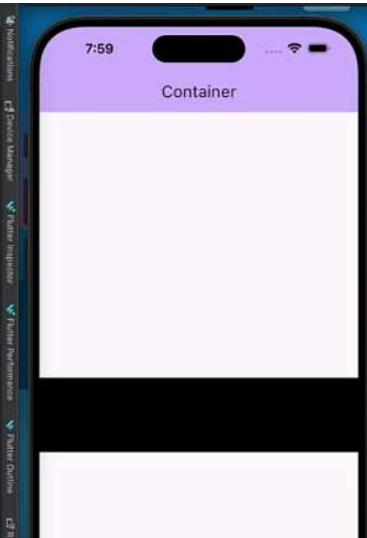
```
import 'package:flutter/material.dart';

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

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        centerTitle: true,
        backgroundColor: Theme.of(context).colorScheme.inversePrimary,
        title: const Text("Container"),
      ), // AppBar

      body: Center(
        child: Container(
          height: 100,
          width: 100,
          color: Colors.black,
        ), // Container
      ), // Center
    ); // Scaffold
  }
}
```

Double.infinity



A screenshot of a Flutter application running on an iPhone. The title bar says "Container". The main content area consists of three horizontal bars: a thin black bar at the top, a thick black bar in the middle, and a thin white bar at the bottom.

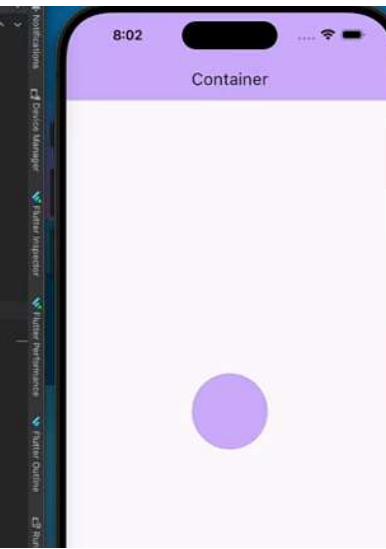
```
import 'package:flutter/material.dart';

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

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        centerTitle: true,
        backgroundColor: Theme.of(context).colorScheme.inversePrimary,
        title: const Text("Container"),
      ), // AppBar

      body: Center(
        child: Container(
          height: 100,
          width: double.infinity,
          color: Colors.black,
        ), // Container
      ), // Center
    ); // Scaffold
  }
}
```

shape



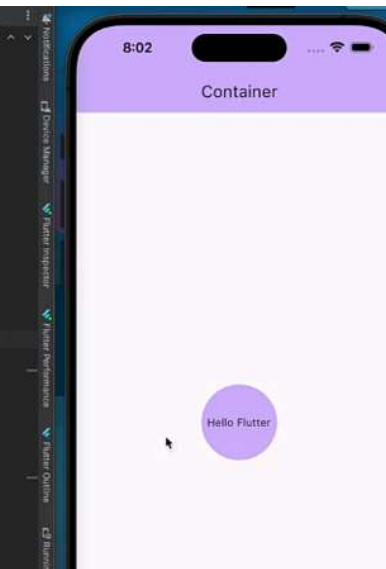
```
import 'package:flutter/material.dart';

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

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        centerTitle: true,
        backgroundColor: Theme.of(context).colorScheme.inversePrimary,
        title: const Text("Container"),
      ), // AppBar

      body: Center(
        child: Container(
          height: 100,
          margin: EdgeInsets.symmetric(horizontal: 20),
          width: 100,
          decoration: BoxDecoration(
            shape: BoxShape.circle,
            color: Theme.of(context).colorScheme.inversePrimary,
          ), // BoxDecoration
        ), // Container
      ), // Center
    ); // Scaffold
  }
}
```

Text in shape



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

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

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        centerTitle: true,
        backgroundColor: Theme.of(context).colorScheme.inversePrimary,
        title: const Text("Container"),
      ), // AppBar

      body: Center(
        child: Container(
          height: 100,
          alignment: Alignment.center,
          margin: EdgeInsets.symmetric(horizontal: 20),
          width: 100,
          decoration: BoxDecoration(
            shape: BoxShape.circle,
            color: Theme.of(context).colorScheme.inversePrimary,
          ), // BoxDecoration
          child: Text("Hello Flutter"),
        ), // Container
      ), // Center
    ); // Scaffold
  }
}
```

Only one side borderRadius



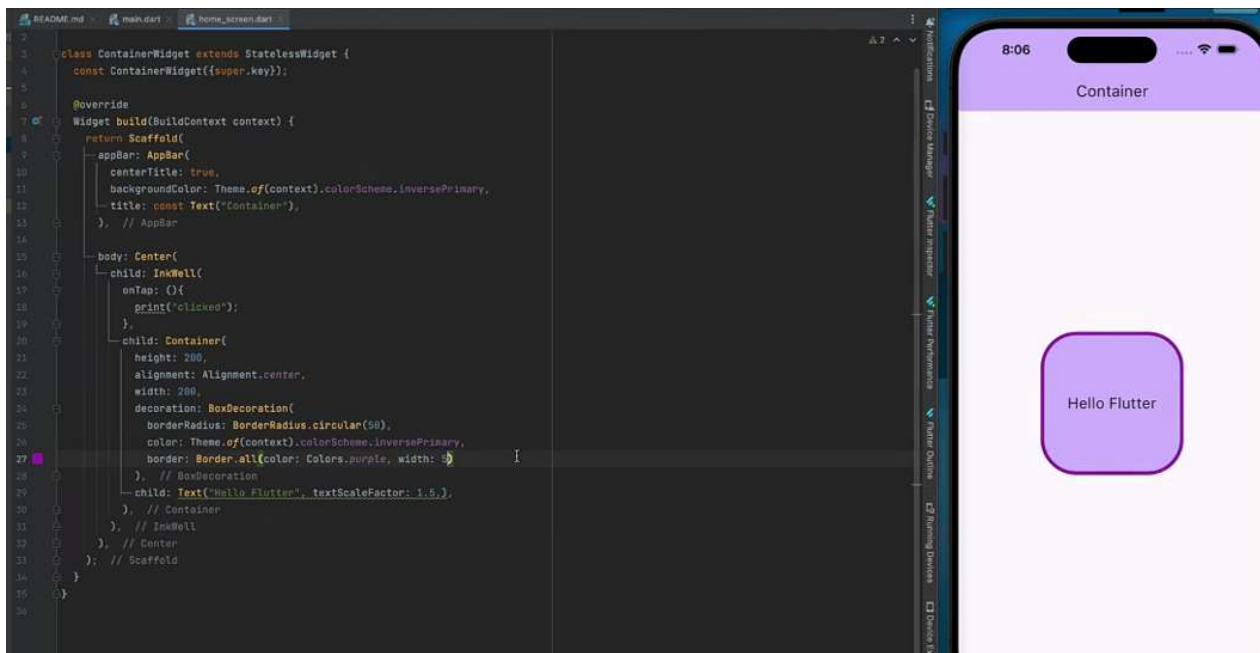
```
import 'package:flutter/material.dart';

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

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        centerTitle: true,
        backgroundColor: Theme.of(context).colorScheme.inversePrimary,
        title: const Text("Container"),
      ), // AppBar

      body: Center(
        child: Container(
          height: 100,
          alignment: Alignment.center,
          margin: EdgeInsets.symmetric(horizontal: 20),
          width: 100,
          decoration: BoxDecoration(
            borderRadius: BorderRadius.only(topLeft: Radius.circular(20)),
            color: Theme.of(context).colorScheme.inversePrimary,
          ), // BoxDecoration
          child: Text("Hello Flutter"),
        ), // Container
      ), // Center
    ); // Scaffold
  }
}
```

Wrapping with InkWell to make it tapable



The image shows a developer's workspace with two main components: a code editor and a mobile application preview.

Code Editor: On the left, the code for `home_screen.dart` is displayed in a dark-themed code editor. The code defines a `ContainerWidget` class that extends `StatelessWidget`. It sets up a `Scaffold` with a purple `AppBar` and a centered `Container` with a purple rounded rectangle and white text. An `InkWell` widget is used to wrap the inner `Container`, which contains the text "Hello Flutter". The `onTap` event of the `InkWell` triggers a print statement.

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

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        centerTitle: true,
        backgroundColor: Theme.of(context).colorScheme.inversePrimary,
        title: const Text("Container"),
      ), // AppBar

      body: Center(
        child: InkWell(
          onTap: () {
            print("clicked");
          },
          child: Container(
            height: 200,
            alignment: Alignment.center,
            width: 200,
            decoration: BoxDecoration(
              borderRadius: BorderRadius.circular(50),
              color: Theme.of(context).colorScheme.inversePrimary,
              border: Border.all(color: Colors.purple, width: 5),
            ), // BoxDecoration
            child: Text("Hello Flutter", textScaleFactor: 1.5),
          ), // Container
        ), // InkWell
      ), // Center
    ); // Scaffold
  }
}
```

Mobile Application Preview: On the right, a screenshot of an iPhone X simulator is shown. The screen displays a purple rounded rectangular button with the text "Hello Flutter" in white. The status bar at the top shows the time as 8:06. The title bar of the simulator window says "Container".

11. clipboard

clipboard 0.1.3

Published 3 years ago • samuelezedi.com Dart 3 compatible

SDK FLUTTER PLATFORM ANDROID IOS LINUX MACOS WEB WINDOWS 1,611 544

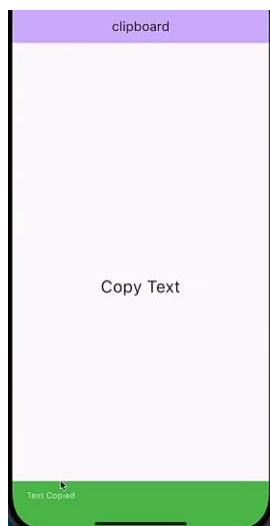
Readme Changelog Example Installing Versions Scores

clipboard

```
class _HomeScreenState extends State<HomeScreen> {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        centerTitle: true,
        backgroundColor: Theme.of(context).colorScheme.inversePrimary,
        title: const Text("clipboard"),
      ), // AppBar

      body: Center(
        child: GestureDetector(
          onLongPress: () {
            FlutterClipboard.copy("Copy Text").then((value) {
              ScaffoldMessenger.of(context).showSnackBar(const SnackBar(content: Text('Text Copied'), backgroundColor: Colors.green,));
            });
          },
          child: const Text('Copy Text', textScaleFactor: 2.0), // GestureDetector
        ), // Center
      ), // scaffold
    );
  }
}
```

Output: showing message with snackbar



13. Hexcolor

hexcolor 3.0.1 

Published 18 months ago · Dart 3 compatible

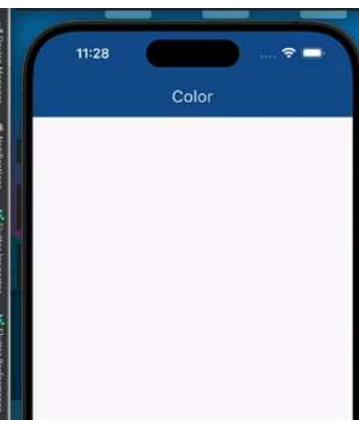
SDK FLUTTER PLATFORM ANDROID IOS LINUX MACOS WEB WINDOWS 306

Readme Changelog Installing Versions Scores 306 130 99% LIKES PUB POINTS POPULARITY

licence BSD 3-Clause pub.dev 211 Publisher unverified uploader

Metadata

hexcolor_example



```
main.dart
1: import 'package:flutter/material.dart';
2: import 'package:hexcolor/hexcolor.dart';
3:
4: class HomeScreen extends StatefulWidget {
5:   const HomeScreen({Key key}) : super(key: key);
6:
7:   @override
8:   State<HomeScreen> createState() => _HomeScreenState();
9: }
10:
11: class _HomeScreenState extends State<HomeScreen> {
12:   @override
13:   Widget build(BuildContext context) {
14:     return Scaffold(
15:       appBar: AppBar(
16:         centerTitle: true,
17:         backgroundColor: HexColor("#0c5298"),
18:         title: Text("Color", style: TextStyle(color: HexColor("#ff00ff"))),
19:       ),
20:     );
21:   }
22: }
```

14. SharedPreferences (like localStorage or AsyncStorage)

shared_preferences 2.2.2  copied to clipboard 

Published 23 days ago · flutter.dev Dart 3 compatible

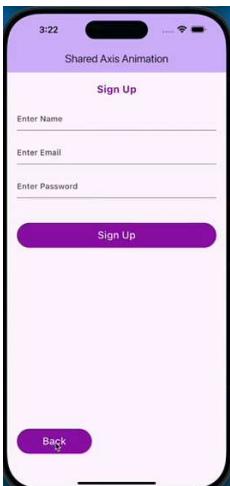
SDK FLUTTER PLATFORM ANDROID IOS LINUX MACOS WEB WINDOWS 8.1K

Readme Changelog Example Installing Versions Scores 8189 140 100% LIKES PUB POINTS POPULARITY

Shared preferences plugin

```
1 README.md < main.dart < home_screen.dart < app_store.dart < view.dart < pubspec.yaml
2
3 import 'package:shared_preferences/shared_preferences.dart';
4
5
6 class AppStore{
7
8   setFullName(String fullName)async{
9     SharedPreferences sp = await SharedPreferences.getInstance();
10    sp.setString("full_name", fullName);
11  }
12
13   Future<String> getFullName()async{
14     SharedPreferences sp = await SharedPreferences.getInstance();
15     final String? fullName = sp.getString("full_name");
16     return fullName??"";
17   }
18
19   removeFullName(String fullName)async{
20     SharedPreferences sp = await SharedPreferences.getInstance();
21     sp.remove("full_name");
22   }
23
24 }
```

Output



A screenshot of a developer's environment showing code and a running application. On the left is a code editor with Dart code for a "Store Data" screen. The code includes imports, a state management provider `AppStore`, and UI components like `TextEditingController`, `Scaffold`, `AppBar`, `Column`, `TextFormField`, `ElevatedButton` for "Submit" and "View", and `const SizedBox`. On the right is a smartphone displaying the "Store Data" screen with a purple header. The screen shows a text input field labeled "Full Name", a purple "Submit" button, and a purple "View" button.

• Animations

The screenshot shows the pub.dev package page for 'animations' version 2.0.11. At the top, there's a navigation bar with a search icon, 'Sign in', and 'Help'. Below the header, the package name 'animations' is displayed in a large blue box along with its version '2.0.11'. A small 'View on GitHub' icon is next to it. Below this, a status bar indicates 'Published 10 months ago' and 'Dart 3 compatible'. A horizontal bar shows supported platforms: 'SDK', 'FLUTTER' (highlighted in blue), 'PLATFORM', 'ANDROID', 'IOS', 'LINUX', 'MACOS', 'WEB', and 'WINDOWS'. To the right of the platform bar is a '6.2K' rating with a thumbs-up icon. Further down, a navigation menu includes 'Readme' (underlined in blue), 'Changelog', 'Example', 'Installing', 'Versions', and 'Scores'. On the far right, popularity metrics are shown: '6270 LIKES', '160 PUB POINTS', and '100% POPULARITY'.

Using

A code editor window displays a Dart file containing Flutter code. The code defines a `ListView` with a single child. This child is an `OpenContainer` widget. Inside the `OpenContainer`, the `closedBuilder` function returns a `Container` with a height of 220, a `BoxDecoration` with a `NetworkImage` as the `image`, and a `BoxFit.cover` fit. The `openBuilder` function returns a `const DetailsScreen` with an `imagesUrl`. The `ListView` also contains a `SizedBox`, a `Container`, and a `ListView.separated`. The entire code block is enclosed in a `],` and a final `),`.

```
body: ListView(
  children: [
    OpenContainer(
      closedBuilder: (context, container){
        return Container(
          height: 220,
          decoration: const BoxDecoration(
            image: DecorationImage(
              fit: BoxFit.cover,
              image: NetworkImage("https://blogimage.vantagecircle.com/content/images/2026/0
        ), // BoxDecoration
      ); // Container
    },
    openBuilder: (context, openContainer ){
      return const DetailsScreen(imagesUrl: 'https://blogimage.vantagecircle.com/content/images/2026/0
    }, // OpenContainer

    SizedBox(...), // SizedBox

    Container(...), // Container

    ListView.separated(...), // ListView.separated
  ],
), // ListView

floatingActionButton: Container(
  padding: const EdgeInsets.all(20),
  decoration: const BoxDecoration(
    shape: BoxShape.circle,
    color: Colors.purple
  ), // BoxDecoration
  child: const Icon(Icons.add, color: Colors.white,),) // Container
```

Output



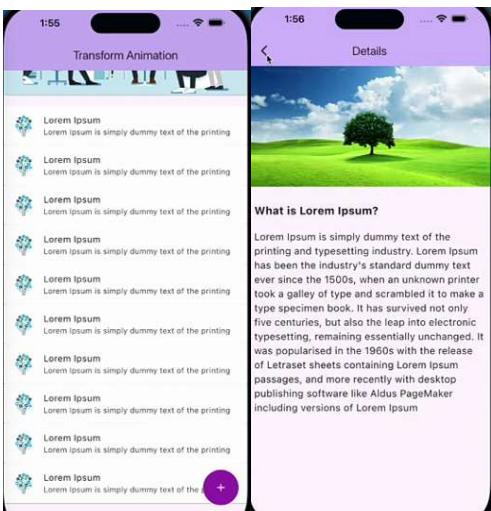
Eg2

```

floatingActionButton: OpenContainer(
  closedShape: RoundedRectangleBorder(
    borderRadius: BorderRadius.circular(100),
  ), // RoundedRectangleBorder
  closedBuilder: (context, container){
    return Container(
      padding: const EdgeInsets.all(20),
      decoration: const BoxDecoration(
        shape: BoxShape.circle,
        color: Colors.purple
      ), // BoxDecoration
      child: const Icon(Icons.add, color: Colors.white,), // Container
    ),
  openBuilder: (context, openContainer ){
    return const DetailsScreen(imagesUrl: 'https://images.ctfassets.net/hrltx12pl8hq/28ECAQipJZ7t
  }, // OpenContainer
); // Scaffold
}

```

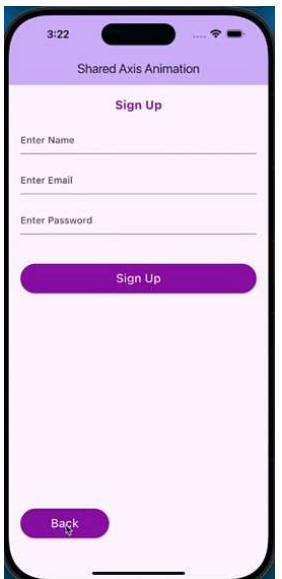
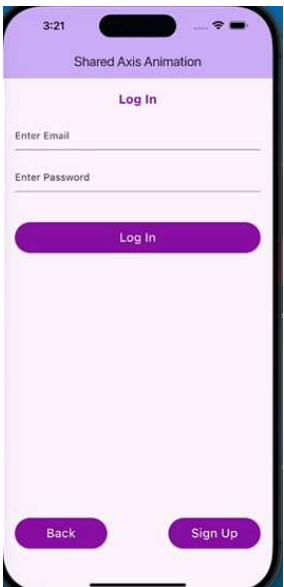
Output: press button and open DetailsScreen



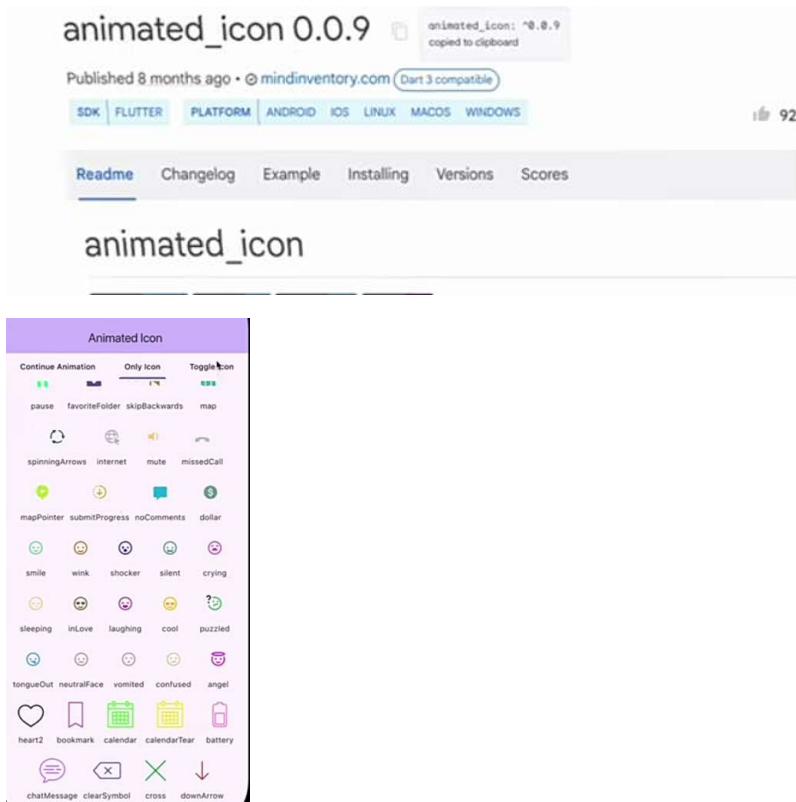
Transitional animation

```
body: Padding(
  padding: const EdgeInsets.all(20),
  child: Column(
    children: [
      PageTransitionSwitcher(
        reverse: isLoginPage,
        transitionBuilder: (Widget child, Animation<double> animation, Animation<double> secondaryAnimation) {
          return SharedAxisTransition(
            animation: animation,
            secondaryAnimation: secondaryAnimation,
            transitionType: SharedAxisTransitionType.horizontal,
            child: child,
          );
        },
        child: isLoginPage ? const LogInScreen() : const SignUpScreen(),
      ), // PageTransitionSwitcher
      const Spacer(),
    ],
  ),
Row(
  mainAxisAlignment: MainAxisAlignment.spaceBetween,
  children: [
    SizedBox(
      height: 50,
      width: 150,
      child: ElevatedButton(
        style: ElevatedButton.styleFrom(backgroundColor: Colors.purple, foregroundColor: Colors.white),
        onPressed: () { setState(() { isLoginPage = true; }); },
        child: const Text('Back', style: TextStyle(fontSize: 20)),
      ),
    ), // SizedBox
    SizedBox(
      height: 50,
      width: 150,
      child: ElevatedButton(
        style: ElevatedButton.styleFrom(backgroundColor: Colors.purple, foregroundColor: Colors.white),
        onPressed: () { setState(() { isLoginPage = false; }); },
        child: const Text('Sign Up', style: TextStyle(fontSize: 20)),
      ),
    ), // SizedBox
  ],
), // Row
const SizedBox(height: 50),
Row(
  mainAxisAlignment: MainAxisAlignment.spaceBetween,
  children: [
    Visibility(
      visible: !isLoginPage,
      child: SizedBox(
        height: 50,
        width: 150,
        child: ElevatedButton(
          style: ElevatedButton.styleFrom(backgroundColor: Colors.purple, foregroundColor: Colors.white),
          onPressed: () { setState(() { isLoginPage = true; }); },
          child: const Text('Back', style: TextStyle(fontSize: 20)),
        ),
      ),
    ), // SizedBox
    Visibility(
      visible: isLoginPage,
      child: SizedBox(
        height: 50,
        width: 150,
        child: ElevatedButton(
          style: ElevatedButton.styleFrom(backgroundColor: Colors.purple, foregroundColor: Colors.white),
          onPressed: () { setState(() { isLoginPage = false; }); },
          child: const Text('Sign Up', style: TextStyle(fontSize: 20)),
        ),
      ),
    ), // Visibility
  ],
), // Row
```

Output



• Animated Icons



Using

The screenshot shows the Android Studio code editor with two files open. The top file is `_HomeScreenState.dart` and the bottom file is `main.dart`. Both files contain Dart code for a Flutter application. In the `_HomeScreenState.dart` file, there is a line of code:

```
animateIcon: AnimateIcons.bell,
```

 which is highlighted with a red arrow. The `main.dart` file has a similar line:

```
animateIcon: AnimateIcons.settings,
```

. The code is part of a `Scaffold` widget's body, which contains a `Center` widget with an `AnimateIcon` child.

```
class _HomeScreenState extends State<HomeScreen> {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        backgroundColor: Theme.of(context).colorScheme.inversePrimary,
        title: const Text("Animated Icon"), // AppBar
      ),
      body: Center(
        child: AnimateIcon(
          key: UniqueKey(),
          onTap: () {},
          iconType: IconType.continueAnimation,
          height: 70,
          width: 70,
          color: Color.fromRGBO(
            Random.secure().nextInt(255),
            Random.secure().nextInt(255),
            Random.secure().nextInt(255),
            1), // Color.fromRGBO
          animateIcon: AnimateIcons.bell,
        ), // AnimateIcon
      ), // Center
    ); // Scaffold
  }
}

class _HomeScreenState extends State<HomeScreen> {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        backgroundColor: Theme.of(context).colorScheme.inversePrimary,
        title: const Text("Animated Icon"), // AppBar
      ),
      body: Center(
        child: AnimateIcon(
          key: UniqueKey(),
          onTap: () {},
          iconType: IconType.continueAnimation,
          height: 70,
          width: 70,
          color: Colors.red,
          animateIcon: AnimateIcons.settings,
        ), // AnimateIcon
      ), // Center
    ); // Scaffold
  }
}
```

• Confette

confetti 0.8.0 

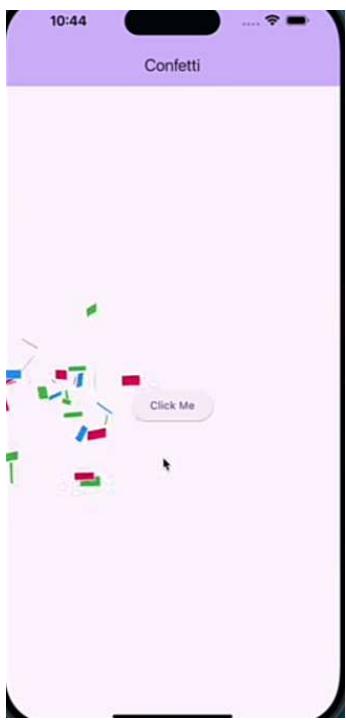
Published 28 days ago • ⚡ funwith.app Dart 3 compatible

SDK FLUTTER PLATFORM ANDROID IOS LINUX MACOS WEB WINDOWS 1.3K

Readme Changelog Example Installing Versions Scores 1316 LIKES 160 PUB POINTS 99% POPULARITY

```
2 import 'package:flutter/material.dart';
3
4
5 class HomeScreen extends StatefulWidget {
6     const HomeScreen({super.key});
7
8     @override
9     State<HomeScreen> createState() => _HomeScreenState();
10 }
11
12 class _HomeScreenState extends State<HomeScreen> {
13
14     late ConfettiController _controllerCenter;
15
16     @override
17     void initState() {
18         _controllerCenter =
19             ConfettiController(duration: const Duration(seconds: 2));
20         super.initState();
21     }
22     @override
23     Widget build(BuildContext context) {
24         return Scaffold(
25             appBar: AppBar(
26                 appBar: AppBar(
27                     backgroundColor: Theme.of(context).colorScheme.inversePrimary,
28                     title: const Text('Confetti'),
29                 ), // AppBar
30
31                 body: Stack(
32                     children: [
33
34                         Align(
35                             alignment: Alignment.center,
36                             child: ConfettiWidget(
37                                 confettiController: _controllerCenter,
38                                 blastDirection: pi, // radial value - LEFT
39                                 particleDrag: 0.05, // apply drag to the confetti
40                                 emissionFrequency: 0.05, // how often it should emit
41                                 numberOfParticles: 20, // number of particles to emit
42                                 gravity: 0.05, // gravity - or fall speed
43                                 shouldLoop: false,
44                                 colors: const [
45                                     Colors.green,
46                                     Colors.blue,
47                                     Colors.pink
48                                 ], // manually specify the colors to be used
49                                 strokeWidth: 1,
50                                 strokeColor: Colors.white,
51                             ), // ConfettiWidget
52                         ), // Align
53
54                         Center(child: ElevatedButton(onPressed: (){}, child: Text('Click Me'))),
55
56                     ],
57                 ),
58             ),
59         );
60     }
61 }
```

Output



• Flutter Launcher icon (App icon)

flutter_launcher_icons 0.13.1

Published 6 months ago • @ fluttercommunity.dev Dart 3 compatible

SDK DART FLUTTER PLATFORM ANDROID IOS LINUX MACOS WINDOWS

6.0K

Readme Changelog Example Installing Versions Scores

6004 130 100%

Likes Pub Points Popularity

Flutter Launcher Icons

Flutter Community

flutter_launcher_icons

Publisher

@ fluttercommunity.dev

Metadata

Guide

1. Setup the config file

Add your Flutter Launcher Icons configuration to your `pubspec.yaml` or create a new config file called `flutter_launcher_icons.yaml`. An example is shown below. More complex examples can be found in the example projects.

```
dev_dependencies:  
  flutter_launcher_icons: "^\u00b7.13.1"  
  
flutter_launcher_icons:  
  android: "launcher_icon"  
  ios: true  
  image_path: "assets/icon/icon.png"  
  min_sdk_android: 21 # android min sdk min:16, default 21  
  web:  
    generate: true  
    image_path: "path/to/image.png"  
    background_color: "#hexcode"  
    theme_color: "#hexcode"  
  windows:  
    generate: true  
    image_path: "path/to/image.png"  
    icon_size: 48 # min:48, max:256, default: 48  
  macos:  
    generate: true  
    image_path: "path/to/image.png"
```

If you name your configuration file something other than `flutter_launcher_icons.yaml` or `pubspec.yaml` you will need to specify the name of the file when running the package.

```
flutter pub get  
flutter pub run flutter_launcher_icons -f <your config file name here>
```

```
# The following adds the Cupertino Icons font to your application.
# Use with the CupertinoIcons class for iOS style icons.
cupertino_icons: ^1.0.2
flutter_launcher_icons: ^0.13.1

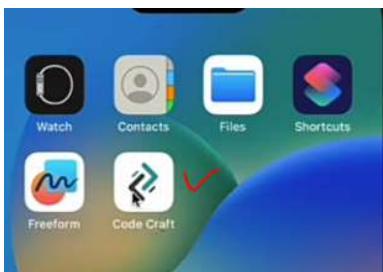
dev_dependencies:
  flutter_test:
    sdk: flutter

flutter_launcher_icons:
  android: true
  ios: true
  image_path: "assets/spp_icon.png"
```

After setting up the configuration, all that is left to do is run the package.

```
flutter pub get
flutter pub run flutter_launcher_icons
```

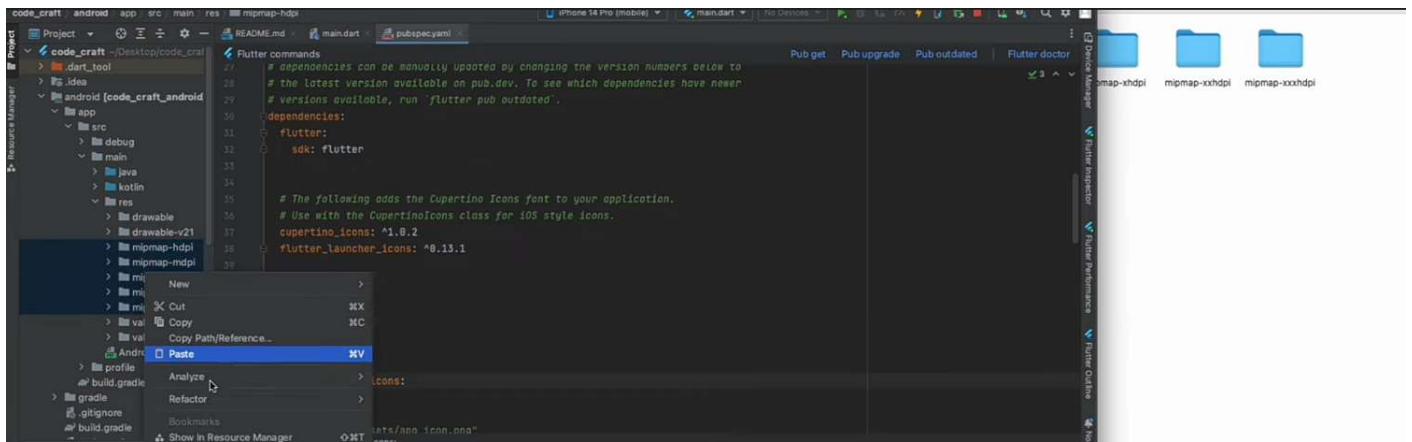
Run :



Method 2: Manullay add icon

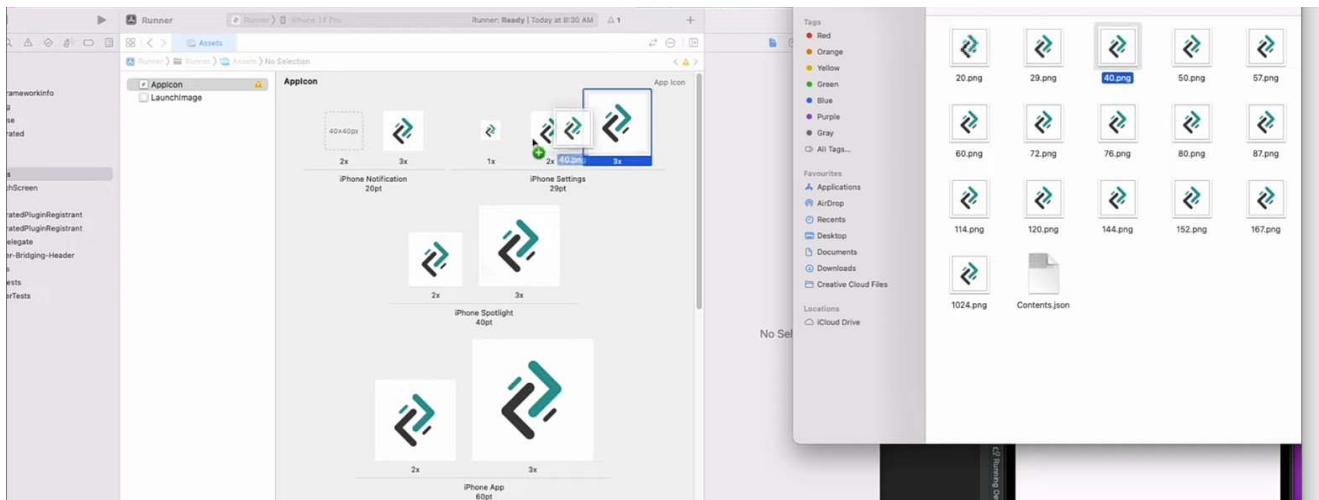
A screenshot of the 'App Icon Generator' website. The interface includes tabs for 'App Icon Generator', 'App Icon', and 'Image Sets'. On the left, there's a preview area showing a stylized 'C' and 'D' logo. To the right, there are sections for generating icons for 'iOS and macOS' (with checkboxes for iPhone, iPad, watchOS, and macOS) and 'Android' (with a checked checkbox for Android). A file input field is labeled 'File name: ic_launcher' with a note below it saying 'Change file name for all generated Android Images'. A large blue 'Generate' button is at the bottom.

For android



For ios: open wcx like file in xcode

Replace icon according to pixels



- Rename application name



You can provide app name in the following ways

Same name for all apps

```
flutter pub run rename_app:main file="My App Name"
```

Separate name for specified platform(s) and same for all others

```
flutter pub run rename_app:main android="Android Name" ios="IOS Name" others="Others Name"
```

Separate name for each platform

```
flutter pub run rename_app:main android="Android Name" ios="IOS Name" web="Web Name" mac="Mac Name"
```

- Flutter toast (similar style like android toast)

fluttertoast 8.2.2

Published 5 months ago • @karthikponnam.dev Dart 3 compatible

SDK FLUTTER PLATFORM ANDROID IOS WEB

3.1K

Readme Changelog Example Installing Versions Scores

3197 140 100%
LIKES PUB POINTS POPULARITY

Publisher

@karthikponnam.dev

Common toast class

The screenshot shows a code editor with a file named 'common_toast.dart'. The code defines a function 'commonToast' that takes a string message and returns a 'Fluttertoast.showToast' call with specific parameters:

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

commonToast(String message){
  return Fluttertoast.showToast(
    msg: message,
    toastLength: Toast.LENGTH_LONG,
    gravity: ToastGravity.CENTER,
    timeInSecForIosWeb: 1,
    fontSize: 16.0
  );
}
```

Using and output

The screenshot shows a code editor on the left displaying Dart code for a HomeScreen widget. The code imports flutter/material.dart and common_toast.dart, and defines a HomeScreen class that extends StatelessWidget. It contains an AppBar with the title "Toast Message" and a centerTitle. Below the AppBar is a Center widget containing an ElevatedButton with the onPressed callback commonToast("Flutter Toast") and a child Text("Show Toast"). On the right, a mobile application is running on a Pixel 7 Pro device. The app has a purple header bar with the text "Toast Message". The main screen shows a white button labeled "Show Toast". At the bottom, there is a blue circular icon with a white arrow pointing right, labeled "Flutter Toast".

```
1 import 'package:flutter/material.dart';
2 import 'package:toast_message/common_toast.dart';
3
4 class HomeScreen extends StatelessWidget {
5   const HomeScreen({super.key});
6
7   @override
8   Widget build(BuildContext context) {
9     return Scaffold(
10       appBar: AppBar(
11         backgroundColor: Theme.of(context).colorScheme.inversePrimary,
12         title: const Text("Toast Message"),
13         centerTitle: true,
14       ), // AppBar
15
16       body: Center(
17         child: ElevatedButton(
18           onPressed: () {
19             commonToast("Flutter Toast");
20           },
21           child: const Text("Show Toast"), // ElevatedButton
22         ), // Center
23     ); // Scaffold
24   }
25 }
```

• Image picker

The screenshot shows the image_picker package page on pub.dev. The package version is 1.0.4, published 2 months ago, and is Dart 3 compatible. It has 5.9K GitHub stars. The package is available for SDK, FLUTTER, PLATFORM, ANDROID, IOS, LINUX, MACOS, WEB, and WINDOWS. The Readme tab is selected. Other tabs include Changelog, Example, Installing, Versions, and Scores. The page also shows 5954 likes, 140 pub points, and 100% popularity.

image_picker 1.0.4 copied to clipboard

Published 2 months ago • [flutter.dev](#) Dart 3 compatible

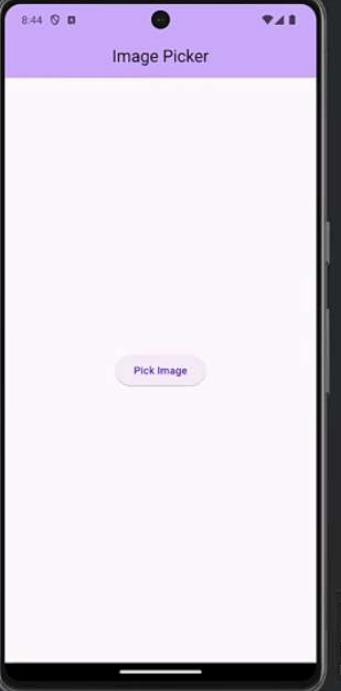
SDK | FLUTTER | PLATFORM | ANDROID | IOS | LINUX | MACOS | WEB | WINDOWS

Readme Changelog Example Installing Versions Scores

5.9K

5954 | 140 | 100%

Likes | Pub Points | Popularity



```
dynamic selectedImage;
```

```
@override
```

```
Widget build(BuildContext context) {
```

```
    return Scaffold(
```

```
        appBar: AppBar(
```

```
            centerTitle: true,
```

```
            backgroundColor: Theme.of(context).colorScheme.inversePrimary,
```

```
            title: const Text("Image Picker"),
```

```
        ), // AppBar
```

```
        body: Center(
```

```
            child: Column(
```

```
                mainAxisAlignment: MainAxisAlignment.center,
```

```
                children: [
```

```
                    selectedImage == null ? SizedBox():
```

```
                    Container(
```

```
                        height: 200,
```

```
                        width: double.infinity,
```

```
                        margin: const EdgeInsets.all(20),
```

```
                        decoration: BoxDecoration(
```

```
                            color: Colors.blueGrey,
```

```
                            borderRadius: BorderRadius.circular(20),
```

```
                            image: DecorationImage(
```

```
                                fit: BoxFit.fill,
```

```
                                image: FileImage(selectedImage)) // DecorationImage
```

```
                        ), // BoxDecoration
```

```
                    ), // Container
```

```
                    ElevatedButton(
```

```
                        onPressed: ()async{
```

```
                            final picker = ImagePicker();
```

```
                            final pickImage = await picker.pickImage(source: ImageSource.gallery);
```

```
                            if(pickImage != null){
```

```
                                File pickFile = File(pickImage.path);
```

```
                                setState(() {
```

```
                                    selectedImage = pickFile;
```

```
                                });
```

```
                            }else{
```

```
                                print("Empty!");
```

```
                            }
```

```
                        },
```

```
                        child: const Text("Pick Image")) // ElevatedButton
```

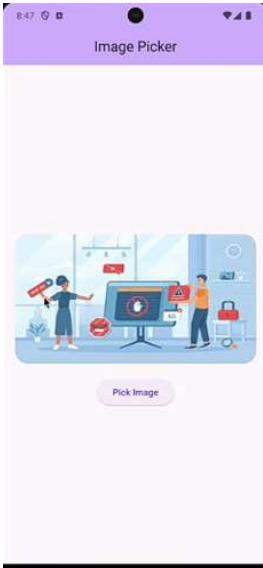
```
                ],
```

```
            ), // Column
```

```
        ), // Center
```

```
    ); // Scaffold
```

Output

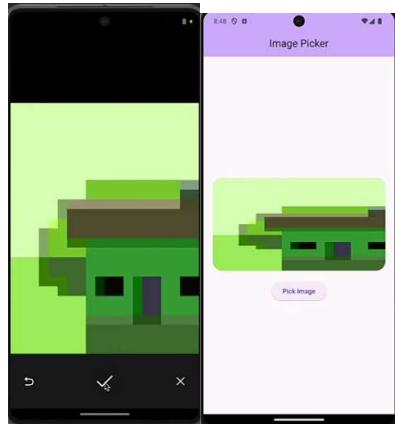


For Camera image picking:

```
    ElevatedButton(
      onPressed: ()async{
        final picker = ImagePicker();
        final pickImage = await picker.pickImage(source: ImageSource.camera);

        if(pickImage != null){
          File pickFile = File(pickImage.path);
          setState(() {
```

Output:



• DatePicker

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

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

  @override
  State<HomeScreen> createState() => _HomeScreenState();
}

class _HomeScreenState extends State<HomeScreen> {

  TextEditingController datePickerController = TextEditingController();

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        centerTitle: true,
        backgroundColor: Theme.of(context).colorScheme.inversePrimary,
        title: const Text("Date Picker"),
      ), // AppBar

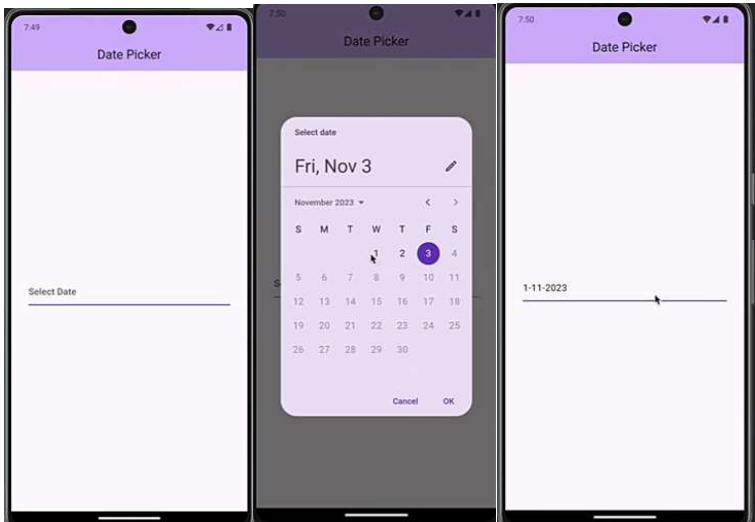
      body: Center(
        child: Padding(
          padding: const EdgeInsets.symmetric(horizontal: 30),
          child: TextField(
            readOnly: true,
            controller: datePickerController,
            decoration: const InputDecoration(hintText: "Select Date"),
          ),
        ),
      ),
    );
  }

  void _selectDate(BuildContext context) {
    datePickerController.text = '';
    setState(() {
      datePickerController.text = value;
    });
  }
}
```

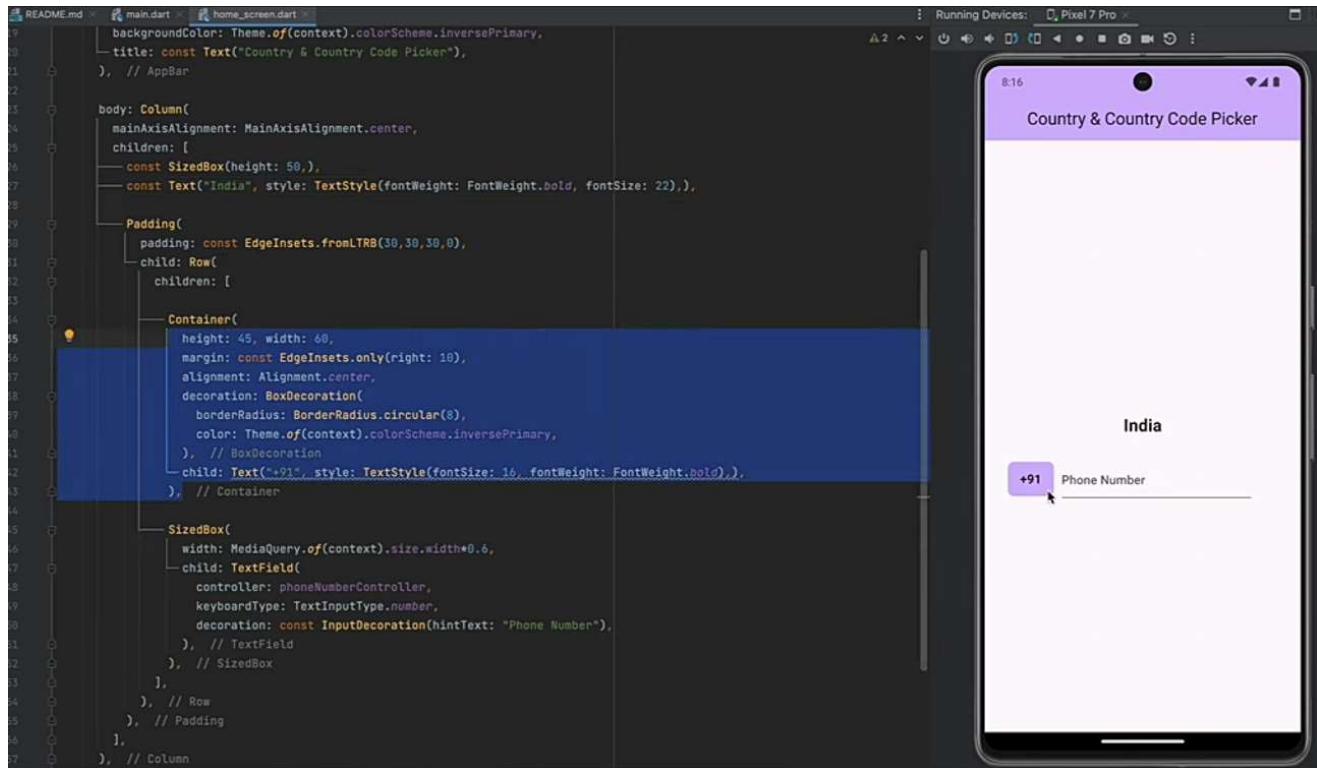
```
Future<String> _selectDate(BuildContext context)async{
  DateTime? select = await showDatePicker(
    context: context,
    initialDate: DateTime.now(),
    firstDate: DateTime(2000),
    lastDate: DateTime.now(),
  );

  return "${select!.day}-${select.month}-${select.year}";
}
```

Output



• Country picker



[fl_country_code_picker 0.1.6+1](#)

Published 24 days ago (Dart 3 compatible)

SDK FLUTTER PLATFORM ANDROID IOS LINUX MACOS WEB WINDOWS

73

[Readme](#) [Changelog](#) [Example](#) [Installing](#) [Versions](#) [Scores](#)

73 | 140 | 96%
LIKES | FAV POINTS | POPULARITY

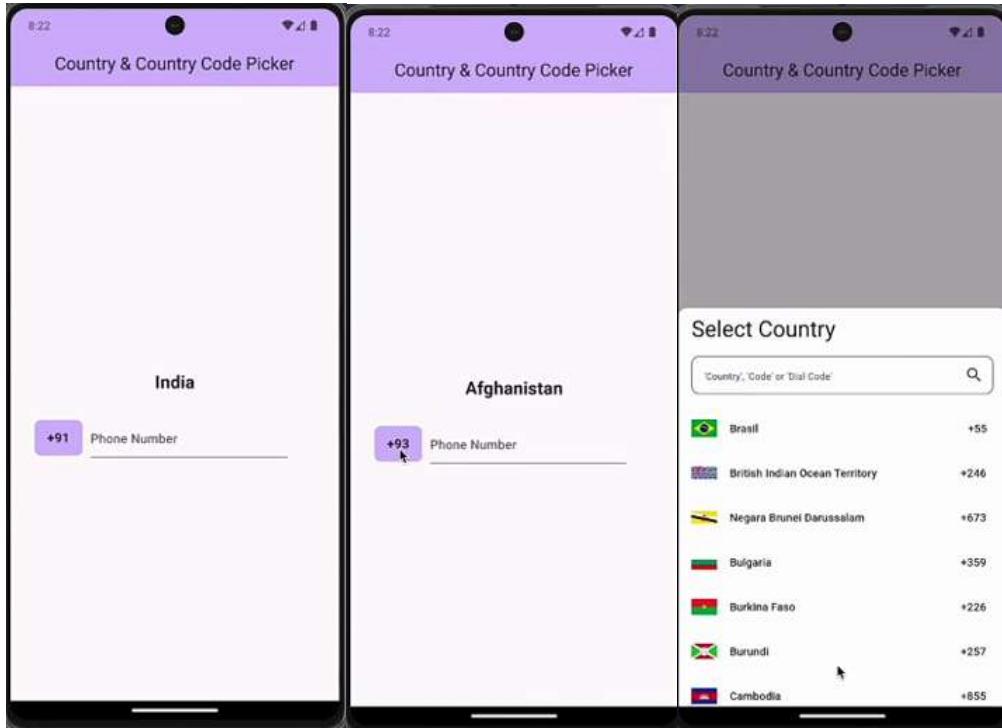
[fl_country_code_picker](#)

Publisher
unverified uploader

Using

```
15     TextEditingController phoneNumberController = TextEditingController();
16
17     String countryCode = "+91";
18     String countryName = "India";
19
20     @override
21     Widget build(BuildContext context) {
22         return Scaffold(
23             appBar: AppBar(
24                 centerTitle: true,
25                 backgroundColor: Theme.of(context).colorScheme.inversePrimary,
26                 title: const Text('Country & Country Code Picker'),
27             ), // AppBar
28
29             body: Column(
30                 mainAxisAlignment: MainAxisAlignment.center,
31                 children: [
32                     const SizedBox(height: 50),
33                     Text(countryName, style: TextStyle(fontWeight: FontWeight.bold, fontSize: 22)),
34
35                     Padding(
36                         padding: const EdgeInsets.fromLTRB(30, 30, 30, 0),
37                         child: Row(
38                             children: [
39
40                                 GestureDetector(
41                                     onTap: () async {
42                                         final picker = await pickerController.showPicker(context: context);
43                                         setState(() {
44                                             countryCode = picker?.dialCode.toString() ?? countryCode;
45                                             countryName = picker?.name.toString() ?? countryCode;
46                                         });
47                                     },
48                                     child: Container(
49                                         height: 45, width: 60,
50                                         margin: const EdgeInsets.only(right: 10),
51                                         alignment: Alignment.center,
52                                         decoration: BoxDecoration(
53                                             borderRadius: BorderRadius.circular(8),
54                                             color: Theme.of(context).colorScheme.inversePrimary,
55                                         ), // BoxDecoration
56                                         child: Text(countryCode, style: TextStyle(fontSize: 16, fontWeight: FontWeight.bold)),
57                                     ), // Container
58                                 ),
59                             ],
60                         ), // Row
61                     ),
62                     Padding(
63                         padding: const EdgeInsets.all(10),
64                         child: SizedBox(
65                             width: MediaQuery.of(context).size.width * 0.6,
66                             child: TextField(
67                                 controller: phoneNumberController,
68                                 keyboardType: TextInputType.number,
69                                 decoration: const InputDecoration(hintText: "Phone Number"),
70                             ), // TextField
71                         ), // SizedBox
72                     ),
73                 ],
74             ), // Column
75         ); // Scaffold
76     }
77 }
```

Output:



• Language picker

language_picker 0.4.3 copy language_picker: ^0.4.3 copied to clipboard

Published 10 months ago • @ wafrat.com Dart 3 compatible

SDK | FLUTTER | PLATFORM | ANDROID | IOS | LINUX | MACOS | WEB | WINDOWS

24

Readme Changelog Example Installing Versions Scores

language_picker

24 | 110 | 92%
LIKES PUB POINTS POPULARITY

Publisher
@ wafrat.com

Using

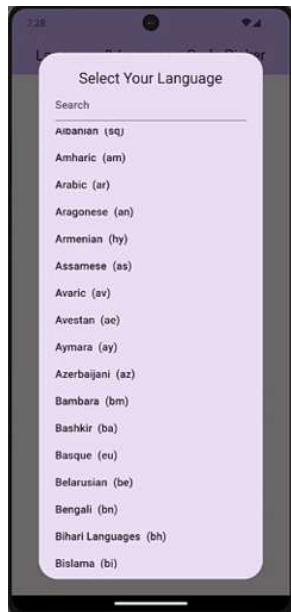
```
11 class _HomeScreenState extends State<HomeScreen> {
12   @override
13   Widget build(BuildContext context) {
14     return Scaffold(
15       appBar: AppBar(
16         centerTitle: true,
17         backgroundColor: Theme.of(context).colorScheme.inversePrimary,
18         title: const Text("Language & Language Code Picker"),
19       ), // AppBar
20
21       body: Center(
22         child: Column(
23           mainAxisAlignment: MainAxisAlignment.center,
24           children: [
25             Text("English", style: const TextStyle(fontWeight: FontWeight.bold, fontSize: 22)),
26
27             const SizedBox(height: 50),
28
29             ElevatedButton(
30               onPressed: () {
31                 _selectLanguage();
32               },
33               child: const Text("Select Your Language")) // ElevatedButton
34             ],
35           ), // Column
36         ), // Center
37     ); // Scaffold
38   }
39
40   _selectLanguage(){
41     return showDialog(
42       context: context,
43       builder: (context){
44         return LanguagePickerDialog();
45       });
46   }
47 }
```

Output



Updating

```
1  _selectLanguage(){
2    return showDialog(
3      context: context,
4      builder: (context){
5        return LanguagePickerDialog(
6          title: Text("Select Your Language"),
7          isSearchable: true,
8          searchInputDecoration: InputDecoration(hintText: "Search"),| |
9
10         itemBuilder: (Language language){
11           return Row(
12             children: [
13               Text(language.name),
14               SizedBox(width: 8),
15               Text("${language.isoCode}"))
16             ],
17           ); // Row
18         },
19       );
20     }
21   }
```



• Scratcher

scratcher 2.4.0

Published 58 days ago • @ rykowski.dev Dart 3 compatible

SDK FLUTTER PLATFORM ANDROID IOS LINUX MACOS WEB WINDOWS

391 391

[Readme](#) [Changelog](#) [Example](#) [Installing](#) [Versions](#) [Scores](#)

391 LIKES 140 PUBL POINTS 96% POPULARITY

scratcher

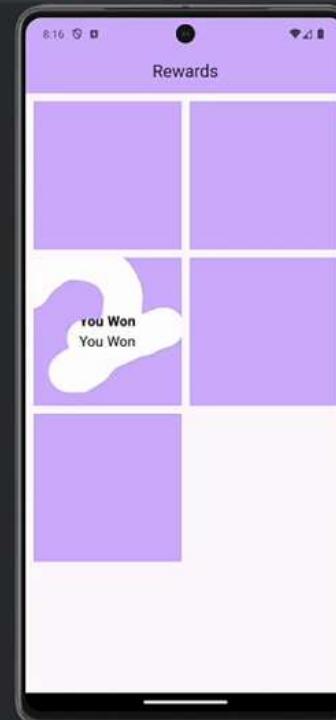
Publisher

@ rykowski.dev

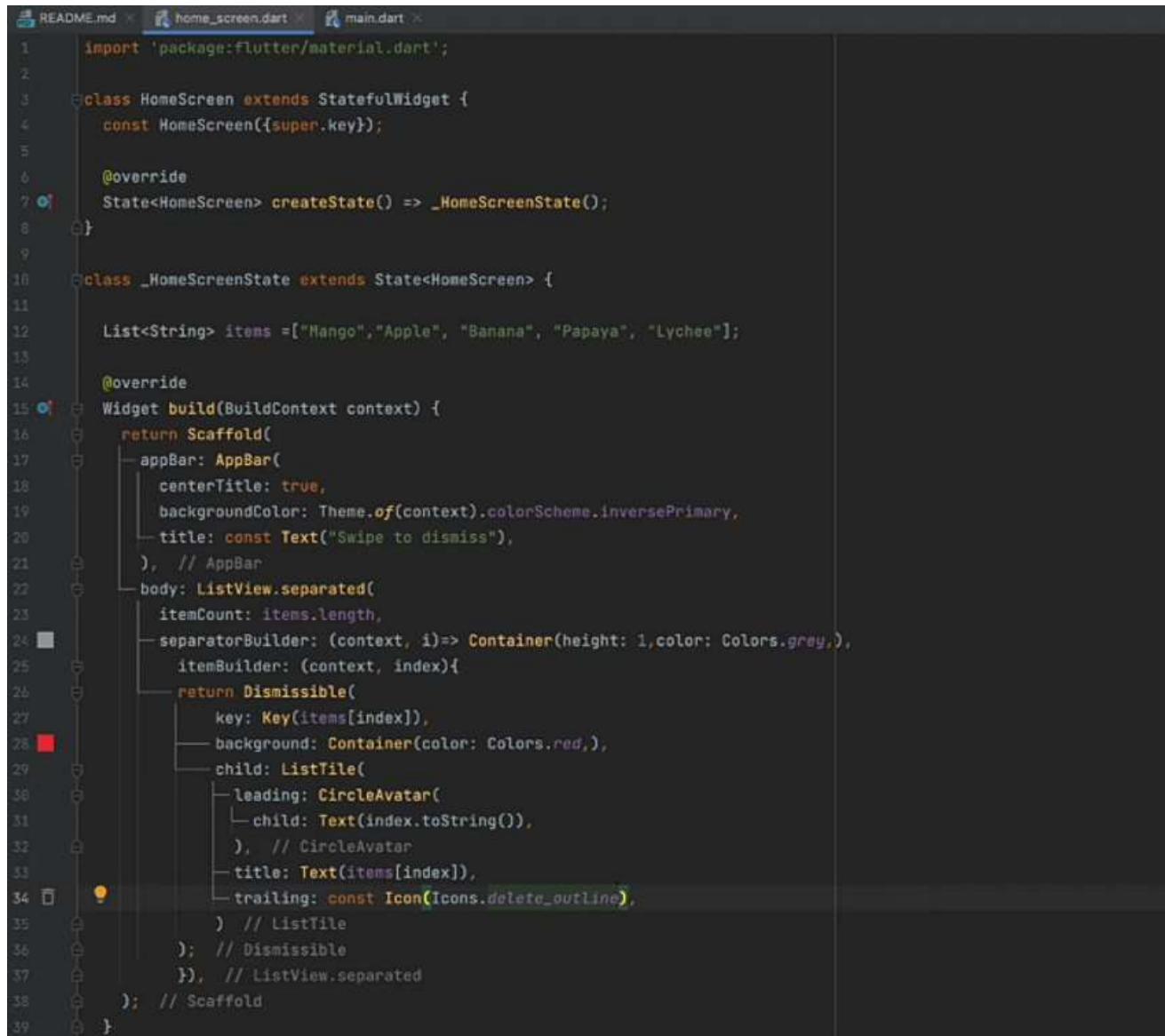
Scratch card widget which temporarily hides content from the user.

Metadata

```
1: //Rewards.dart
2: import 'package:flutter/material.dart';
3: import 'package:scratchy_flutter/scratches.dart';
4: 
5: class Rewards extends StatelessWidget {
6:   @override
7:   Widget build(BuildContext context) {
8:     return Scaffold(
9:       appBar: AppBar(
10:         centerTitle: true,
11:         backgroundColor: Theme.of(context).colorScheme.inversePrimary,
12:         title: const Text("Rewards"),
13:       ), // AppBar
14: 
15:       body: GridView.builder(
16:         itemCount: 5,
17:         padding: EdgeInsets.all(10),
18:         gridDelegate: const SliverGridDelegateWithFixedCrossAxisCount(
19:           crossAxisCount: 2,
20:           childAspectRatio: 2/3,
21:           mainAxisSpacing: 10,
22:           crossAxisSpacing: 10
23:         ), // SliverGridDelegateWithFixedCrossAxisCount
24:         itemBuilder: (context, index){
25:           return Container(
26:             decoration: BoxDecoration(
27:               color: Theme.of(context).colorScheme.inversePrimary,
28:               borderRadius: BorderRadius.circular(10)
29:             ), // BoxDecoration
30:             child: Scratcher(
31:               brushSize: 50,
32:               threshold: 50,
33:               color: Theme.of(context).colorScheme.inversePrimary,
34:               child: Container(
35:                 alignment: Alignment.center,
36:                 color: Colors.white,
37:                 child: Column(
38:                   mainAxisAlignment: MainAxisAlignment.center,
39:                   children: [
40:                     Text("You Won", style: TextStyle(color: Colors.black, fontSize: 18, fontWeight: FontWeight.bold)),
41:                     Text("You Won", style: TextStyle(color: Colors.black, fontSize: 18)),
42:                   ],
43:                 ), // Column
44:               ), // Scratcher
45:             ), // Container
46:           ); // Column
47:         }, // itemBuilder
48:       ), // GridView
49:     );
50:   }
51: }
```



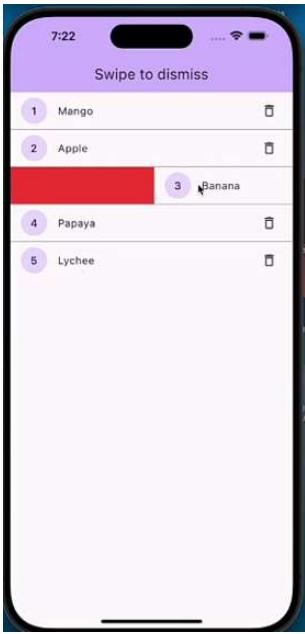
- Dismissable



The screenshot shows a code editor with three tabs at the top: README.md, home_screen.dart, and main.dart. The home_screen.dart tab is active and displays the following Dart code:

```
1 import 'package:flutter/material.dart';
2
3 class HomeScreen extends StatefulWidget {
4   const HomeScreen({super.key});
5
6   @override
7   State<HomeScreen> createState() => _HomeScreenState();
8 }
9
10 class _HomeScreenState extends State<HomeScreen> {
11
12   List<String> items = ["Mango", "Apple", "Banana", "Papaya", "Lychee"];
13
14   @override
15   Widget build(BuildContext context) {
16     return Scaffold(
17       appBar: AppBar(
18         centerTitle: true,
19         backgroundColor: Theme.of(context).colorScheme.inversePrimary,
20         title: const Text("Swipe to dismiss"),
21       ), // AppBar
22       body: ListView.separated(
23         itemCount: items.length,
24         separatorBuilder: (context, i) => Container(height: 1, color: Colors.grey),
25         itemBuilder: (context, index) {
26           return Dismissible(
27             key: Key(items[index]),
28             background: Container(color: Colors.red),
29             child: ListTile(
30               leading: CircleAvatar(
31                 child: Text(index.toString()),
32               ), // CircleAvatar
33               title: Text(items[index]),
34               trailing: const Icon(Icons.delete_outline),
35             ), // ListTile
36           ); // Dismissible
37         }, // ListView.separated
38       ); // Scaffold
39     }
40 }
```

Output



onDismissed() event

The screenshot shows the same application running on an iPhone X simulator. The list now only contains four items: 1. Mango, 2. Apple, 3. Banana, and 4. Papaya. The 'Banana' item is no longer visible, having been dismissed. A red rectangular box highlights the word 'Deleted' at the bottom of the list, which is the result of the 'onDismissed' event being triggered.

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

  @override
  State<HomeScreen> createState() => _HomeScreenState();
}

class _HomeScreenState extends State<HomeScreen> {
  List<String> items = ["Mango", "Apple", "Banana", "Papaya", "Lychee"];

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        centerTitle: true,
        backgroundColor: Theme.of(context).colorScheme.inversePrimary,
        title: const Text("Swipe to dismiss"),
      ),
      body: ListView.separated(
        itemCount: items.length,
        separatorBuilder: (context, i) => Container(height: 1, color: Colors.grey),
        itemBuilder: (context, index) {
          return Dismissible(
            key: Key(items[index]),
            background: Container(color: Colors.red),
            onDismissed: (direction) {
              items.removeAt(index);
              ScaffoldMessenger.of(context).showSnackBar(const SnackBar(
                backgroundColor: Colors.redAccent,
                content: Text("Deleted", style: TextStyle(fontSize: 20, color: Colors.white)),));
            },
            child: ListTile(
              leading: CircleAvatar(
                child: Text("${index+1}"),
              ),
              title: Text(items[index]),
            ),
          );
        },
      ),
    );
  }
}
```

- CachedImage (to handle error when image not found)

[cached_network_image 3.3.0](#)

Published 53 days ago • @ baseflow.com Dart 3 compatible

SDK FLUTTER PLATFORM ANDROID IOS MACOS

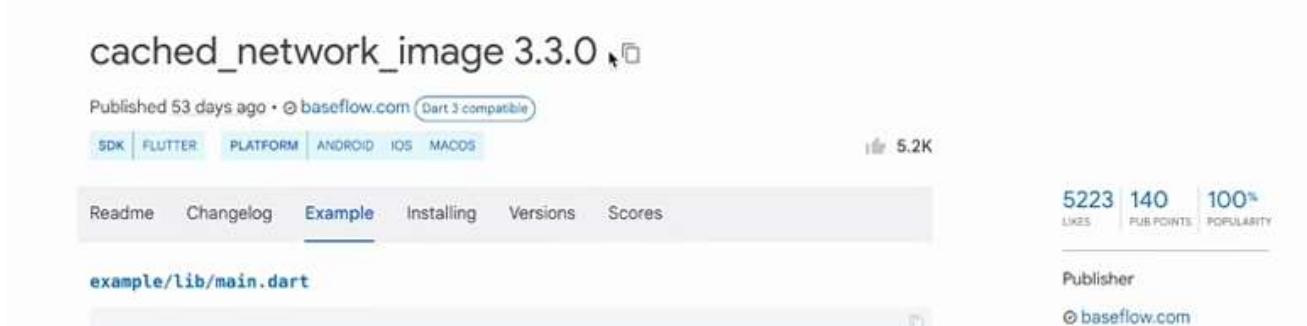
5.2K

Readme Changelog Example Installing Versions Scores

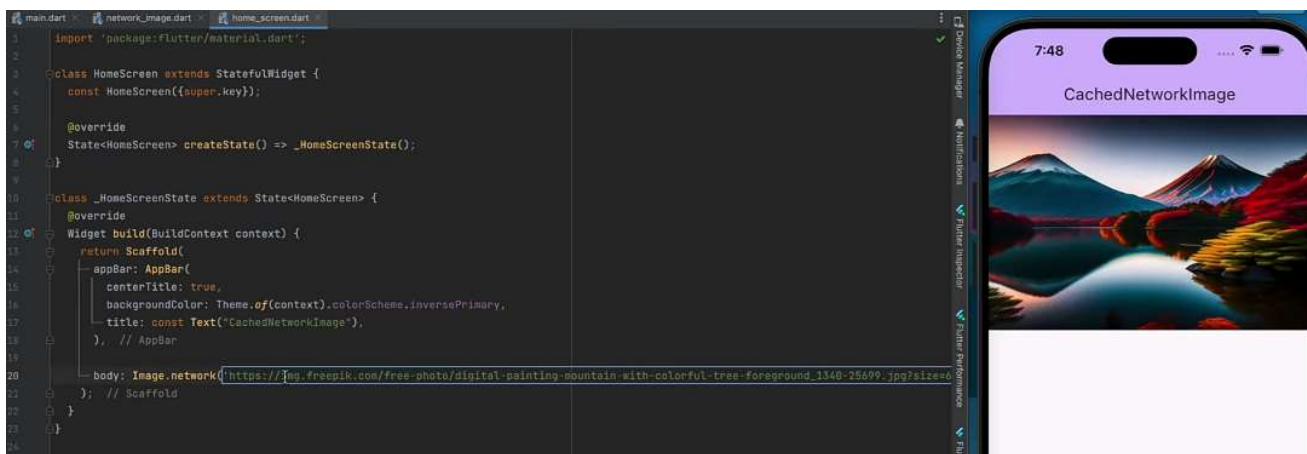
example/lib/main.dart

5223 140 100% LIKES PUB POINTS POPULARITY

Publisher @ baseflow.com



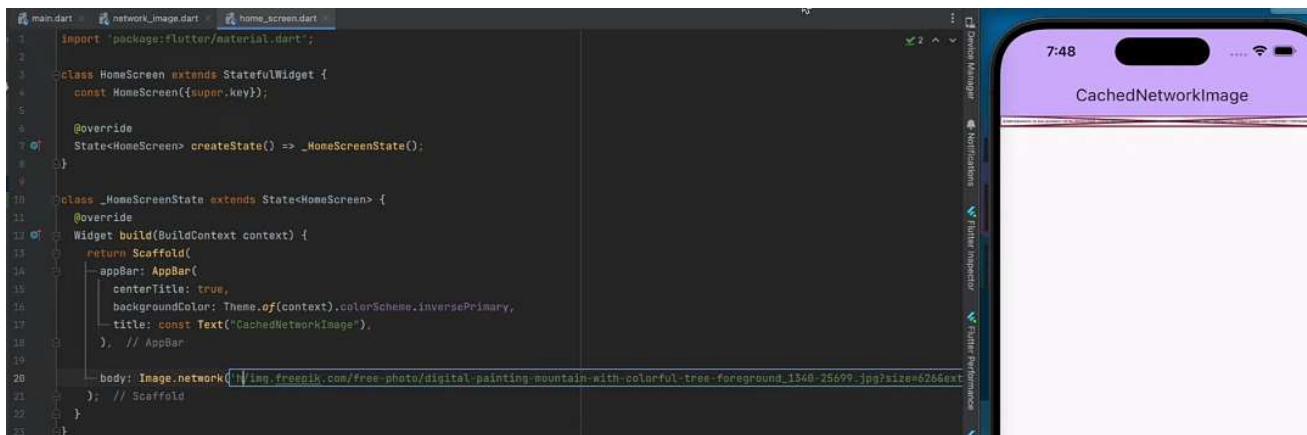
Normal image using Image.network



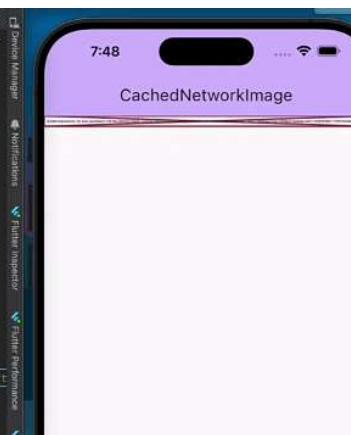
```
main.dart network_image.dart home_screen.dart
1 import 'package:flutter/material.dart';
2
3 class HomeScreen extends StatefulWidget {
4   const HomeScreen({super.key});
5
6   @override
7   State<HomeScreen> createState() => _HomeScreenState();
8 }
9
10 class _HomeScreenState extends State<HomeScreen> {
11   @override
12   Widget build(BuildContext context) {
13     return Scaffold(
14       appBar: AppBar(
15         centerTitle: true,
16         backgroundColor: Theme.of(context).colorScheme.inversePrimary,
17         title: const Text("CachedNetworkImage"),
18       ), // AppBar
19
20       body: Image.network("https://img.freepik.com/free-photo/digital-painting-mountain-with-colorful-tree-foreground_1340-25699.jpg?size=640");
21     ); // Scaffold
22   }
23 }
```



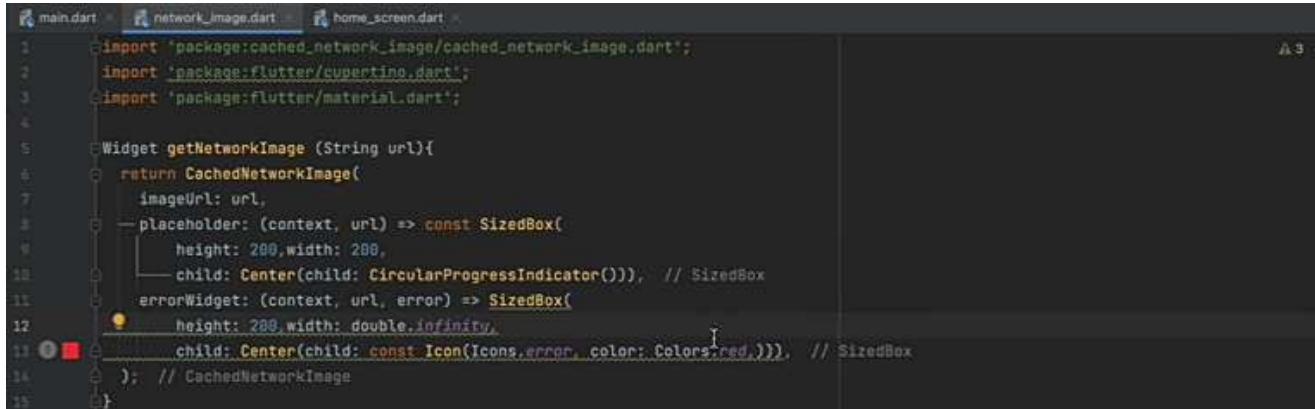
When image not found



```
main.dart network_image.dart home_screen.dart
1 import 'package:flutter/material.dart';
2
3 class HomeScreen extends StatefulWidget {
4   const HomeScreen({super.key});
5
6   @override
7   State<HomeScreen> createState() => _HomeScreenState();
8 }
9
10 class _HomeScreenState extends State<HomeScreen> {
11   @override
12   Widget build(BuildContext context) {
13     return Scaffold(
14       appBar: AppBar(
15         centerTitle: true,
16         backgroundColor: Theme.of(context).colorScheme.inversePrimary,
17         title: const Text("CachedNetworkImage"),
18       ), // AppBar
19
20       body: Image.network("https://img.freepik.com/free-photo/digital-painting-mountain-with-colorful-tree-foreground_1340-25699.jpg?size=6266ext");
21     ); // Scaffold
22   }
23 }
```

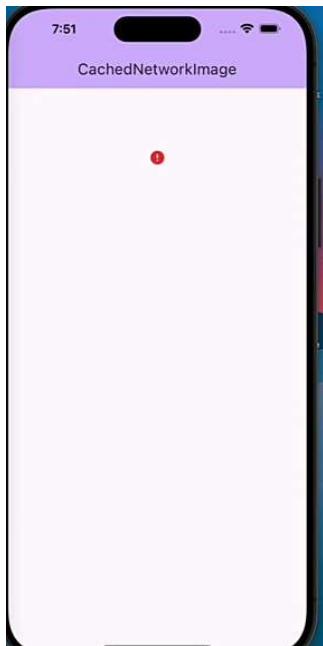


Function



```
1 import 'package:cached_network_image/cached_network_image.dart';
2 import 'package:flutter/cupertino.dart';
3 import 'package:flutter/material.dart';
4
5 Widget getNetworkImage(String url){
6   return CachedNetworkImage(
7     imageUrl: url,
8     placeholder: (context, url) => const SizedBox(
9       height: 200,width: 200,
10      child: Center(child: CircularProgressIndicator()), // SizedBox
11    ),
12    errorWidget: (context, url, error) => SizedBox(
13      height: 200,width: double.infinity,
14      child: Center(child: const Icon(Icons.error, color: Colors.red,)), // SizedBox
15    ); // CachedNetworkImage
}
```

Using and output when image is not found



• Flutter vibrate



flutter_vibrate 1.3.0 [Copy link](#) copied to clipboard

Published 2 years ago Dart 3 compatible

SDK FLUTTER PLATFORM ANDROID iOS 169

Readme Changelog Example Installing Versions Scores

Donate Buy Me A Coffee Donate PayPal GitHub pages no status Stars 85 pub.v1.3.0

169 LIKES 130 PUB POINTS 98% POPULARITY

Publisher unverified uploader

Getting Started

Make sure you add the following permissions to your Android Manifest

```
<uses-permission android:name="android.permission.VIBRATE"/>
```

[example/lib/main.dart](#)

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

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

class MyApp extends StatefulWidget {
  const MyApp({Key? key}) : super(key: key);

  @override
  _MyAppState createState() => _MyAppState();
}

class _MyAppState extends State<MyApp> {
  bool _canVibrate = true;
  final Iterable<Duration> pauses = [
    const Duration(milliseconds: 500),
    const Duration(milliseconds: 1000),
    const Duration(milliseconds: 500),
  ];

  @override
  void initState() {
    super.initState();
    _init();
  }

  Future<void> _init() async {
    bool canVibrate = await Vibrate.canVibrate;
    setState(() {
      _canVibrate = canVibrate;
      canVibrate
        ? debugPrint('This device can vibrate')
        : debugPrint('This device cannot vibrate');
    });
  }

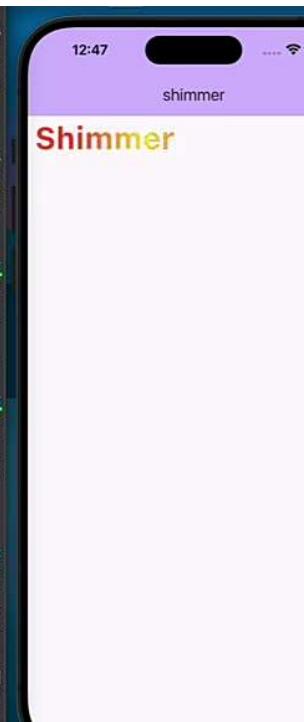
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Flutter Vibration Example',
      theme: ThemeData(
        primarySwatch: Colors.blue,
      ),
      home: Scaffold(
        appBar: AppBar(
          title: Text('Flutter Vibration Example'),
        ),
        body: Center(
          child: Text(_canVibrate ? 'Device can vibrate' : 'Device cannot vibrate'),
        ),
      ),
    );
  }
}
```

- [Shimmer effect \(when youtube don't have net it shows\)](#)

[Shimmer 3.0.0](#) Published 12 months ago •  hunghd.dev [SDK](#) [FLUTTER](#) [PLATFORM](#) [ANDROID](#) [IOS](#) [LINUX](#) [MACOS](#) [WEB](#) [WINDOWS](#) 4.6K[Shimmer](#)

Publisher

 hunghd.dev

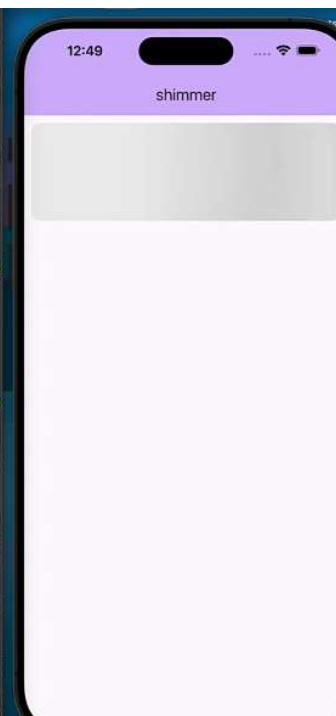


Code editor showing the `home_screen.dart` file:

```
main.dart home_screen.dart pubspec.yaml
override
  State<HomeScreen> createState() => _HomeScreenState();
}

class _HomeScreenState extends State<HomeScreen> {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        centerTitle: true,
        backgroundColor: Theme.of(context).colorScheme.inversePrimary,
        title: const Text("shimmer"),
      ), // AppBar

      body: SizedBox(
        width: 200.0,
        height: 100.0,
        child: Shimmer.fromColors(
          baseColor: Colors.red,
          highlightColor: Colors.yellow,
          child: Text(
            'Shimmer',
            textAlign: TextAlign.center,
            style: TextStyle(
              fontSize: 40.0,
              fontWeight: FontWeight.bold,
            ),
          ), // Text
        ), // Shimmer.fromColors
      ), // SizedBox
    ); // Scaffold
  }
}
```



Code editor showing the `home_screen.dart` file with changes in the `body` section:

```
main.dart home_screen.dart colors.dart pubspec.yaml
override
  State<HomeScreen> createState() => _HomeScreenState();
}

class _HomeScreenState extends State<HomeScreen> {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        centerTitle: true,
        backgroundColor: Theme.of(context).colorScheme.inversePrimary,
        title: const Text("shimmer"),
      ), // AppBar

      body: SizedBox(
        height: 150.0,
        child: Shimmer.fromColors(
          baseColor: Color(0xFFFFFFFF),
          highlightColor: Color(0xFFD9EAD3),
          child: Container(
            margin: EdgeInsets.all(10),
            decoration: BoxDecoration(
              color: Colors.white,
              borderRadius: BorderRadius.circular(12),
            ), // BoxDecoration
          ), // Container
        ), // Shimmer.fromColors
      ), // SizedBox
    ); // Scaffold
  }
}
```

```

7   @override
8     State<HomeScreen> createState() => _HomeScreenState();
9   }
10
11   class _HomeScreenState extends State<HomeScreen> {
12     @override
13     Widget build(BuildContext context) {
14       return Scaffold(
15         appBar: AppBar(
16           centerTitle: true,
17           backgroundColor: Theme.of(context).colorScheme.inversePrimary,
18           title: const Text("shimmer"),
19         ), // AppBar
20
21         body: ListView.builder(
22           itemCount: 5,
23           itemBuilder: (context, index){
24             return SizedBox(
25               height: 150.0,
26               child: Shimmer.fromColors(
27                 baseColor: Color(0xFFFFFFFF),
28                 highlightColor: Color(0xFFDADBD8),
29                 child: Container(
30                   margin: EdgeInsets.all(10),
31                   decoration: BoxDecoration(
32                     color: Colors.white,
33                     borderRadius: BorderRadius.circular(12)
34                   ), // BoxDecoration
35                 ), // Container
36               ), // Shimmer.fromColors
37             ); // SizedBox
38           }, // ListView.builder
39         );
40       }
41     }

```

- [Newton particles](#)

[newton_particles 0.1.8](#)

Published 5 days ago • [@7omtech.fr](#) Dart 3 compatible

[SDK](#) [FLUTTER](#) [PLATFORM](#) [ANDROID](#) [IOS](#) [LINUX](#) [MACOS](#) [WEB](#) [WINDOWS](#)

1.0.0 65

[Readme](#) [Changelog](#) [Example](#) [Installing](#) [Versions](#) [Scores](#)

65 LIKES 160 PUB POINTS 83% POPULARITY

Publisher

[@7omtech.fr](#)

Metadata

Newton

Particle Emitter for Flutter

Newton is a highly configurable particle emitter package for Flutter that allows you to create captivating animations such as rain, smoke, explosions, and more. With Newton, you can easily add visually stunning effects to your Flutter applications.

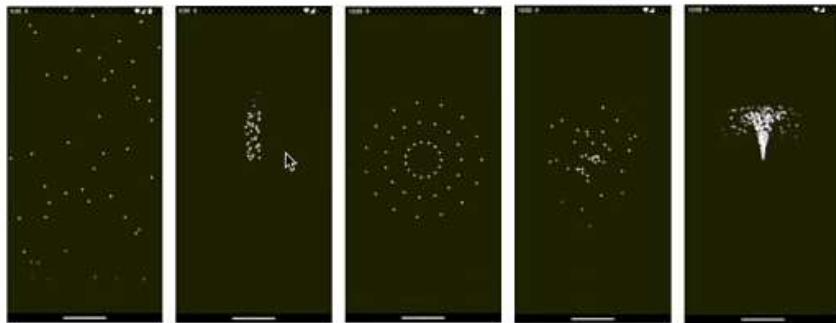
Rain

Smoke

Pulse

Explode

Fountain



configurable particle emitters. Add rain, smoke, and more effects to your Flutter app!

[Homepage](#)

[Repository \(GitHub\)](#)

[View/report issues](#)

[Documentation](#)

[API reference](#)

[License](#)

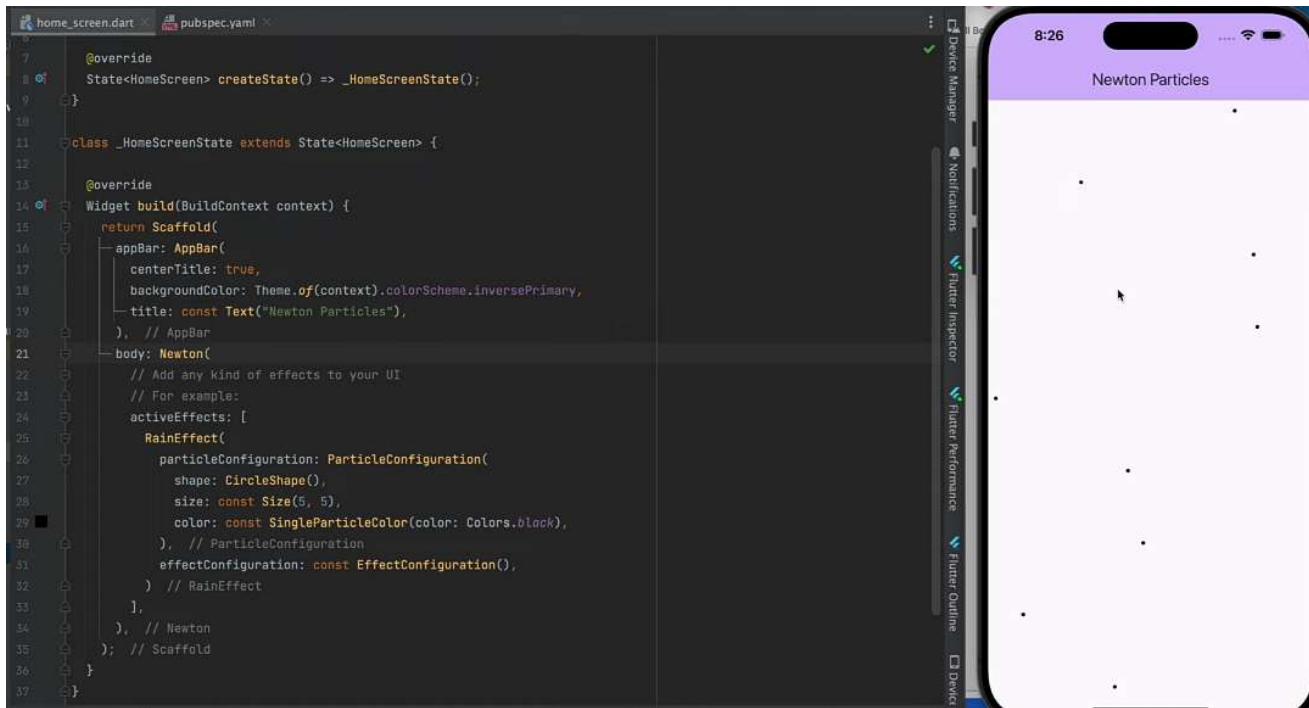
[MIT \(LICENSE\)](#)

[Dependencies](#)

[flutter, vector_math](#)

[More](#)

Packages that depend on newton_particles



- Flutter ripple effect

[flutter_ripple 0.0.3](#)

Published 2 years ago Dart 3 compatible

SDK FLUTTER PLATFORM ANDROID IOS LINUX MACOS WEB WINDOWS

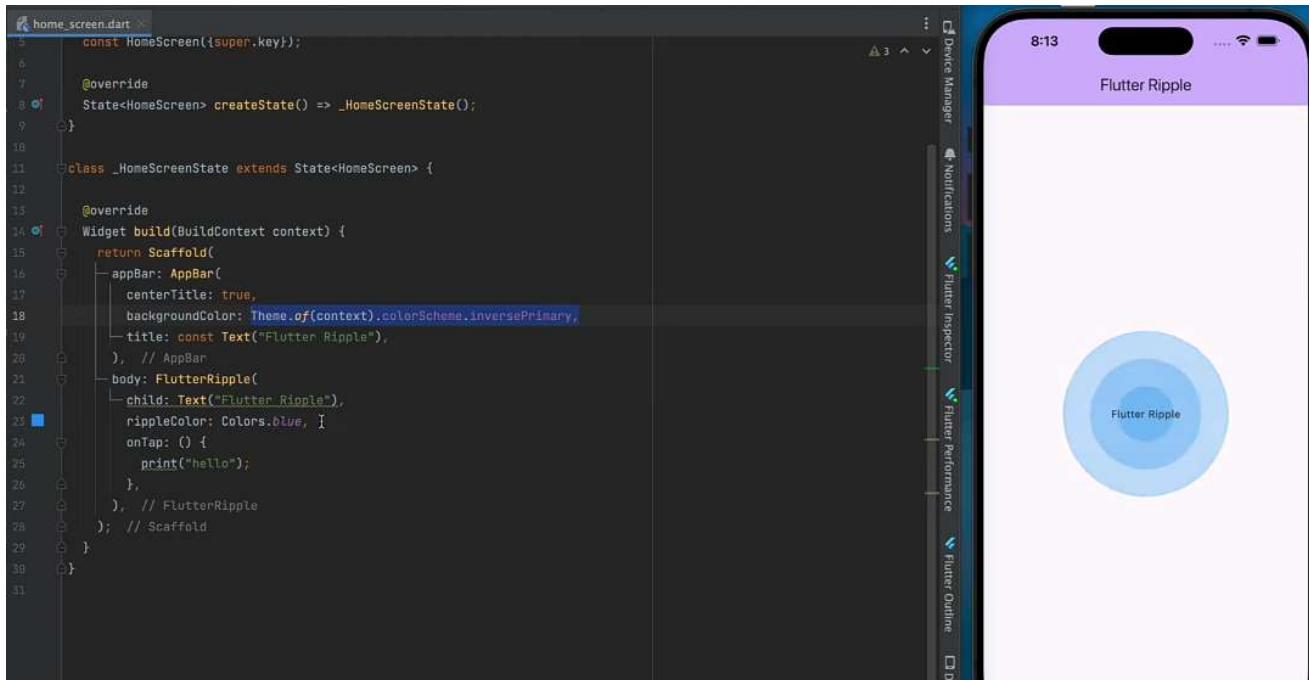
16

[Readme](#) [Changelog](#) [Example](#) [Installing](#) [Versions](#) [Scores](#)

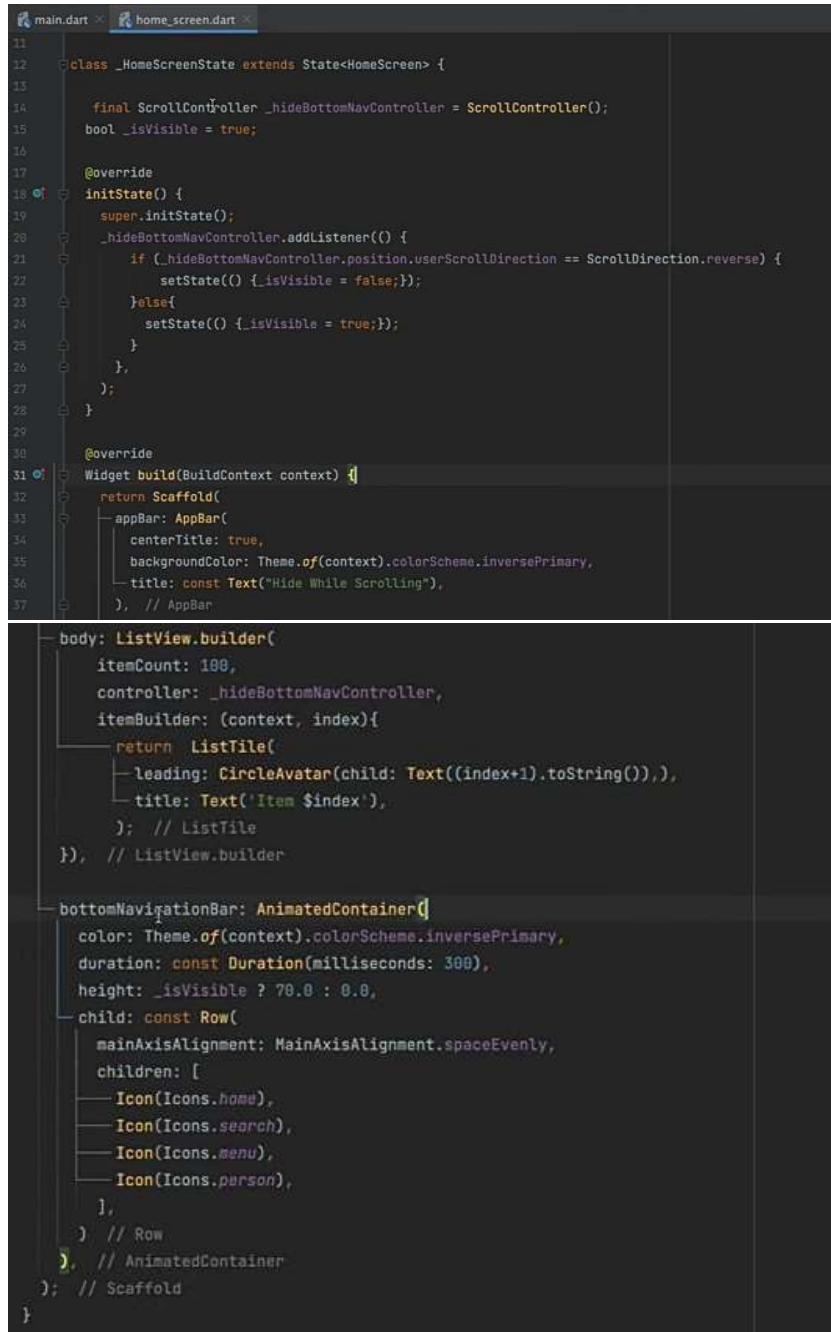
16 LIKES 150 PUB POINTS 87% POPULARITY

[flutter_ripple](#)

Publisher
unverified uploader

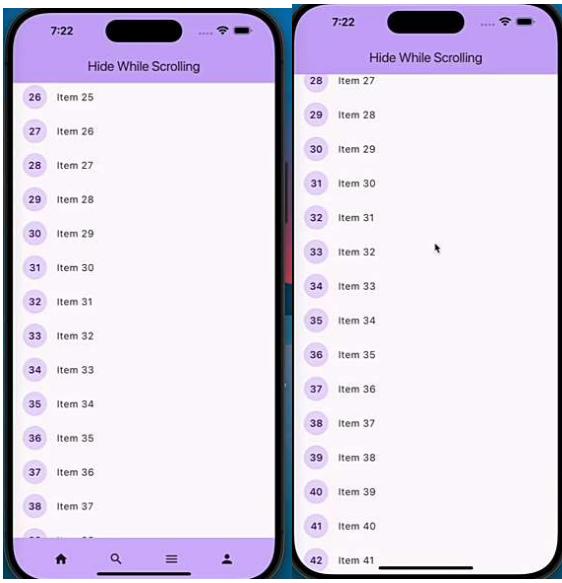


- Hide bottom navigation while scrolling



```
11
12 class _HomeScreenState extends State<HomeScreen> {
13
14     final ScrollController _hideBottomNavController = ScrollController();
15     bool _isVisible = true;
16
17     @override
18     void initState() {
19         super.initState();
20         _hideBottomNavController.addListener(() {
21             if (_hideBottomNavController.position.userScrollDirection == ScrollDirection.reverse) {
22                 setState(() {_isVisible = false});
23             } else{
24                 setState(() {_isVisible = true});
25             }
26         });
27     }
28
29     @override
30     Widget build(BuildContext context) {
31         return Scaffold(
32             appBar: AppBar(
33                 centerTitle: true,
34                 backgroundColor: Theme.of(context).colorScheme.inversePrimary,
35                 title: const Text("Hide While Scrolling"),
36             ), // AppBar
37
38             body: ListView.builder(
39                 itemCount: 100,
40                 controller: _hideBottomNavController,
41                 itemBuilder: (context, index){
42                     return ListTile(
43                         leading: CircleAvatar(child: Text((index+1).toString()),),
44                         title: Text('Item $index'),
45                     ); // ListTile
46                 }, // ListView.builder
47
48                 bottomNavigationBar: AnimatedContainer(
49                     color: Theme.of(context).colorScheme.inversePrimary,
50                     duration: const Duration(milliseconds: 300),
51                     height: _isVisible ? 70.0 : 0.0,
52                     child: const Row(
53                         mainAxisAlignment: MainAxisAlignment.spaceEvenly,
54                         children: [
55                             Icon(Icons.home),
56                             Icon(Icons.search),
57                             Icon(Icons.menu),
58                             Icon(Icons.person),
59                         ],
60                     ); // Row
61                 ), // AnimatedContainer
62             ); // Scaffold
63     }
64 }
```

Output



• Geolocator

The screenshot shows the package page for 'geolocator 11.0.0' on pub.dev. At the top right, there's a 'Flutter Favorite' button. Below it, the package name 'geolocator 11.0.0' is displayed with a copy icon. A timestamp 'Published 38 days ago' and a link to 'baseflow.com' are shown. The package is marked as 'Dart 3 compatible'. Below this, there are tabs for 'SDK', 'FLUTTER' (which is selected), 'PLATFORM', and 'ANDROID', 'IOS', 'MACOS', 'WEB', 'WINDOWS'. The 'FLUTTER' tab has a '5.0K' rating icon. To the right, there are statistics: '5033 LIKES', '140 PUB POINTS', and '100% POPULARITY'. Below these stats, it says 'Publisher baseflow.com'. The main content area features a title 'Flutter Geolocator Plugin'.

AndroidX

The geolocator plugin requires the AndroidX version of the Android Support Libraries. This means you need to make sure your Android project supports AndroidX. Detailed instructions can be found [here](#).

The TL;DR version is:

1. Add the following to your "gradle.properties" file:

```
android.useAndroidX=true  
android.enableJetifier=true
```

2. Make sure you set the `compileSdkVersion` in your "android/app/build.gradle" file to 33:

```
android {  
    compileSdkVersion 33  
    ...  
}
```

3. Make sure you replace all the `android.` dependencies to their AndroidX counterparts (a full list can be found here: [Migrating to AndroidX](#)).

Permissions

On Android you'll need to add either the `ACCESS_COARSE_LOCATION` or the `ACCESS_FINE_LOCATION` permission to your Android Manifest. To do so open the `AndroidManifest.xml` file (located under `android/app/src/main`) and add one of the following two lines as direct children of the `<manifest>` tag (when you configure both permissions the `ACCESS_FINE_LOCATION` will be used by the geolocator plugin):

```
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />  
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
```

Starting from Android 10 you need to add the `ACCESS_BACKGROUND_LOCATION` permission (next to the `ACCESS_COARSE_LOCATION` or the `ACCESS_FINE_LOCATION` permission) if you want to continue receiving updates even when your App is running in the background (note that the geolocator plugin doesn't support receiving and processing location updates while running in the background):

```
<uses-permission android:name="android.permission.ACCESS_BACKGROUND_LOCATION" />
```

NOTE: Specifying the `ACCESS_COARSE_LOCATION` permission results in location updates with an accuracy approximately equivalent to a city block. It might take a long time (minutes) before you will get your first locations fix as `ACCESS_COARSE_LOCATION` will only use the network services to calculate the position of the device. More information can be found [here](#).

widget_of_the_week ~/Desktop/widget

Flutter commands

```

1  plugins {
2      id "com.android.application"
3      id "Kotlin-Android"
4      id "dev.flutter.flutter-gradle-plugin"
5  }
6
7  def localProperties = new Properties()
8  def localPropertiesFile = rootProject.file('local.properties')
9  if (localPropertiesFile.exists()) {
10      localPropertiesFile.withReader('UTF-8') { reader ->
11          localProperties.load(reader)
12      }
13  }
14
15  def flutterVersionCode = localProperties.getProperty('flutter.versionCode')
16  if (flutterVersionCode == null) {
17      flutterVersionCode = '1'
18  }
19
20  def flutterVersionName = localProperties.getProperty('flutter.versionName')
21  if (flutterVersionName == null) {
22      flutterVersionName = '1.0'
23  }
24
25  android {
26      namespace "com.example.widget_of_the_week"
27      compileSdkVersion 33
28      ndkVersion flutter.ndkVersion
29
30      compileOptions {
31          sourceCompatibility JavaVersion.VERSION_1_8
32          targetCompatibility JavaVersion.VERSION_1_8
33      }
34
35      kotlinOptions {
36          jvmTarget = '1.8'
37      }
38  }

```

Open for Editing in Android Studio

```

import 'package:geolocator/geolocator.dart';

/// Determine the current position of the device.
///
/// When the location services are not enabled or permissions
/// are denied the 'Future' will return an error.
Future<Position> _determinePosition() async {
    bool serviceEnabled;
    LocationPermission permission;

    // Test if location services are enabled.
    serviceEnabled = await Geolocator.isLocationServiceEnabled();
    if (!serviceEnabled) {
        // Location services are not enabled don't continue
        // accessing the position and request users of the
        // App to enable the location services.
        return Future.error('Location services are disabled.');
    }

    permission = await Geolocator.checkPermission();
    if (permission == LocationPermission.denied) {
        permission = await Geolocator.requestPermission();
        if (permission == LocationPermission.denied) {
            // Permissions are denied, next time you could try
            // requesting permissions again (this is also where
            // Android's shouldShowRequestPermissionRationale
            // returned true. According to Android guidelines
            // your App should show an explanatory UI now.
            return Future.error('Location permissions are denied');
        }
    }

    if (permission == LocationPermission.deniedForever) {
        // Permissions are denied forever, handle appropriately.
        return Future.error(
            'Location permissions are permanently denied, we cannot request permissions.');
    }

    // When we reach here, permissions are granted and we can
    // continue accessing the position of the device.
    return await Geolocator.getCurrentPosition();
}

```

```
import 'package:geolocator/geolocator.dart';
await Geolocator.openAppSettings();
await Geolocator.openLocationSettings();
```

Utility methods

To calculate the distance (in meters) between two geocoordinates you can use the `distanceBetween` method. The `distanceBetween` method takes four parameters:

Parameter	Type	Description
startLatitude	double	Latitude of the start position
startLongitude	double	Longitude of the start position
endLatitude	double	Latitude of the destination position
endLongitude	double	Longitude of the destination position

```
import 'package:geolocator/geolocator.dart';
double distanceInMeters = Geolocator.distanceBetween(52.2165157, 6.9437819, 52.3546274, 4.81);
```

The screenshot shows a code editor with a dark theme. The main.dart file is open, displaying code related to location permissions. The code imports flutter/cupertino.dart, flutter/material.dart, and geolocator/geolocator.dart. It defines a Future<Position> function named determinePosition that takes a BuildContext context. Inside, it uses await to get the GeolocatorPlatform instance. It checks if the service is enabled and handles denied permission by showing an alert. Finally, it returns the current position.

```
1 import 'package:flutter/cupertino.dart';
2 import 'package:flutter/material.dart';
3 import 'package:geolocator/geolocator.dart';
4
5 Future<Position> determinePosition(BuildContext context) async {
6   final GeolocatorPlatform _geolocatorPlatform = GeolocatorPlatform.instance;
7
8   bool serviceEnabled;
9   LocationPermission permission;
10
11   serviceEnabled = await _geolocatorPlatform.isLocationServiceEnabled();
12   if (!serviceEnabled) {
13     return Future.error('Location services are disabled.');
14   }
15
16   permission = await _geolocatorPlatform.checkPermission();
17   if (permission == LocationPermission.denied) {
18     permission = await _geolocatorPlatform.requestPermission();
19
20     if (permission == LocationPermission.denied) {
21       alertPopup(context);
22     }
23   }
24
25   if (permission == LocationPermission.deniedForever) {
26     alertPopup(context);
27   }
28
29   return await _geolocatorPlatform.getCurrentPosition();
30 }
```

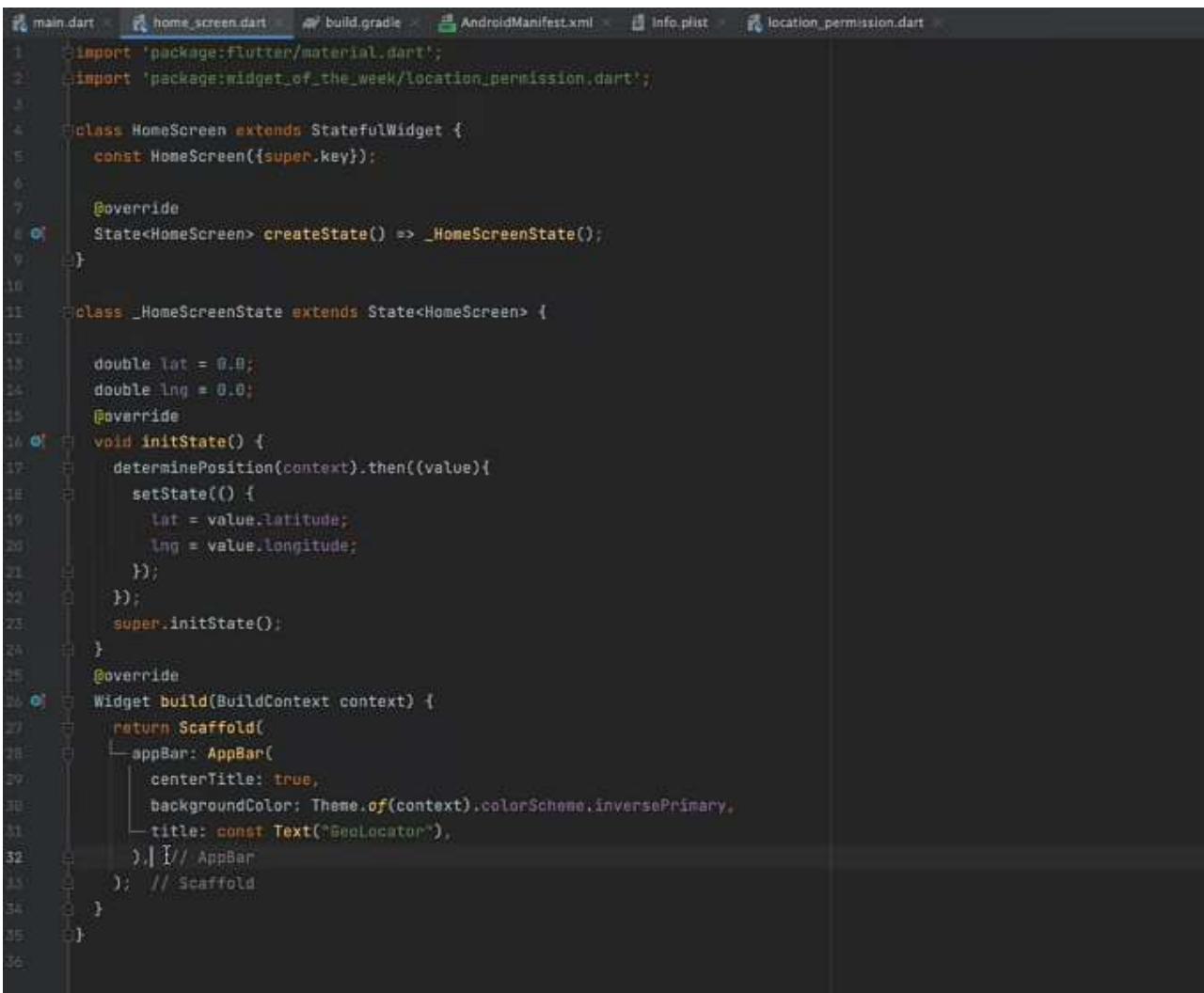
```

alertPopup(BuildContext context){
  showDialog(context: context,
    builder: (context){
      return AlertDialog(
        title: const Text('Location Permission'),
        content: Column(
          mainAxisSize: MainAxisSize.min,
          children: [
            const Text('Location permission is required to use app.'),
            const SizedBox(height: 20),

            ElevatedButton(onPressed: ()async{
              final GeolocatorPlatform _geolocatorPlatform = GeolocatorPlatform.instance;
              await _geolocatorPlatform.openAppSettings();
            }, child: const Text('Ok')) // ElevatedButton
          ],
        );
      );
    });
}

```

Home

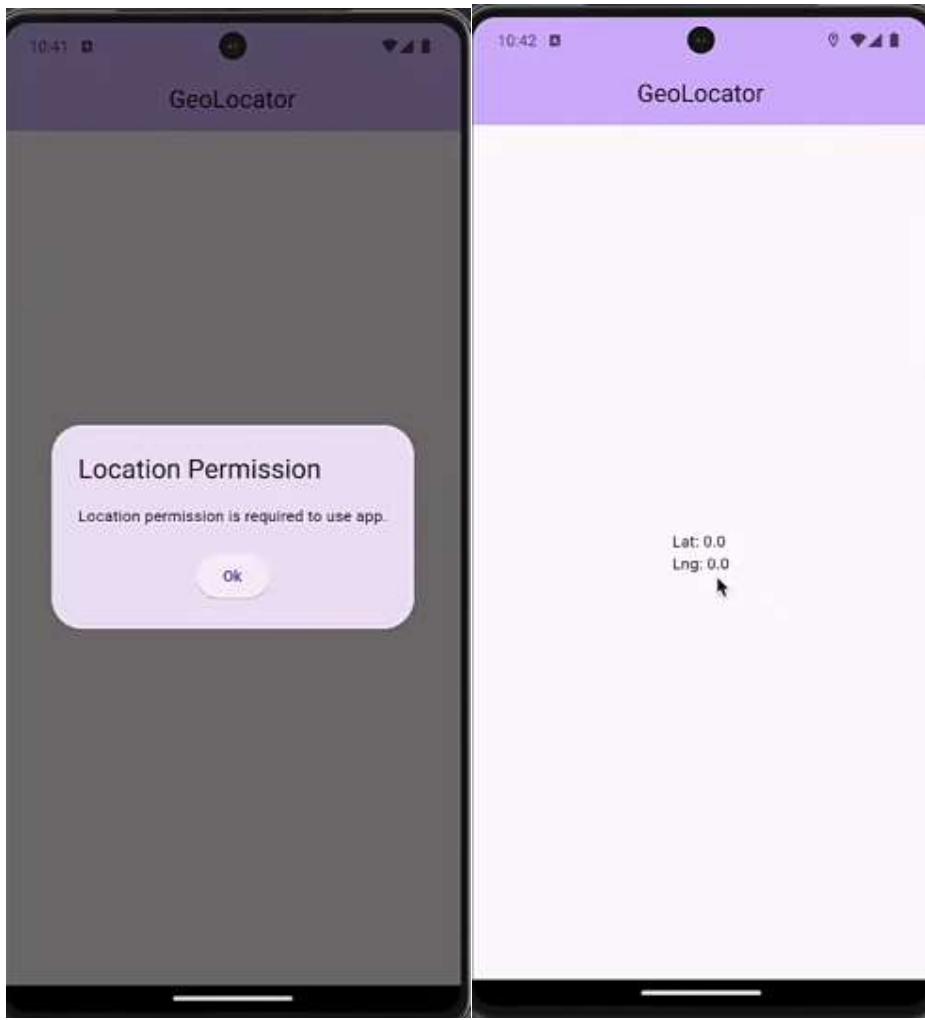


```

main.dart  home_screen.dart  build.gradle  AndroidManifest.xml  info.plist  location_permission.dart
1 import 'package:flutter/material.dart';
2 import 'package:widget_of_the_week/location_permission.dart';
3
4 class HomeScreen extends StatefulWidget {
5   const HomeScreen({super.key});
6
7   @override
8   State<HomeScreen> createState() => _HomeScreenState();
9 }
10
11 class _HomeScreenState extends State<HomeScreen> {
12
13   double lat = 0.0;
14   double lng = 0.0;
15   @override
16   void initState() {
17     determinePosition(context).then((value){
18       setState(() {
19         lat = value.latitude;
20         lng = value.longitude;
21       });
22     });
23     super.initState();
24   }
25   @override
26   Widget build(BuildContext context) {
27     return Scaffold(
28       appBar: AppBar(
29         centerTitle: true,
30         backgroundColor: Theme.of(context).colorScheme.inversePrimary,
31         title: const Text("GeoLocator"),
32       ), // AppBar
33     ); // Scaffold
34   }
35 }

```

Output



- **Flutter mock location**

fluttermocklocation 0.0.5

Published 4 months ago Dart 3 compatible

fluttermocklocation: ^0.0.5 copied to clipboard

SDK FLUTTER PLATFORM ANDROID ⚡ 5

[Readme](#) [Changelog](#) [Example](#) [Installing](#) [Versions](#) [Scores](#)

fluttermocklocation

This Flutter plugin is designed to assist developers in setting a mock location on Android devices during the testing phase of applications that require location data, such as geofencing apps, and others that rely on GPS positioning.

By enabling the simulation of location data, developers can ensure their applications behave as expected in various geographic locations without the need to physically move the device. This tool is essential for thorough testing of location-based features, ensuring a wide range of scenarios can be tested efficiently and effectively.

To use the application based on this plugin, please enable Developer Options on your Android device.

Within Developer Options, select "Select mock location app" and choose your app.

- Video player

appinio_video_player 1.3.0

Published 4 months ago •  Dart 3 compatible

SDK FLUTTER PLATFORM ANDROID IOS WEB

229

Readme Changelog Example Installing Versions Scores

Custom Video Player

This package wraps the official video_player package by flutter and extends it with a fully customisable control bar, a fullscreen mode and adjustable video settings. For every control in this video player you can decide if you want to show it and if so how it should look.

```
class _MyHomePageState extends State<MyHomePage> {
  late VideoPlayerController videoPlayerController;
  late CustomVideoPlayerController _customVideoPlayerController;

  String videoUrl =
    "http://commondatastorage.googleapis.com/gtv-videos-bucket/sample/BigBuckBunny.mp4";

  @override
  void initState() {
    super.initState();
    videoPlayerController = VideoPlayerController.network(videoUrl)
      ..initialize().then((value) => setState(() {}));
    _customVideoPlayerController = CustomVideoPlayerController(
      context: context,
      videoPlayerController: videoPlayerController,
    );
  }

  @override
  void dispose() {
    _customVideoPlayerController.dispose();
    super.dispose();
  }

  @override
  Widget build(BuildContext context) {
    return CupertinoPageScaffold(
      navigationBar: CupertinoNavigationBar(
        middle: Text(widget.title),
      ),
      child: SafeArea(
        child: CustomVideoPlayer(
          customVideoPlayerController: _customVideoPlayerController
        ),
      ),
    );
  }
}
```

- Qr flutter

qr_flutter 4.1.0

Published 14 months ago • @ theyakka.com Dart 3 compatible

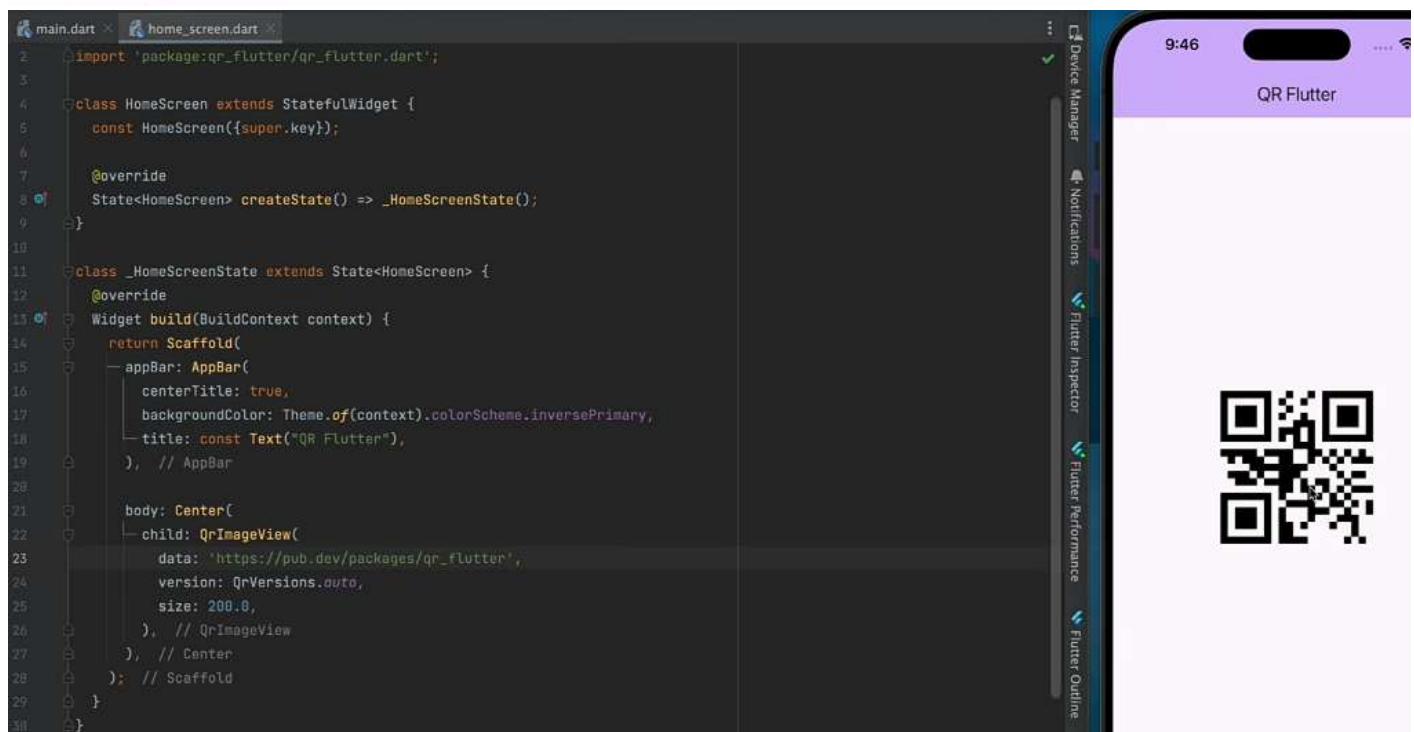
SDK FLUTTER PLATFORM ANDROID IOS LINUX MACOS WEB WINDOWS

1.9K

Readme Changelog Example Installing Versions Scores

QR.FLUTTER

QR.Flutter is a Flutter library for simple and fast QR code rendering via a Widget or custom painter.



The screenshot shows a Flutter development environment. On the left, there are two code files: `main.dart` and `home_screen.dart`. The `home_screen.dart` file contains the following Dart code:

```
1 import 'package:qr_flutter/qr_flutter.dart';
2
3 class HomeScreen extends StatefulWidget {
4   const HomeScreen({super.key});
5
6   @override
7   State<HomeScreen> createState() => _HomeScreenState();
8 }
9
10 class _HomeScreenState extends State<HomeScreen> {
11   @override
12   Widget build(BuildContext context) {
13     return Scaffold(
14       appBar: AppBar(
15         centerTitle: true,
16         backgroundColor: Theme.of(context).colorScheme.inversePrimary,
17         title: const Text("QR Flutter"),
18       ), // AppBar
19
20       body: Center(
21         child: QrImageView(
22           data: 'https://pub.dev/packages/qr_flutter',
23           version: QrVersions.auto,
24           size: 200.0,
25         ), // QrImageView
26       ), // Center
27     ); // Scaffold
28   }
29 }
30 
```

On the right, a mobile device icon displays the app's interface with the title "QR Flutter". Below the device icon is a large QR code generated from the URL in the code. The IDE interface includes a sidebar with "Device Manager", "Notifications", "Flutter Inspector", "Flutter Performance", and "Flutter Outline".

- Flutter Link previewer

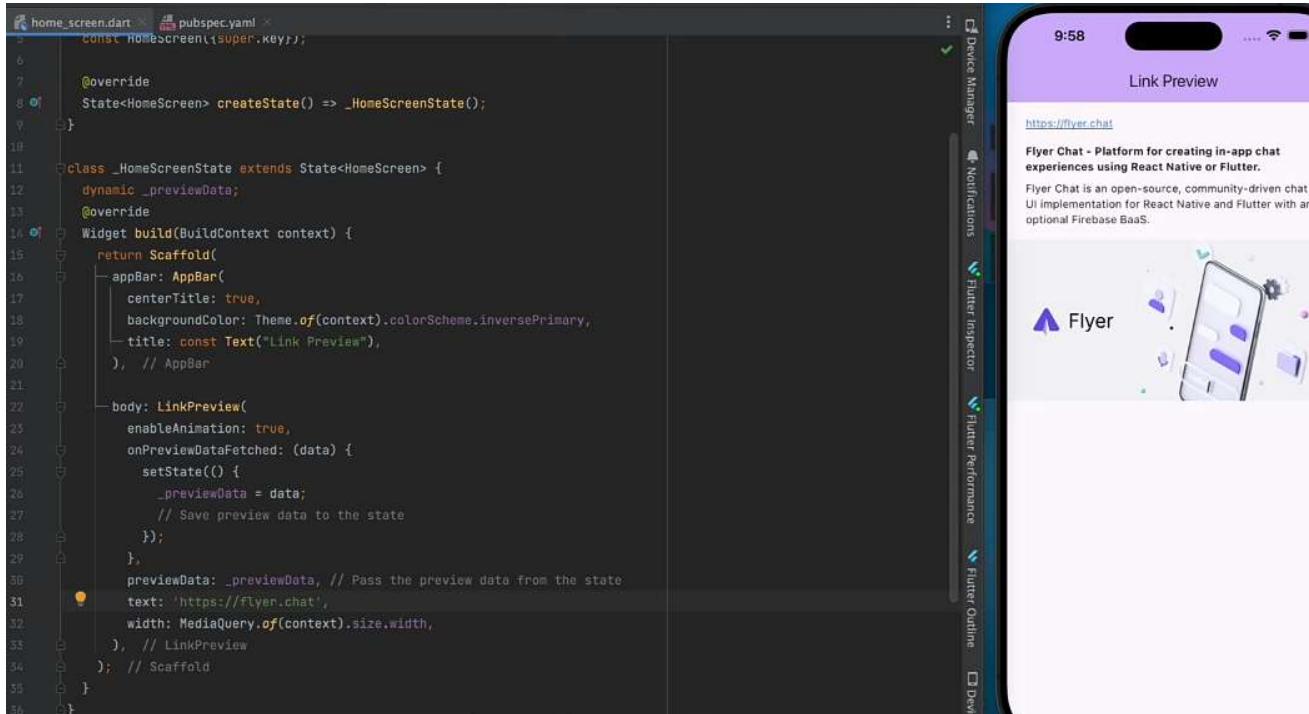
flutter_link_previewer 3.2.2  flutter_link_previewer: ^3.2.2 copied to clipboard

Published 11 months ago • ⚡ flyer.chat Dart 3 compatible

SDK FLUTTER PLATFORM ANDROID IOS LINUX MACOS WEB WINDOWS 277

Readme Changelog Example Installing Versions Scores

Flutter Link Previewer

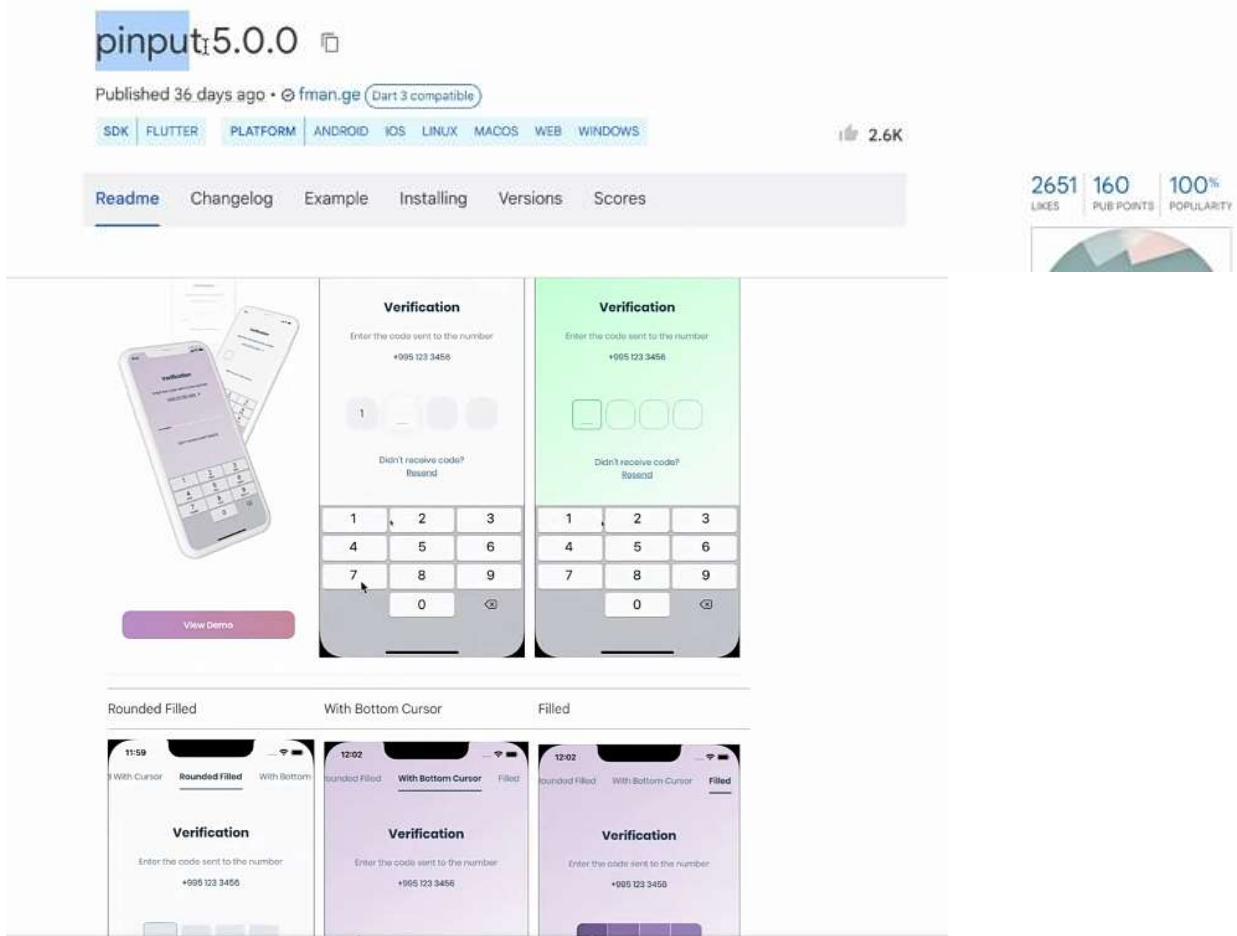


```
home_screen.dart pubspec.yaml
1 const HomeScreen({super.key});
2
3 @override
4 State<HomeScreen> createState() => _HomeScreenState();
5
6
7 class _HomeScreenState extends State<HomeScreen> {
8   dynamic _previewData;
9   @override
10  Widget build(BuildContext context) {
11    return Scaffold(
12      appBar: AppBar(
13        centerTitle: true,
14        backgroundColor: Theme.of(context).colorScheme.inversePrimary,
15        title: const Text("Link Preview"),
16      ), // AppBar
17
18      body: LinkPreview(
19        enableAnimation: true,
20        onPreviewDataFetched: (data) {
21          setState(() {
22            _previewData = data;
23            // Save preview data to the state
24          });
25        },
26        previewData: _previewData, // Pass the preview data from the state
27        text: 'https://flyer.chat',
28        width: MediaQuery.of(context).size.width,
29      ), // LinkPreview
30    ); // Scaffold
31  }
32}
```

Device Manager
Notifications
Flutter inspector
Flutter Performance
Flutter Online
Devic...

Link Preview
<https://flyer.chat>
Flyer Chat - Platform for creating in-app chat experiences using React Native or Flutter.
Flyer Chat is an open-source, community-driven chat UI implementation for React Native and Flutter with an optional Firebase BaaS.

- Otp verification ui flutter

pinput 5.0.0

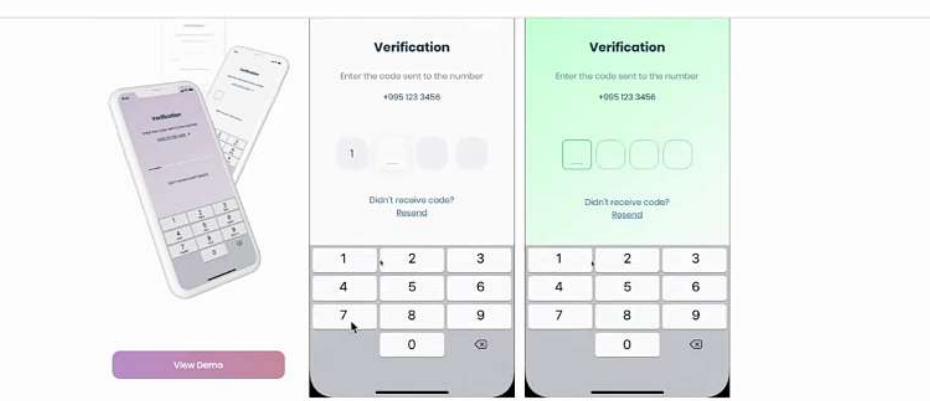
Published 36 days ago • fman.ge (Dart 3 compatible)

SDK FLUTTER PLATFORM ANDROID IOS LINUX MACOS WEB WINDOWS

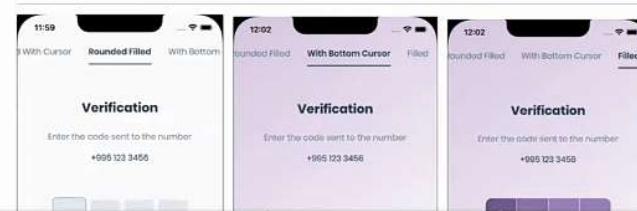
2.6K

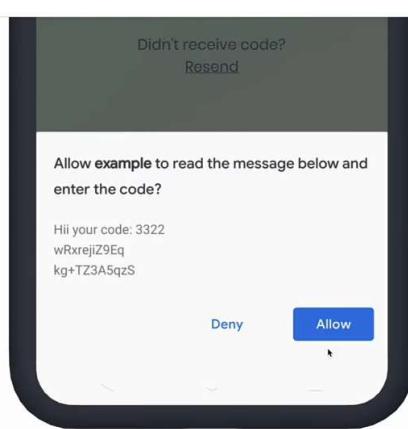
Readme Changelog Example Installing Versions Scores

2651 LIKES 160 PUB POINTS 100% POPULARITY



Rounded Filled With Bottom Cursor Filled



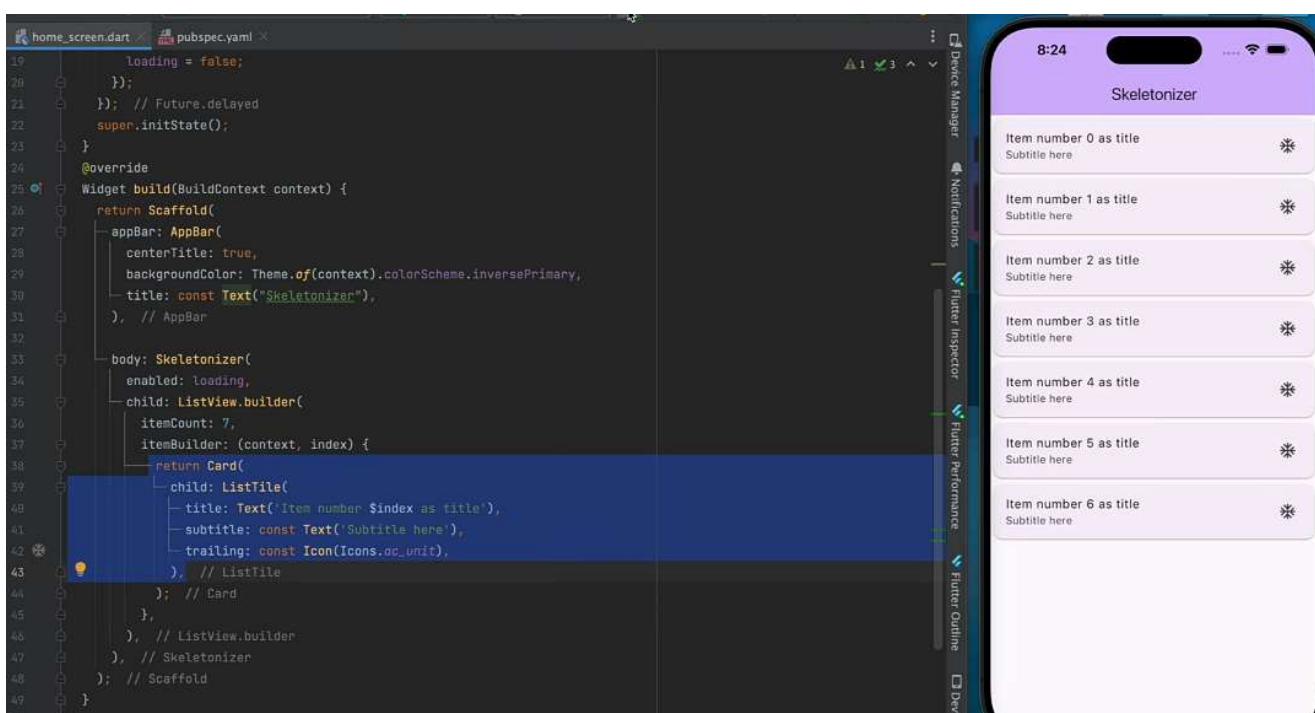
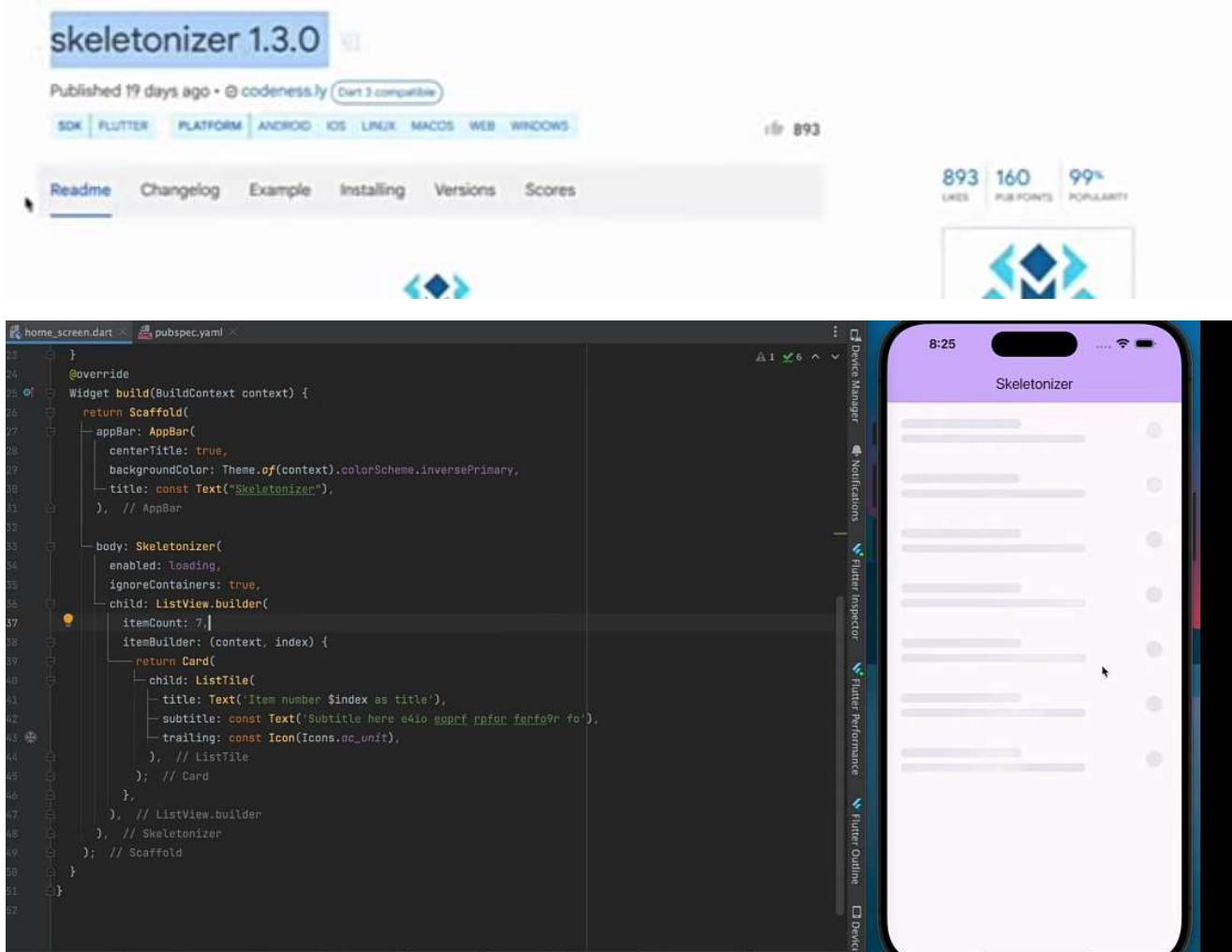


Output



```
home_screen.dart
57 children: [
58   Directionality(
59     // Specify direction if desired
60     textDirection: TextDirection.ltr,
61     child: Pinput(
62       controller: pinController,
63       focusNode: focusNode,
64       defaultPinTheme: defaultPinTheme,
65       separatorBuilder: (index) => const SizedBox(width: 8),
66       validator: (value) {
67         return value == '2222' ? null : 'Pin is incorrect';
68       },
69       hapticFeedbackType: HapticFeedbackType.lightImpact,
70       onCompleted: (pin) {
71         debugPrint('onCompleted: $pin');
72       },
73       onChanged: (value) {
74         debugPrint('onChanged: $value');
75       },
76       cursor: Column(
77         mainAxisAlignment: MainAxisAlignment.end,
78         children: [
79           Container(
80             margin: const EdgeInsets.only(bottom: 9),
81             width: 22,
82             height: 1,
83             color: focusedBorderColor,
84           ), // Container
85         ],
86       ), // Column
87       focusedPinTheme: defaultPinTheme.copyWith(
88         decoration: defaultPinTheme.decoration!.copyWith(
```

• Flutter loading effect



• Flutter page transition

page_transition 2.1.0

Published 10 months ago • [yasinilhan.com](#) Dart 3 compatible

SDK FLUTTER PLATFORM ANDROID IOS LINUX MACOS WEB WINDOWS

1.3K

Readme Changelog Example [Installing](#) Versions Scores

1324 LIKES 160 PUB POINTS 100% POPULARITY

[example/lib/main.dart](#)

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

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

/// Example App
class MyApp extends StatelessWidget {
  // This widget is the root of your application.
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      debugShowCheckedModeBanner: false,
      title: 'Flutter Demo',
      theme: ThemeData(
        primarySwatch: Colors.blue,
        pageTransitionsTheme: PageTransitionsTheme(builders: {
          TargetPlatform.iOS:
              PageTransition(type: PageTransitionType.fade, child: this)
            .matchingBuilder,
        })),
    home: MyHomePage(),
    onGenerateRoute: (settings) {
      switch (settings.name) {
```

Publisher [yasinilhan.com](#)

Metadata

Flutter Page Transition
Package which is the transition second page

Repository (GitHub)
[View/report issues](#)

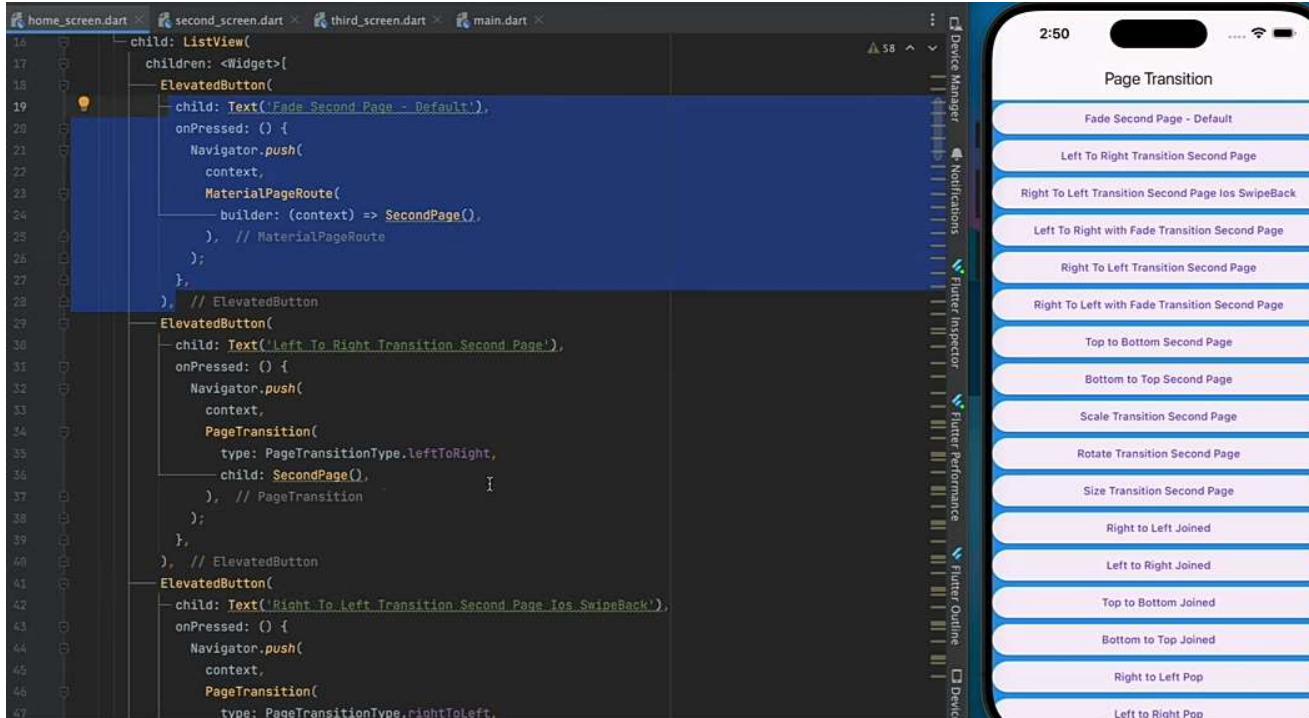
Documentation

API reference

License

[BSD-2-Clause \(license\)](#)

Dependencies



The screenshot shows the Flutter DevTools interface with the 'Device Manager' tab selected. On the left, there's a code editor window displaying the 'main.dart' file with several ElevatedButton widgets that demonstrate different page transition types. On the right, a large preview window titled 'Page Transition' displays a vertical list of 18 transition examples, each with a unique color and name. The transitions listed are: Fade Second Page - Default, Left To Right Transition Second Page, Right To Left Transition Second Page (ios SwipeBack), Left To Right with Fade Transition Second Page, Right To Left Transition Second Page, Right To Left with Fade Transition Second Page, Top to Bottom Second Page, Bottom to Top Second Page, Scale Transition Second Page, Rotate Transition Second Page, Size Transition Second Page, Right to Left Joined, Left to Right Joined, Top to Bottom Joined, Bottom to Top Joined, Right to Left Pop, and Left to Right Pop.

- Animate do

animate_do 3.3.4 

Published 3 months ago • fernando-herrera.com Dart 3 compatible

SDK FLUTTER PLATFORM ANDROID IOS LINUX MACOS WEB WINDOWS 4.2K

Readme Changelog Example Installing Versions Scores 4244 LIKES 160 PUB POINTS 99% POPULARITY

animate_do

Null-Safety, Dart 3, with zero external dependencies

iOS, Android, Linux, Mac, Web, Windows ready



ShakeY(
 from: 100,
 duration: const Duration(seconds: 5),
 infinite: true,
 child: const Square(),
) // ShakeY

FadeInUp(from: 0, child: const Square()),
FadeInUp(from: 50, child: const Square()),
FadeInUp(from: 100, child: const Square()),
FadeInUp(from: 150, child: const Square()),
FadeInUp(from: 200, child: const Square()),
FadeInUp(from: 250, child: const Square()),

BounceInDown(from: 0, child: const Square()),
BounceInDown(from: 50, child: const Square()),
BounceInDown(from: 100, child: const Square()),
BounceInDown(from: 150, child: const Square()),
BounceInDown(from: 200, child: const Square()),
BounceInDown(from: 250, child: const Square()),

The screenshot shows a code editor on the left with Dart code for a home screen. The code includes imports for 'package:flutter/material.dart' and 'package:animate_do/animate_do.dart'. It defines a state class '_HomeScreenState' that extends 'State<HomeScreen>'. The build method returns a 'Scaffold' with an 'AppBar' containing a title 'animate_do'. The 'body' of the scaffold contains a 'Center' widget with a 'Row' of four purple containers. The 'Row' has 'mainAxisAlignment: MainAxisAlignment.spaceAround'. Each container has a height of 100 and a width of 100, colored purple. To the right of the code editor is a mobile phone simulation displaying the app. The phone's status bar shows the time as 9:39 and battery level. The app's title bar is labeled 'animate_do'. The main screen shows the four purple containers arranged horizontally with space between them.

```
10 }
11 
12 class _HomeScreenState extends State<HomeScreen> {
13 
14     @override
15     Widget build(BuildContext context) {
16         return Scaffold(
17             appBar: AppBar(
18                 centerTitle: true,
19                 backgroundColor: Theme.of(context).colorScheme.inversePrimary,
20                 title: const Text("animate_do"),
21             ), // AppBar
22 
23             body: Center(
24 
25                 child: Row(
26                     mainAxisAlignment: MainAxisAlignment.spaceAround,
27                     children: <Widget>[
28 
29                         FadeInLeft(child: Container(height: 100, width: 100, color: Colors.purple,)),
30                         FadeInUp(child: Container(height: 100, width: 100, color: Colors.purple,)),
31                         FadeInDown(child: Container(height: 100, width: 100, color: Colors.purple,)),
32                         FadeInRight(child: Container(height: 100, width: 100, color: Colors.purple,)),
33 
34                     ],
35                 ), // Row
36 
37             ), // Center
38         ); // Scaffold
39     }
40 }
```

- Animated search bar



A screenshot of a Flutter development environment. On the left, the code editor shows `main.dart`, `home_screen.dart`, and `pubspec.yaml`. The `home_screen.dart` file contains the following code:

```
5
6  const HomeScreen({super.key});
7
8  @override
9  State<HomeScreen> createState() => _HomeScreenState();
10
11 class _HomeScreenState extends State<HomeScreen> {
12   TextEditingController textController = TextEditingController();
13
14   @override
15   Widget build(BuildContext context) {
16     return Scaffold(
17       appBar: AppBar(
18         centerTitle: true,
19         backgroundColor: Theme.of(context).colorScheme.inversePrimary,
20         title: const Text("Animated Search Bar"),
21       ), // AppBar
22       body: Padding(
23         padding: const EdgeInsets.all(20),
24         child: AnimSearchBar(
25           width: 400,
26           textController: textController,
27           onSuffixTap: () {
28             setState(() {
29               textController.clear();
30             });
31           },
32         ),
33         onSubmitted: (String value) {
34           ...
35         }
36       );
37     }
38   }
39 }
```

The right side shows a preview of the app running on an iPhone. The screen has a purple header bar with the title "Animated Search Bar". Below it is a white search bar with a magnifying glass icon. The main body of the screen is a light gray color.

A screenshot of a Flutter development environment, similar to the one above. The code editor shows `main.dart`, `home_screen.dart`, and `pubspec.yaml`. The `home_screen.dart` file contains the same code as the first screenshot.

The right side shows a preview of the app running on an iPhone. The screen has a purple header bar with the title "Animated Search Bar". Below it is a white search bar with a magnifying glass icon. The search bar has the placeholder text "Search..." and a clear button (an 'X' icon).

- Sticky header

sticky_headers 0.3.0+2

Published 2 years ago • ⚭ fluttercommunity.dev Dart 3 compatible

SDK FLUTTER PLATFORM ANDROID IOS LINUX MACOS WEB WINDOWS

1️⃣ 865

[Readme](#) [Changelog](#) [Example](#) [Installing](#) [Versions](#) [Scores](#)

Use it:

```
class Example extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return ListView.builder(itemBuilder: (context, index) {
      return StickyHeader(
        header: Container(
          height: 50.0,
          color: Colors.blueGrey[700],
          padding: EdgeInsets.symmetric(horizontal: 16.0),
          alignment: Alignment.centerLeft,
          child: Text('Header #$index',
            style: const TextStyle(color: Colors.white),
          ),
        ),
        content: Container(
          child: Image.network(
            imageForIndex(index),
            fit: BoxFit.cover,
            width: double.infinity,
            height: 200.0,
          ),
        ),
      );
    });
  }
}
```

• Deeplinking in flutter

app_links 6.3.3

Published 16 days ago • [@openapi4j.org](#) Dart 3 compatible

SDK FLUTTER PLATFORM ANDROID IOS LINUX MACOS WEB WINDOWS

1 879

[Readme](#) [Changelog](#) [Example](#) [Installing](#) [Versions](#) [Scores](#)

app_links

Android App Links, Deep Links, iOS Universal Links and Custom URL schemes handler (desktop included linux, macOS, Windows).

This plugin allows you to open your app from:

- HTTPS URLs instead of the browser.
- custom schemes.

Getting Started

Before using the plugin, you'll need to setup each platform you target.

All those configurations below are also accessible in the example project.

- Android
- iOS

• Flutter countdown timer

slide_countdown 2.0.0

Published 22 days ago • [@farhanfadila.site](#) Dart 3 compatible • Latest: 2.0.0 / Prerelease: 2.0.1-dev+1

SDK FLUTTER PLATFORM ANDROID IOS LINUX MACOS WEB WINDOWS

1 375

[Readme](#) [Changelog](#) [Example](#) [Installing](#) [Versions](#) [Scores](#)

375 | 160 | 97%
LIKES PUB POINTS POPULARITY

⌚ Slide Countdown

A Flutter package to create easy slide animation countdown / countup timer

[Platform](#) [Flutter](#) [Donate](#) [PayPal](#) [Donate](#) [farhanfadila](#)

- ⏳ Support Count down and Count up
- ⏱ Control duration
- 🎉 Callback finished
- 🎁 Easily custom duration layout with `RawSlideCountdown` and `RowDigitItem` widget.

Thanks to `pausable_timer`, this package use `pausable_timer` for helping control duration.

Please likes for support this package 



Publisher

[@farhanfadila.site](#)

Metadata

A Flutter package to create easy slide animation countdown / countup timer.

The screenshot shows an IDE interface with two main panes. The left pane displays the Dart code for a `Countdown` screen, and the right pane shows a preview of the app running on an iPhone.

Code (home_screen.dart):

```
14  Widget build(BuildContext context) {  
15    return Scaffold(  
16      appBar: AppBar(  
17        backgroundColor: Theme.of(context).colorScheme.inversePrimary,  
18        title: const Text('Countdown'),  
19      ), // AppBar  
20      body: Center(  
21        child: const Column(  
22          mainAxisAlignment: MainAxisAlignment.center,  
23          crossAxisAlignment: CrossAxisAlignment.start,  
24          children: [  
25            SlideCountdown(  
26              duration: Duration(seconds: 30, minutes: 10, hours: 1, days: 1),  
27            ), // SlideCountdown  
28            SizedBox(height: 20),  
29            SlideCountdownSeparated(  
30              duration: Duration(days: 2),  
31            ), // SlideCountdownSeparated  
32            ...  
33          ], // Column  
34        ), // Center  
35      ); // Scaffold  
36    }  
37  }  
38
```

Preview (iPhone):

The preview shows a purple header bar with the text "Countdown". Below it is a white content area. In the top right corner of the content area, there is a red rounded rectangle containing the text "01:01:10:27". At the bottom of the content area, there are four smaller red rounded rectangles arranged horizontally, containing the numbers "01", "23", "59", and "48".