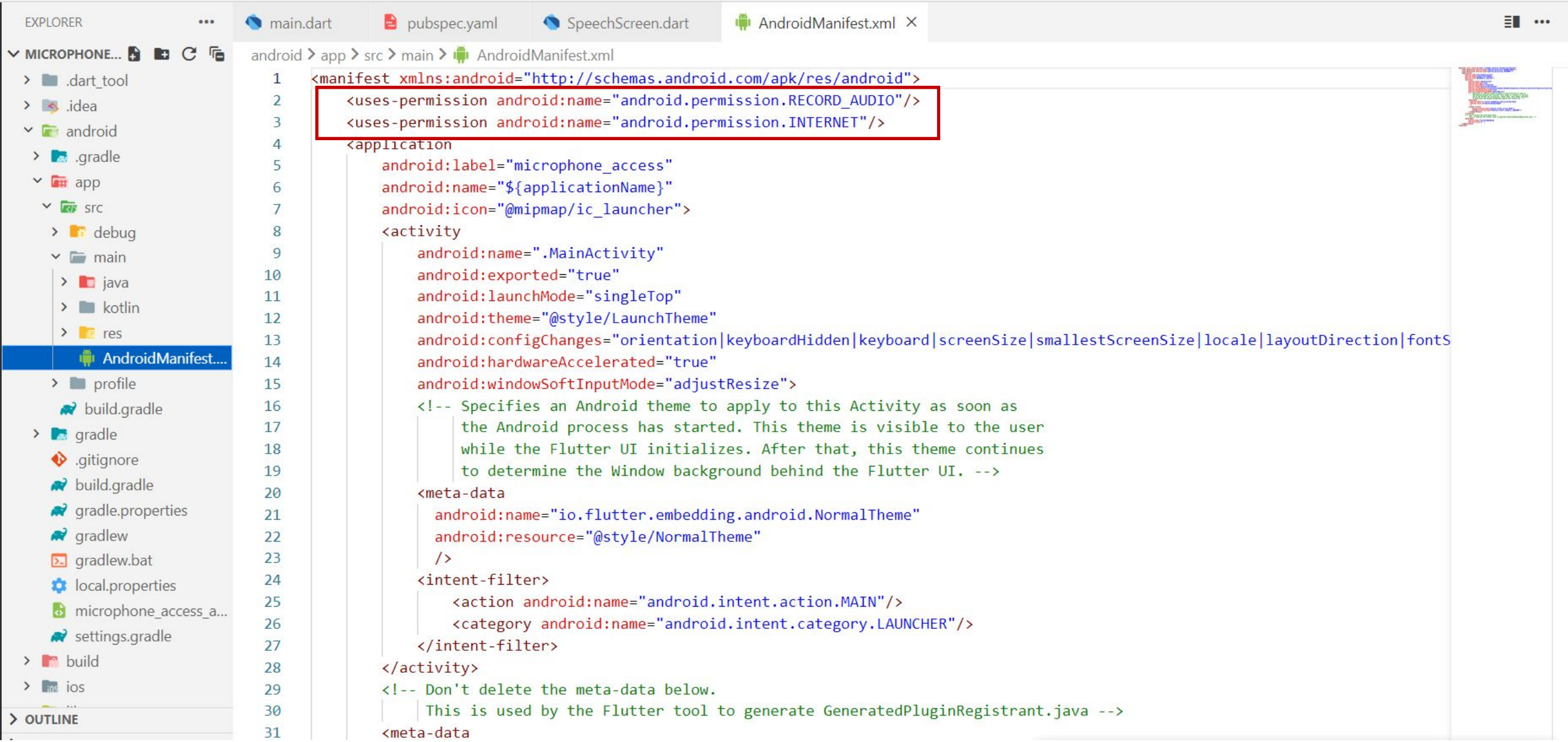
Accessing Hardware Components Using Flutter (accessing microphone)

main.dart | pubspec.yaml

pubspec.yaml

```yaml
sdk: '>=3.1.0 <4.0.0'

# Dependencies specify other packages that your package needs in order to work.
# To automatically upgrade your package dependencies to the latest versions
# consider running `flutter pub upgrade --major-versions`. Alternatively,
# dependencies can be manually updated by changing the version numbers below to
# the latest version available on pub.dev. To see which dependencies have newer
# versions available, run `flutter pub outdated`.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
  avatar_glow: ^2.0.2
  speech_to_text: ^6.3.0

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: ^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.
```


EXPLORER
main.dart
pubspec.yaml
SpeechScreen.dart
AndroidManifest.xml
android > app > src > main > AndroidManifest.xml
MICROPHONE...
.dart_tool
.idea
android
.gradle
app
src
debug
main
java
kotlin
res
AndroidManifest....
profile
build.gradle
gradle
.gitignore
build.gradle
gradle.properties
gradlew
gradlew.bat
local.properties
microphone_access_a...
settings.gradle
build
ios
OUTLINE
1 <manifest xmlns:android="http://schemas.android.com/apk/res/android">
2 <uses-permission android:name="android.permission.RECORD_AUDIO"/>
3 <uses-permission android:name="android.permission.INTERNET"/>
4 <application
5 android:label="microphone_access"
6 android:name="${applicationName}"
7 android:icon="@mipmap/ic_launcher">
8 <activity
9 android:name=".MainActivity"
10 android:exported="true"
11 android:launchMode="singleTop"
12 android:theme="@style/LaunchTheme"
13 android:configChanges="orientation|keyboardHidden|keyboard|screenSize|smallestScreenSize|locale|layoutDirection|fontS
14 android:hardwareAccelerated="true"
15 android:windowSoftInputMode="adjustResize">
16 <!-- Specifies an Android theme to apply to this Activity as soon as
17 the Android process has started. This theme is visible to the user
18 while the Flutter UI initializes. After that, this theme continues
19 to determine the Window background behind the Flutter UI. -->
20 <meta-data
21 android:name="io.flutter.embedding.android.NormalTheme"
22 android:resource="@style/NormalTheme"
23 />
24 <intent-filter>
25 <action android:name="android.intent.action.MAIN"/>
26 <category android:name="android.intent.category.LAUNCHER"/>
27 </intent-filter>
28 </activity>
29 <!-- Don't delete the meta-data below.
30 This is used by the Flutter tool to generate GeneratedPluginRegistrant.java -->
31 <meta-data

Main.dart

main.dart

lib > main.dart > MyApp > build

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

Run | Debug | Profile
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(
      home: SpeechScreen(),
    ); // MaterialApp
  }
}

```

Using

To recognize text from the microphone import the package and call the plugin, like so:

Minimal

```
import 'package:speech_to_text/speech_to_text.dart' as stt;

    stt.SpeechToText speech = stt.SpeechToText();
    bool available = await speech.initialize( onStatus: statusListener, onError: errorListe
    if ( available ) {
        speech.listen( onResult: resultListener );
    }
    else {
        print("The user has denied the use of speech recognition.");
    }
    // some time later...
    speech.stop()
```

Complete Flutter example

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

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

class MyApp extends StatelessWidget {
  @override
```

Dependencies

clock, flutter, flutter_web_plugins, js, json_annotation, meta, pedantic, speech_to_text_macos, speech_to_text_platform_interface

More

Packages that depend on speech_to_text

main.dart | pubspec.yaml | SpeechScreen.dart

lib > screens > SpeechScreen.dart > _SpeechScreenState > speechToText

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

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

  @override
  State<SpeechScreen> createState() => _SpeechScreenState();
}

class _SpeechScreenState extends State<SpeechScreen> {

  SpeechToText speechToText = SpeechToText();
  var text = "Hold the button and start speaking";
  var isListening = false;

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      floatingActionButtonLocation: FloatingActionButtonLocation.centerFloat,
      floatingActionButton: AvatarGlow(
        endRadius: 75.0,
        animate : isListening,
        duration:Duration(milliseconds:2000),
        glowColor : Colors.red,
        repeat : true,
```

main.dart | pubspec.yaml | SpeechScreen.dart

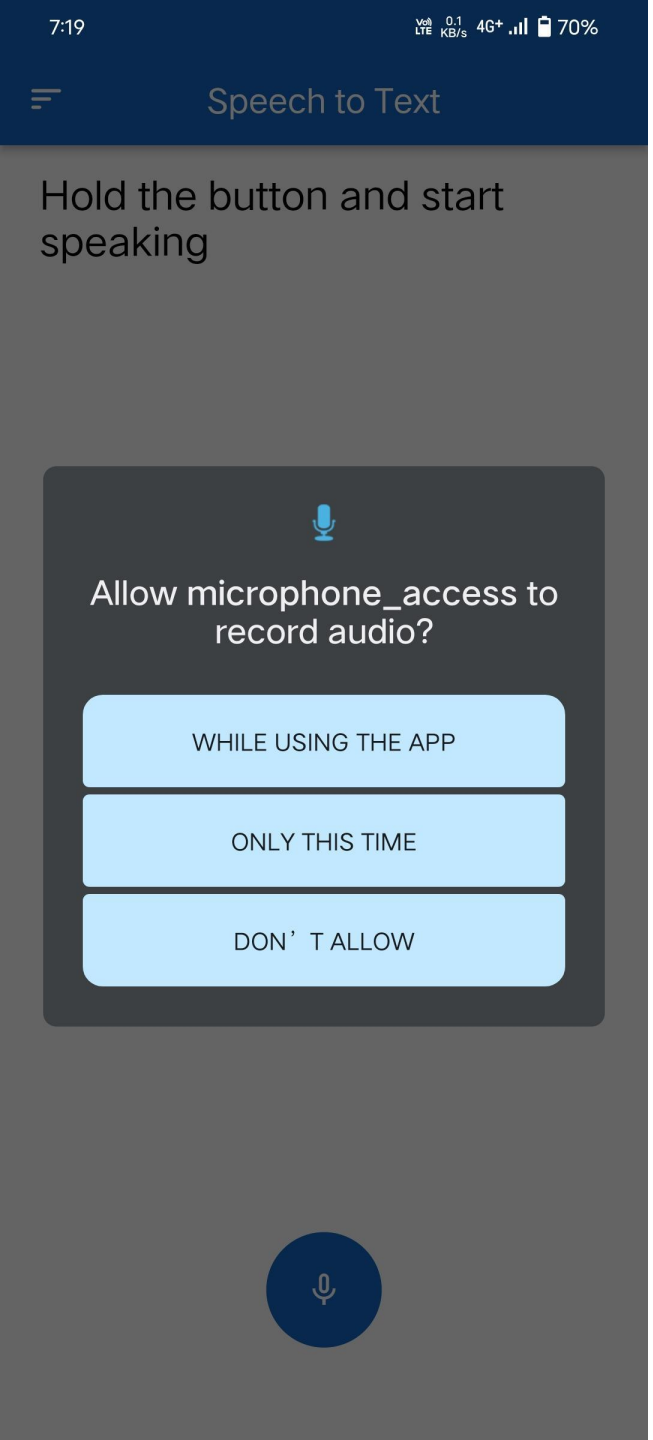
lib > screens > SpeechScreen.dart > _SpeechScreenState > build

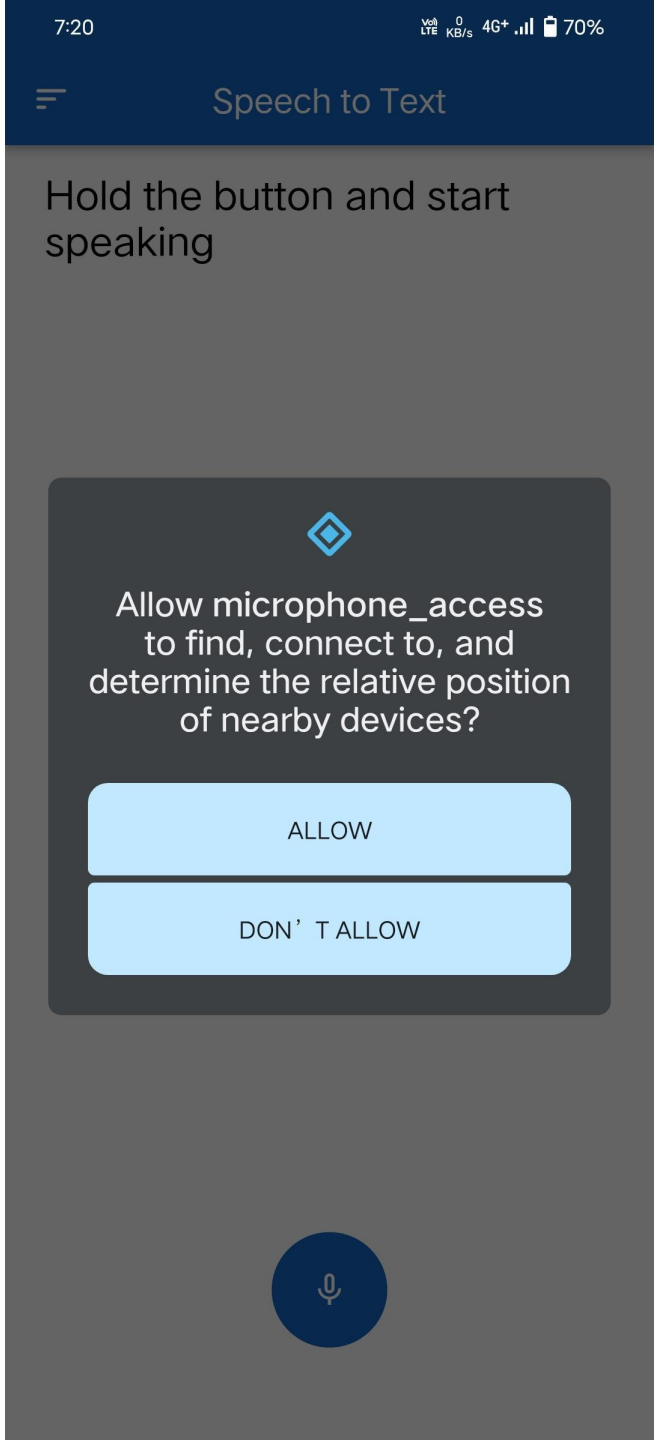
```
26          glowColor : Colors.red,
27          repeat : true,
28          repeatPauseDuration:Duration(milliseconds:100),
29          showTwoGlows : true,
30          child: GestureDetector(
31            onTapDown: (details) async {
32              if(!isListening)
33                {
34                  var available = await speechToText.initialize();
35
36                  if(available)
37                    {
38                      setState(() {
39                        isListening = true;
40                        speechToText.listen(
41                          onResult: (result)
42                              {
43                                setState((){
44                                  text = result.recognizedWords;
45                                });
46                              }
47                        );
48                      });
49                    }
50                }
51
52            },
53            onTapUp: (details){
54              setState(() {
55                isListening = false;
56              });
57              speechToText.stop();
58            },
59
```

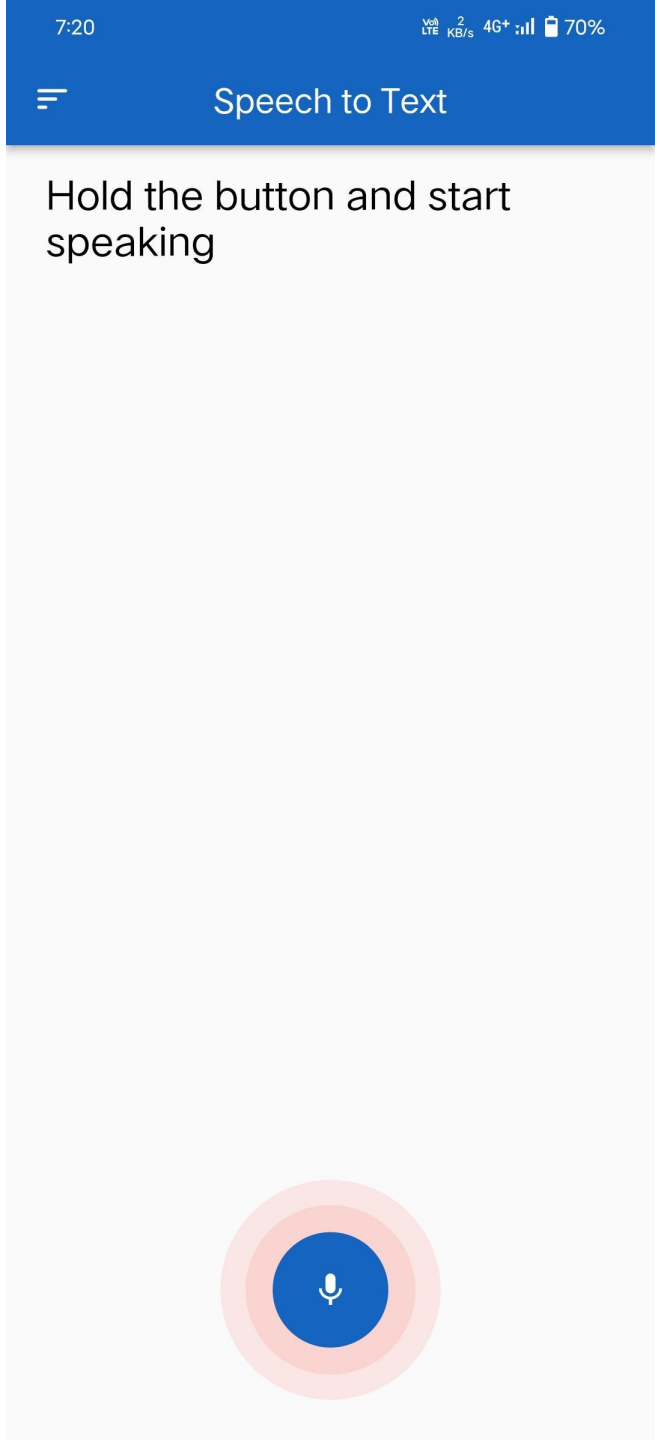
main.dart | pubspec.yaml | SpeechScreen.dart

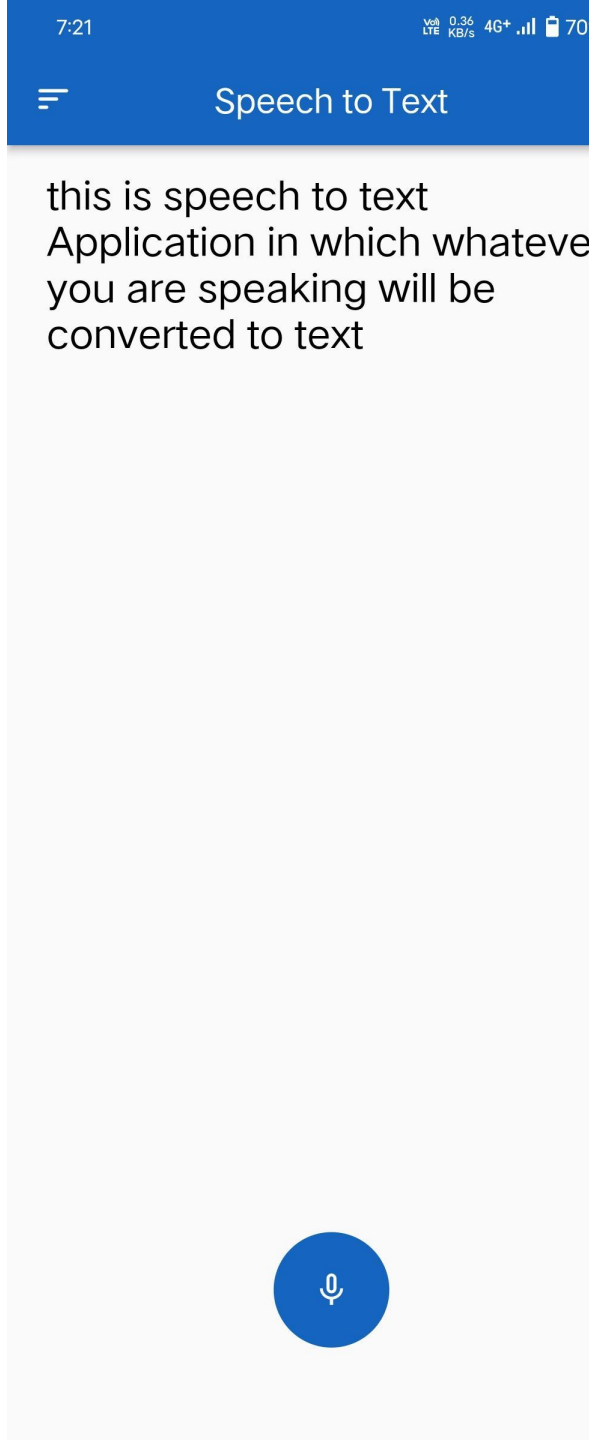
lib > screens > SpeechScreen.dart > _SpeechScreenState > build

```dart
            isListening = false;
          });
          speechToText.stop();
        },

        child: CircleAvatar(
          backgroundColor: Colors.blue.shade800,
          radius : 35,
          child : Icon(isListening ? Icons.mic : Icons.mic_none, color : Colors.white),

        ), // CircleAvatar
      ), // GestureDetector
    ), // AvatarGlow
    appBar : AppBar(
      backgroundColor: Colors.blue.shade800,
      leading : const Icon(Icons.sort_rounded, color : Colors.white),
      title : Text("Speech to Text"),
      centerTitle: true,
    ), // AppBar
    body : Container(

      padding : const EdgeInsets.symmetric(horizontal: 24, vertical : 16),
      margin: const EdgeInsets.only(bottom:150.0),
      child : Text(
        text,
        style : const TextStyle(fontSize: 24, color : Colors.black),
      ) // Text
    ) // Container
  ); // Scaffold
  }
}

```


7:19
70%
Speech to Text
Hold the button and start speaking
Allow microphone_access to record audio?
WHILE USING THE APP
ONLY THIS TIME
DON’T ALLOW

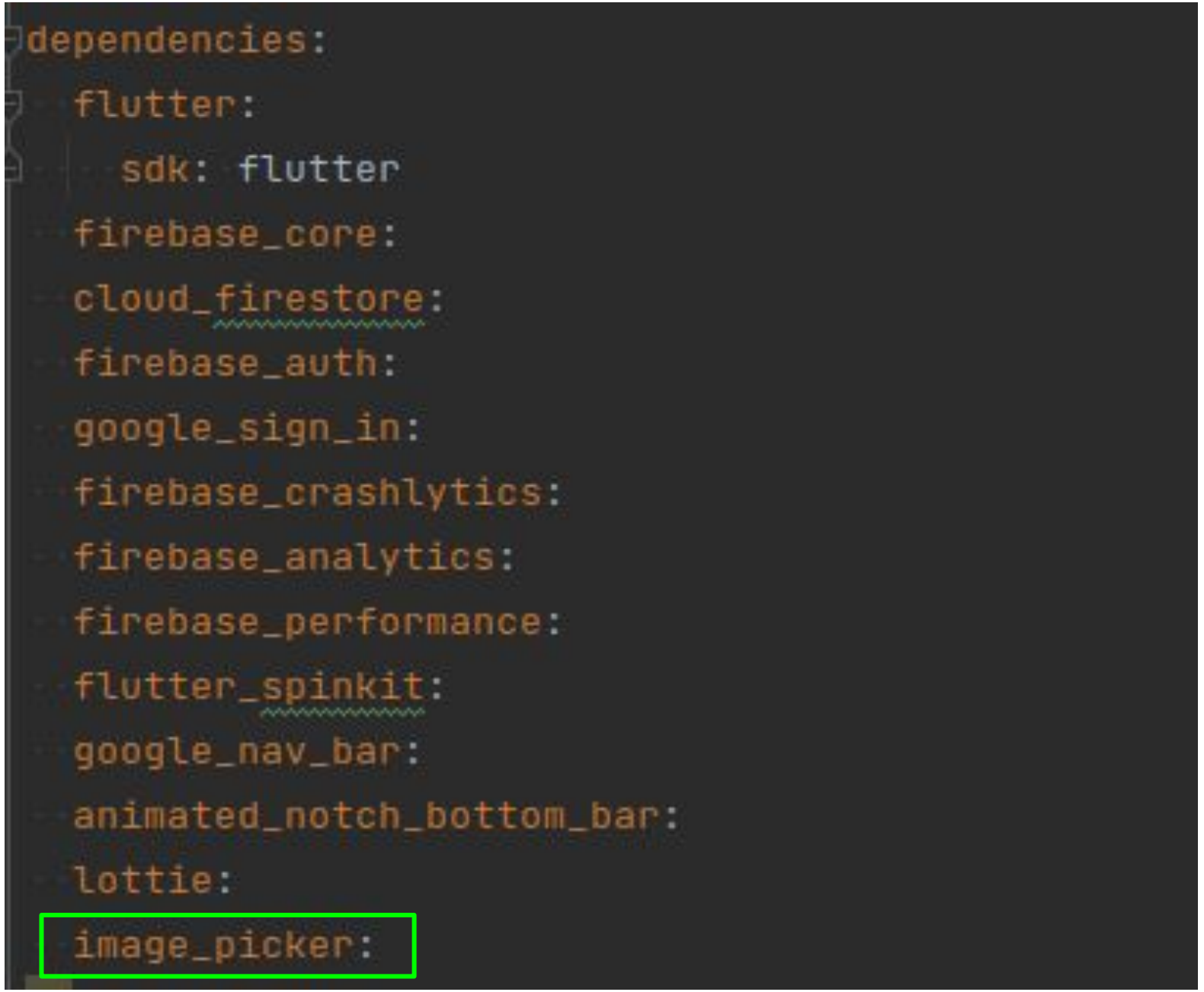

7:20
70%
Speech to Text
Hold the button and start speaking
Allow microphone_access to find, connect to, and determine the relative position of nearby devices?
ALLOW
DON’T ALLOW


7:20
70%
Speech to Text
Hold the button and start speaking


7:21
Speech to Text
this is speech to text Application in which whateve you are speaking will be converted to text

Thank You

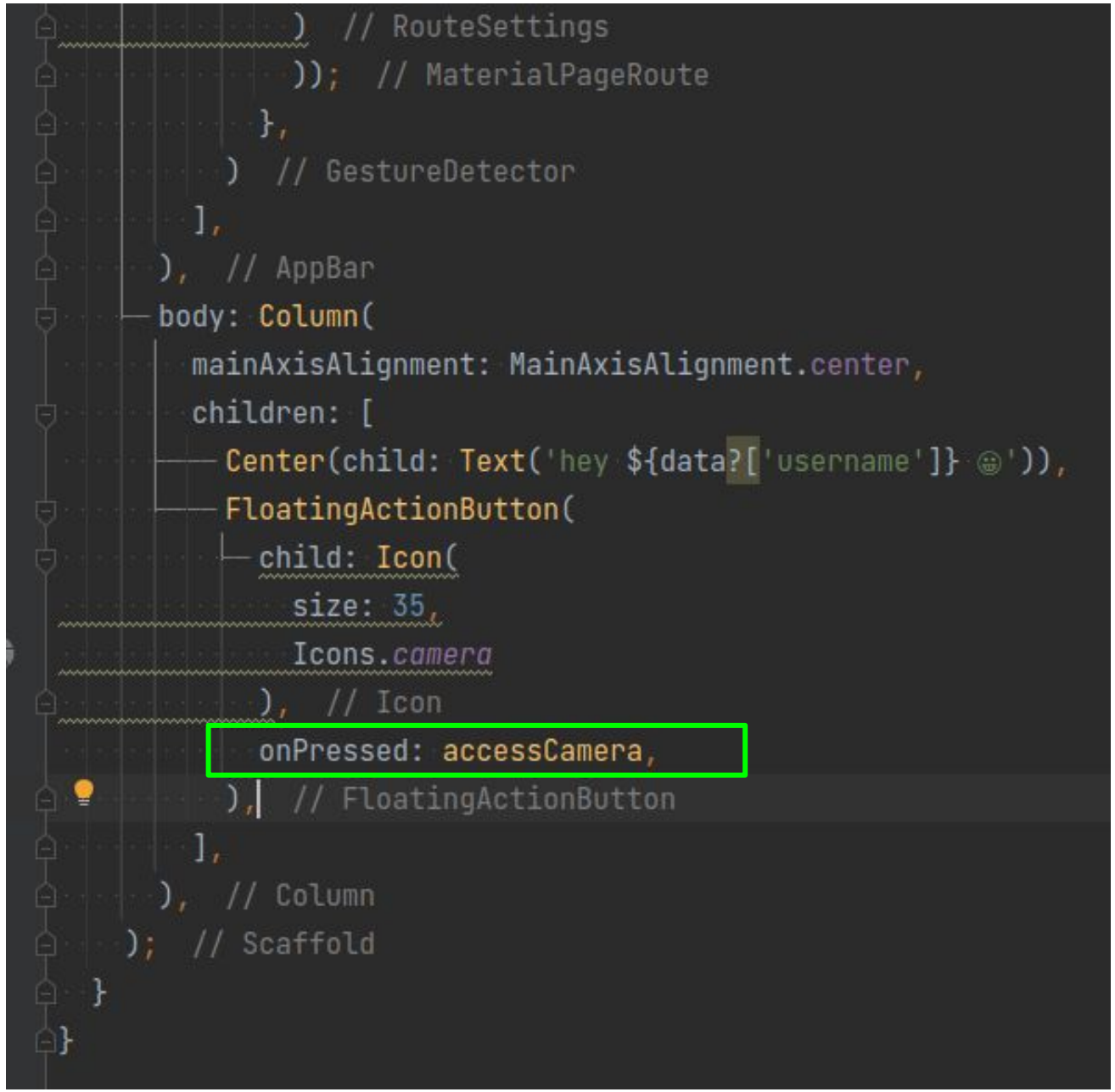
Accessing Hardware Components Using Flutter (accessing camera)

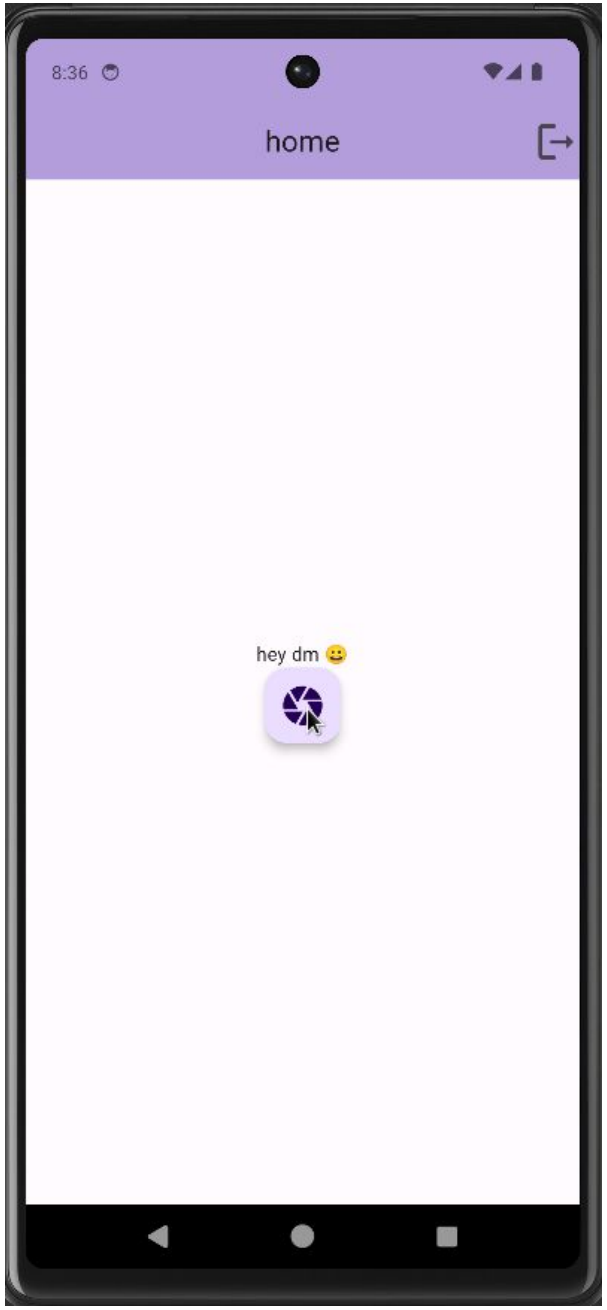

dependencies:
flutter:
sdk: flutter
firebase_core:
cloud_firestore:
firebase_auth:
google_sign_in:
firebase_crashlytics:
firebase_analytics:
firebase_performance:
flutter_spinkit:
google_nav_bar:
animated_notch_bottom_bar:
lottie:
image_picker:

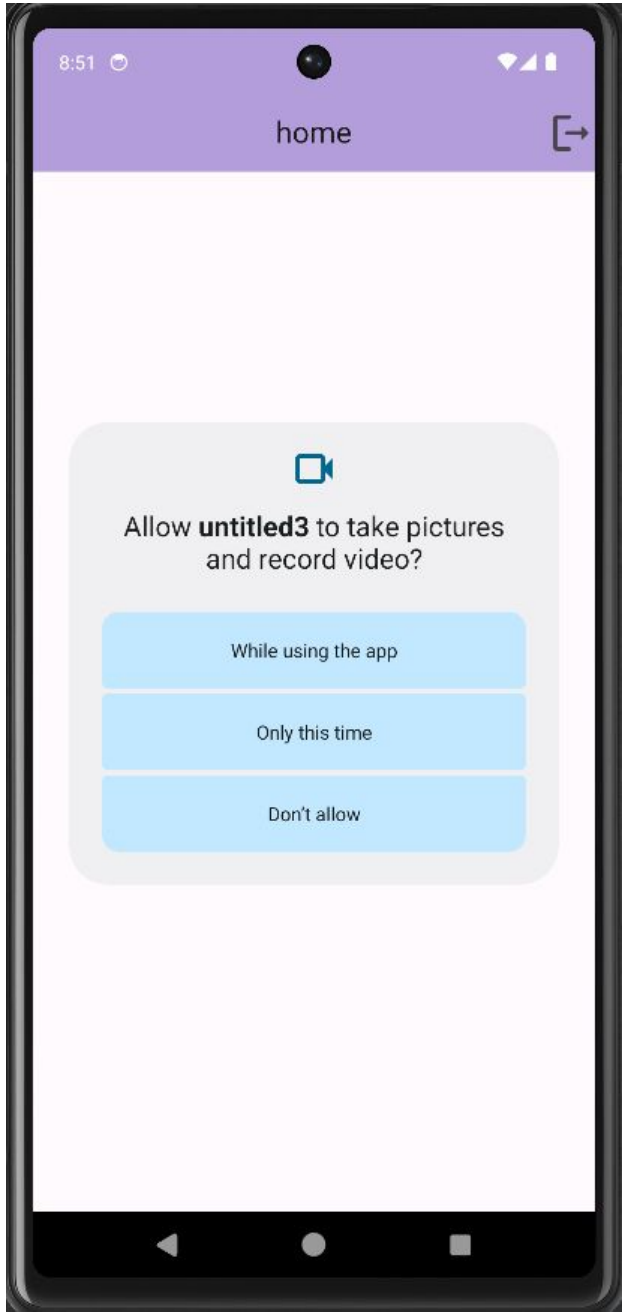
app/src/main/AndroidMainfest.xml

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android">
    <uses-feature
        android:name="android.hardware.camera"
        android:required="true" />
    <uses-permission android:name="android.permission.CAMERA"/>
    <application
        android:label="untitled3"
        android:name="${applicationName}"
        android:icon="@mipmap/ic_launcher">
        <activity
            android:name=".MainActivity"
            android:exported="true"
            android:launchMode="singleTop"
            android:theme="@style/LaunchTheme"
            android:configChanges="orientation|keyboardHidden|keyboard|screenSize|smallestScreen
            android:hardwareAccelerated="true"
            android:windowSoftInputMode="adjustResize">
```

```dart
class _HomeState extends State<Home> {
  XFile? cameraFile;
  @override
  void initState() {
    // TODO: implement initState
    super.initState();
  }
  void accessCamera() async{
    cameraFile=await ImagePicker().pickImage(source: ImageSource.camera);
  }
  @override
  Widget build(BuildContext context) {
  final data = ModalRoute.of(context)!.settings.arguments as Map;
    return Scaffold(
      appBar: AppBar(
        title: Text('home'),
        centerTitle: true,
        backgroundColor: Colors.deepPurple.shade200,
        actions: [
          GestureDetector(
            child: Icon(
              Icons.logout_rounded,
```


) // RouteSettings
)); // MaterialPageRoute
}
) // GestureDetector
],
), // AppBar
body: Column(
mainAxisAlignment: MainAxisAlignment.center,
children: [
Center(child: Text('hey ${data?['username']} 😀')),
FloatingActionButton(
child: Icon(
size: 35,
Icons.camera
), // Icon
onPressed: accessCamera,
), // FloatingActionButton
],
), // Column
); // Scaffold
}
}


8:36
home
hey dm 😀


8:51
home
Allow **untitled3** to take pictures and record video?
While using the app
Only this time
Don't allow

Thank You