

Using Firebase with Flutter



Firebase

Products ▾

Solutions ▾

Pricing

Docs ▾

Community ▾

Support

Make your app the best it can be

Firebase is an app development platform that helps you build and grow apps and games users love. Backed by Google and trusted by millions of businesses around the world.

Get started

Try demo

Watch video

× Create a project (Step 1 of 3)

Let's start with a name for
your project^(?)

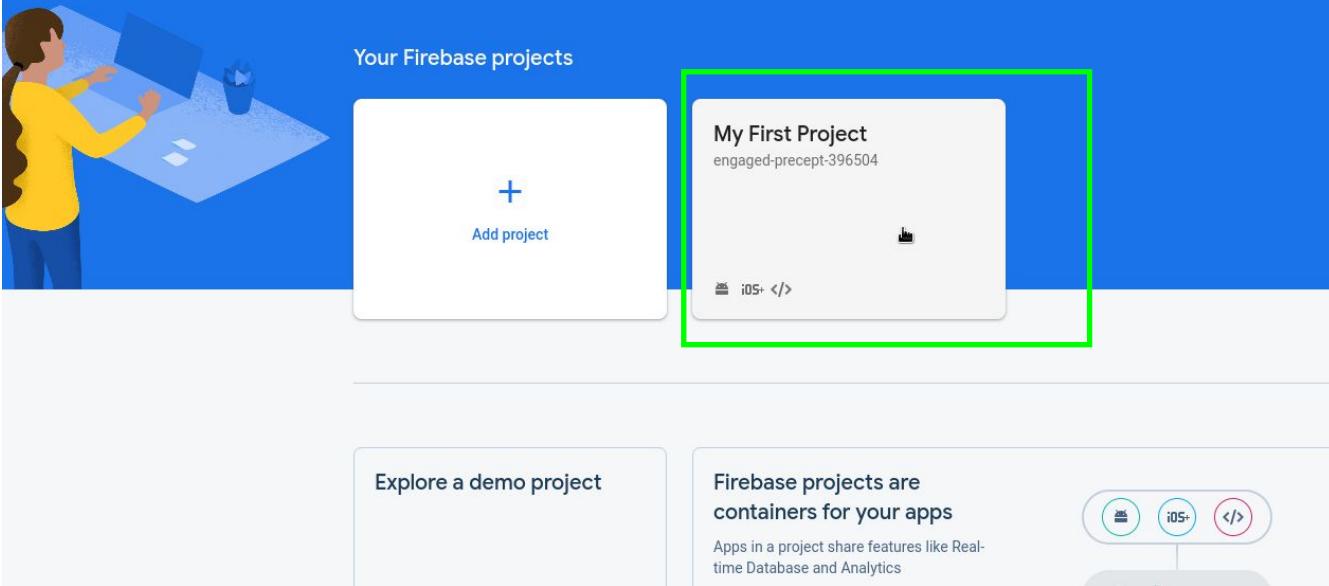
Project name

first firebase project

first-firebase-project-9c824

Continue

Your Firebase projects



My First Project
engaged-precept-396504

iOS+ </>

Add project

Explore a demo project

Firebase projects are containers for your apps

Apps in a project share features like Real-time Database and Analytics

>Your apps

Android apps

-  com.example.untitled3
-  untitled (android)
com.example.untitled
-  untitled1 (android)
com.example.untitled1

Apple apps

-  untitled (ios)

SDK setup and configuration

Need to reconfigure the Firebase SDKs for your app? Revisit the SDK setup instructions or just download the configuration file containing keys and identifiers for your app.

[See SDK instructions](#) [google-services.json](#)

App ID [?](#)
1:951056515194:android:d30cc8b2b833955e753bc3

App nickname

Firebase

Project Overview

Project settings

Users and permissions

Usage and billing

Project shortcuts

Firestore Database

Product categories

Build

Release & Monitor

Analytics

Engage

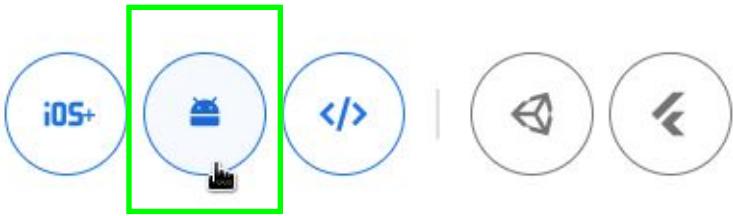
All products

My First Project ▾

Add Firebase to your app

X

Select a platform to get started



Add app

? Revisit the SDK setup instructions or just
| identifiers for your app.

services.json

App ID ⓘ

A screenshot of an IDE showing a project structure. The 'build.gradle' file is open in the center, with several lines of code visible. A red box highlights the 'applicationId' line, which is also highlighted in a larger red box in the bottom right corner of the image. The 'google-services.json' file is also visible in the project tree.

```
37 }  
38 }  
39 kotlinOptions {  
40     jvmTarget = '1.8'  
41 }  
42 sourceSets {  
43     main.java.srcDirs += 'src/main/kotlin'  
44 }  
45 }  
46  
47 defaultConfig {  
48     // TODO: Specify your own unique Application ID (https://developer.android.com/studio)  
49     applicationId "com.example.untitled3"  
50     // You can update the following values to match your application needs.  
51     // For more information, see: https://docs.flutter.dev/deployment/android#reviewing-  
52     minSdkVersion 19  
53     targetSdkVersion 28  
54     versionCode flutterVersionCode.toInt()
```

Add Firebase to your Android app

1 Register app

Android package name ⓘ

com.example.untitled3

App nickname (optional) ⓘ

flutter-app


```
dependencies {  
    classpath 'com.android.tools.build:gradle:7.3.0'  
    classpath "org.jetbrains.kotlin:kotlin-gradle-plugin:$kotlin_version"  
    classpath "com.google.gms:google-services:4.3.15"  
}
```

=> android/build.gradle

```
apply plugin: 'com.android.application'  
apply plugin: 'com.google.gms.google-services'  
apply plugin: 'kotlin-android'  
apply from: "$flutterRoot/packages/flutter_tools/gradle/flutter.gradle"
```

```
an unique Application ID (https://developer.android.com/studio/build/application-id.html).  
applicationId "com.example.untitled3"  
// You can update the following values to match your application needs.  
// For more information, see: https://docs.flutter.dev/deployment/android#reviewing-the-gradle-build-configuration.  
minSdkVersion 19  
targetSdkVersion 28  
versionCode flutterVersionCode.toInteger()  
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' works.  
        signingConfig signingConfigs.debug  
    }  
}  
  
flutter {  
    source '../..'  
}  
  
dependencies {  
    implementation "org.jetbrains.kotlin:kotlin-stdlib-jdk7:$kotlin_version"  
    implementation 'com.android.support:multidex:1.0.3'  
}
```

```
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:
```

pubspec.yaml × main.dart × home.dart × login.dart × signup.dart × AndroidManifest.xml ×

```
1 import 'package:flutter/material.dart';
2 import 'package:untitled3/pages/home.dart';
3 import 'package:untitled3/pages/login.dart';
4 import 'package:untitled3/pages/signup.dart';
5 import 'package:firebase_core/firebase_core.dart';
6 import 'package:google_nav_bar/google_nav_bar.dart';
7 import 'package:animated_notch_bottom_bar/animated_notch_bottom_bar/animated_notch_bottom_bar.dart'
8 import 'package:lottie/lottie.dart';
9
10 void main() {
11   WidgetsFlutterBinding.ensureInitialized();
12   Firebase.initializeApp();
13   runApp(const MyApp());
14 }
```

```
Launching lib/main.dart on sdk gphone64 x86 64 in debug mode...
Running Gradle task 'assembleDebug'...
✓ Built build/app/outputs/flutter-apk/app-debug.apk.
Installing build/app/outputs/flutter-apk/app-debug.apk...
Debug service listening on ws://127.0.0.1:40999/XFN_jE3M00g=/ws
Syncing files to device sdk gphone64 x86 64...
E/OpenGLRenderer( 7171): Unable to match the desired swap behavior.
I/Choreographer( 7171): Skipped 100 frames!  The application may be doing too much work on its main thread.
D/ProfileInstaller( 7171): Installing profile for com.example.untitled3
D/TrafficStats( 7171): tagSocket(121) with statsTag=0xffffffff, statsUid=-1
D/CompatibilityChangeReporter( 7171): Compat change id reported: 3400644; UID 10183; state: DISABLED
```

```
onSubmit() async {
  if (_formKey.currentState?.validate() == true) {

    // can also check if user already exists or not
    final querySnapshot = await FirebaseFirestore.instance
      .collection('mycollection')
      .where('username', isEqualTo: usernameController.text)
      .get();
    if(querySnapshot.docs.isEmpty){
      await FirebaseFirestore.instance.collection('mycollection').add({
        'username': usernameController.text,
        'password': passwordController.text,
      });
      usernameController.text = '';
      passwordController.text = '';
      FocusScope.of(context).unfocus();
      Navigator.pushReplacement(context, MaterialPageRoute(builder: (context)=>Login(), settings: RouteSettings(
        arguments: {'msg': 'yay you can now login with provided credentials'}
      )));
      // RouteSettings, MaterialPageRoute
    }
    Navigator.pushReplacement(context, MaterialPageRoute(builder: (context)=>Login(), settings: RouteSettings(
      arguments: {'msg': 'seems like username already exists! so you may login now!'}
    )));
    // RouteSettings, MaterialPageRoute
  }
}
```

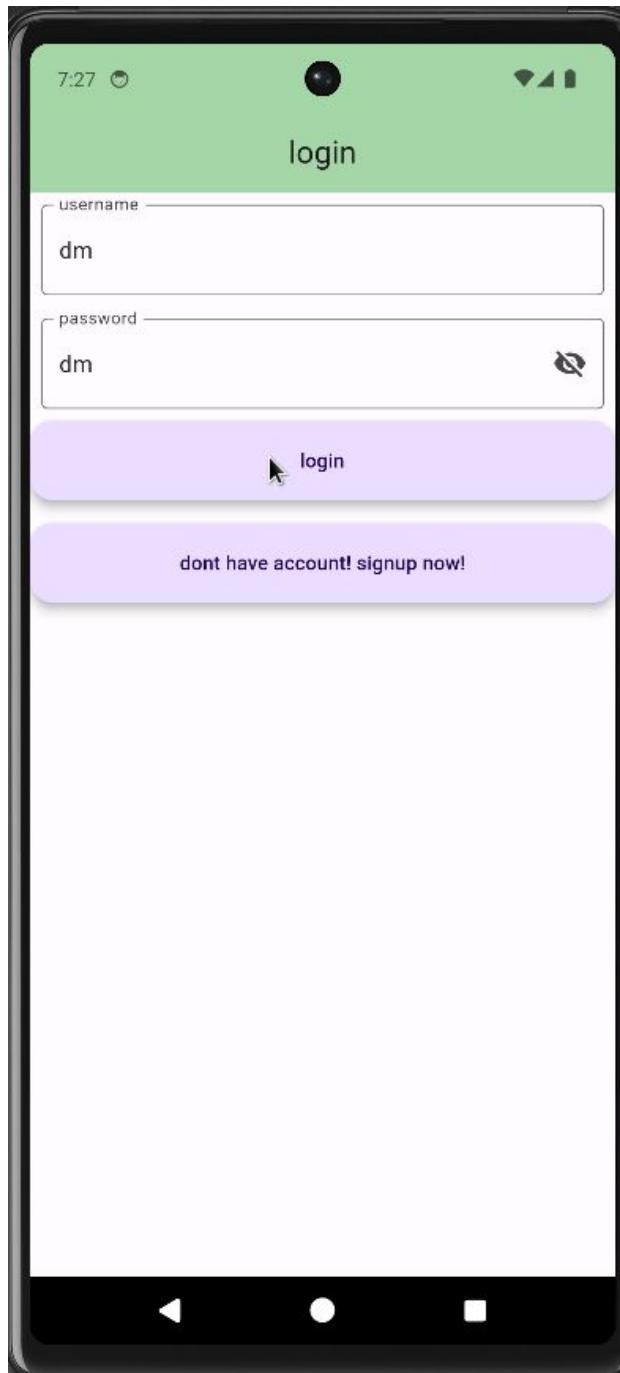
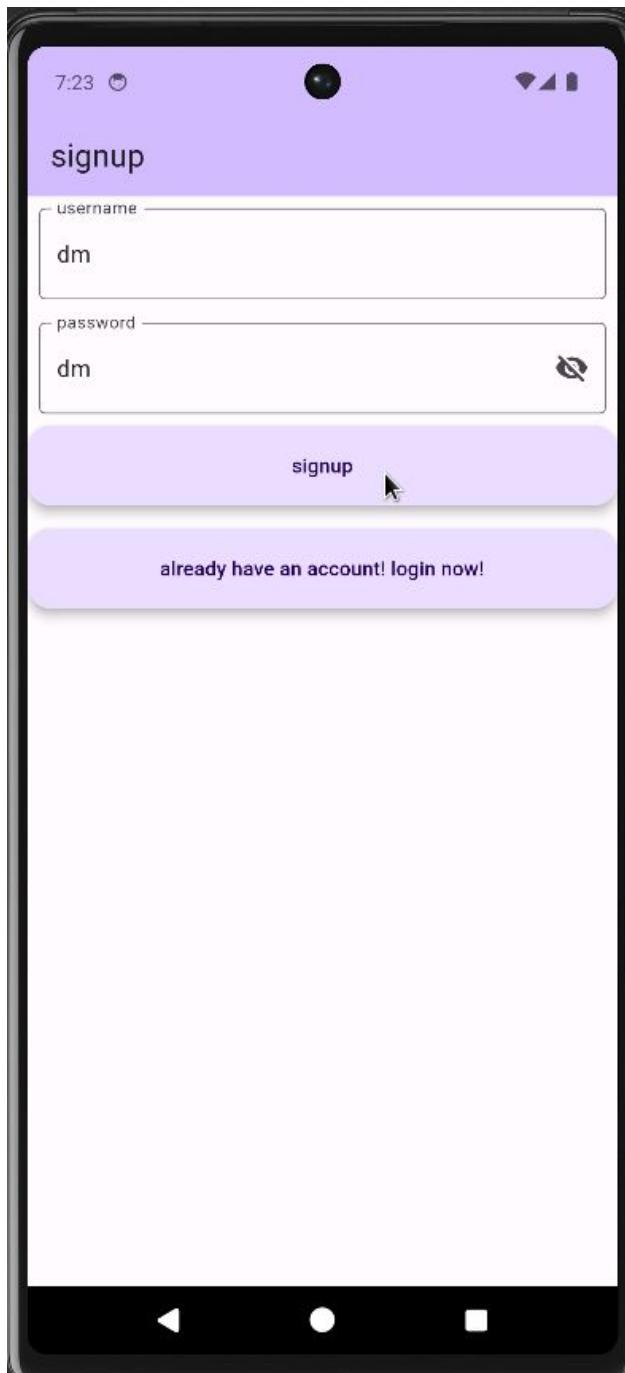
signup.dart

```
FloatingActionButton(  
    heroTag: 'signup',  
    onPressed: () async {  
        setState(() {  
            loading = !loading;  
            ScaffoldMessenger.of(context).showSnackBar(  
                const SnackBar(  
                    content: Row(  
                        children: [  
                            SpinKitWaveSpinner(color: Colors.indigo),  
                            Text('Processing Data'),  
                        ],  
                    ), // Row  
                ), // SnackBar  
            );  
        });  
        await onSubmit();  
        setState(() {  
            loading = !loading;  
            ScaffoldMessenger.of(context).hideCurrentSnackBar();  
            Navigator.of(context).pushNamed('/home');  
        });  
    },  
    child: const Text('signup'),  
), // FloatingActionButton
```

```
onSubmit() async {
  if (_formKey.currentState?.validate() == true) {
    final querySnapshot = await FirebaseFirestore.instance
      .collection('mycollection')
      .where('username', isEqualTo: usernameController.text)
      .get();
    if (querySnapshot.docs.isNotEmpty && querySnapshot.docs.length == 1) {
      if (querySnapshot.docs.first['password'] == passwordController.text) {
        Navigator.pushReplacement(
          context,
          MaterialPageRoute(
            builder: (context) => Home(),
            settings: RouteSettings(arguments: {
              'username': usernameController.text,
              'password': passwordController.text,
            }), // RouteSettings
          )); // MaterialPageRoute
      }
    }
    usernameController.text = '';
    passwordController.text = '';
    // _formKey.currentState?.reset();
    FocusScope.of(context).unfocus();
  }
}
```

login.dart

```
        ),
        child: Text('Login'),
        // disabledElevation: 10,
    ), // FloatingActionButton
```



(default)	mycollection	DnJJlYkFfujwW6efLWIZ
<p>+ Start collection</p> <p>mycollection ></p>	<p>+ Add document</p> <p>41b94nQhzNJ6xPIqctKf 7c6iKOvR8aIo5d9G0eN5 Cu6Nztqqp07YptJSacSG DnJJlYkFfujwW6efLWIZ > G2tjLIWpQ5vnxUSHjeBm GrlX4Vpjha3PA7QB4v8T LIC1MsJhSHGZff6hU6So RabHKWvLMu9FuLYU7SMg WIPXrl3cFoIjKz0WwaZn WOesM6p8R3JJ4Kw6r6tD pqfEjXz9YxidG0pYNyxF r24f4fsPfZRKsVJqJd3w sYMUJ0jf2dx6yYxqlFHs sj0yUwa50j9SHlQ5VrpS</p>	<p>+ Start collection</p> <p>+ Add field</p> <p>password: "dm" username: "dm"</p>

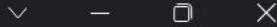
References

<https://youtu.be/Mx24wiPilHg>



Flutter

Connecting and Utilizing
MongoDB with Flutter

[←](#) [→](#) [C](#) [Home](#)

google.com/search?q=mongodb+atlas&rlz=1C1PNFE_enIN1026IN1026&oq=Mongodb+atlas&aqs=chrome.0.35i39i650j0i433i512j0i512l5j69i60.5156j0j7...



Google

mongodb atlas



B

[Download](#)[Login](#)[Videos](#)[Images](#)[Pricing](#)[Free](#)[Tutorial](#)[Vs MongoDB](#)[Create](#)[All filters](#)[Tools](#)[SafeSearch](#)

About 68,20,000 results (0.48 seconds)

**MongoDB**<https://www.mongodb.com> › atlas › database

MongoDB Atlas Database | Multi-Cloud Database Service

MongoDB Atlas is an integrated suite of data services centered around a cloud database designed to accelerate and simplify how you build with data. Build faster ...

[Charts](#) · [Search](#) · [Contact Us](#) · [Multi-Cloud Data Distribution](#)**MongoDB**<https://www.mongodb.com> › cloud › atlas › register

MongoDB Atlas

Simplify your data dependenciesLeverage application data for full-text search, real-time analytics, rich visualizations and more with a single API and minimal ...

People also ask :

What is MongoDB Atlas used for?



What is difference between MongoDB and MongoDB Atlas?



Do I need Atlas for MongoDB?



Search



1:21 PM

ENG US



25-Sep-23

Database Deployments | Cloud: MongoDB C X +

cloud.mongodb.com/v2/64c1e7426059024ed74aaa67#/clusters

G 🔍 ⭐ 🚧 🔍 B :

Atlas Bhargav's O... Access Manager Billing All Clusters Get Help Bhargav

Project 0 Data Services App Services Charts

DEPLOYMENT Database Data Lake

SERVICES Device Sync Triggers Data API Data Federation Search Stream Processing

SECURITY Backup Database Access Network Access Advanced

+ Add Tag

New On Atlas 2

BHARGAV'S ORG - 2023-07-27 > PROJECT 0

Database Deployments

Find a database deployment...

Edit Config Create

TourismManagement Connect View Monitoring Browse Collections ...

FREE SHARED

Monitoring for TourismManagement is Paused
Monitoring will automatically resume when you connect to your cluster.
[Visit the documentation](#) for more info.

VERSION	REGION	CLUSTER TIER	TYPE	BACKUPS	LINKED APP SERVICES	ATLAS SQL	ATLAS SEARCH
6.0.10	AWS / N. Virginia (us-east-1)	M0 Sandbox (General)	Replica Set - 3 nodes	Inactive	None Linked	Connect	Create Index

+ Add Tag

1:23 PM ENG US 25-Sep-23

A small red box highlights the "TourismManagement" cluster entry, and a red arrow points from the center of the box up towards the "Connect" button.

Database Deployments | Cloud: MongoDB X +

cloud.mongodb.com/v2/64c1e7426059024ed74aaa67#/clusters/connect?clusterId=TourismManagement

G 🔍 ⭐ 🚧 ↴ B :

Atlas Bhargav's O... Project 0 Data S DEPLOYMENT Database Data Lake SERVICES Device Sync Triggers Data API Data Federation Search Stream Processing SECURITY Backup Database Access Network Access Advanced New On Atlas 2

All Clusters Get Help Bhargav

BHARGAV'S OR Database Edit Config + Create FREE SHARED ATLAS SQL ATLAS SEARCH Connect Create Index

Connect to TourismManagement

1 Set up connection security 2 Choose a connection method 3 Connect

Connect to your application

Drivers
Access your Atlas data using MongoDB's native drivers (e.g. Node.js, Go, etc.)

Access your data through tools

Compass
Explore, modify, and visualize your data with MongoDB's GUI

Shell
Quickly add & update data using MongoDB's Javascript command-line interface

MongoDB for VS Code
Work with your data in MongoDB directly from your VS Code environment

1:25 PM 25-Sep-23



Bhargav's O...

Access Man...

Project 0

Data Services

BHARGAV'S ORG - 2023-07-2

Database D

Find a database deplo

DEPLOYMENT

Database

Data Lake

SERVICES

Device Sync

Triggers

Data API

Data Federation

Search

Stream Processing

SECURITY

Backup

Database Access

VERSION

REGION

6.0.10

AWS / N.

+ Add Tag

New On Atlas 2

Goto

Connect to TourismManagement



Set up connection security



Choose a connection method



Connect

Connecting with MongoDB Driver

1. Select your driver and version

We recommend installing and using the latest driver version.

Driver

Version

Node.js

5.5 or later

2. Install your driver

Run the following on the command line

```
npm install mongodb
```

[View MongoDB Node.js Driver installation instructions.](#)

3. Add your connection string into your application code

 View full code sample

```
mongodb+srv://<username>:<password>@tourismmanagement.vapyc9c.mongodb.net/?  
retryWrites=true&w=majority
```



Replace **<password>** with the password for the **<username>** user. Ensure any option params are [URL encoded](#).



Search

1:26 PM
ENG US
25-Sep-23

main.dart

pubspec.yaml ×



```
signup_signin_mongodb > pubspec.yaml
  ↵  |  SUK.  ↵=0.1.0 ^4.0.0
23
24  # Dependencies specify other packages that your package needs in order to work.
25  # To automatically upgrade your package dependencies to the latest versions
26  # consider running `flutter pub upgrade --major-versions`. Alternatively,
27  # dependencies can be manually updated by changing the version numbers below to
28  # the latest version available on pub.dev. To see which dependencies have newer
29  # versions available, run `flutter pub outdated`.
30  dependencies:
31    mongo_dart: ^0.9.1
32    flutter:
33      |  sdk: flutter
34
35
36  # The following adds the Cupertino Icons font to your application.
37  # Use with the CupertinoIcons class for iOS style icons.
38  cupertino_icons: ^1.0.2
39
40
41  dev_dependencies:
42    flutter_test:
43      |  sdk: flutter
44
45  # The "flutter_lints" package below contains a set of recommended lints to
46  # encourage good coding practices. The lint set provided by the package is
47  # activated in the `analysis_options.yaml` file located at the root of your
48  # package. See that file for information about deactivating specific lint
49  # rules and activating additional ones.
```

MongoConnect.dart

main.dart 2

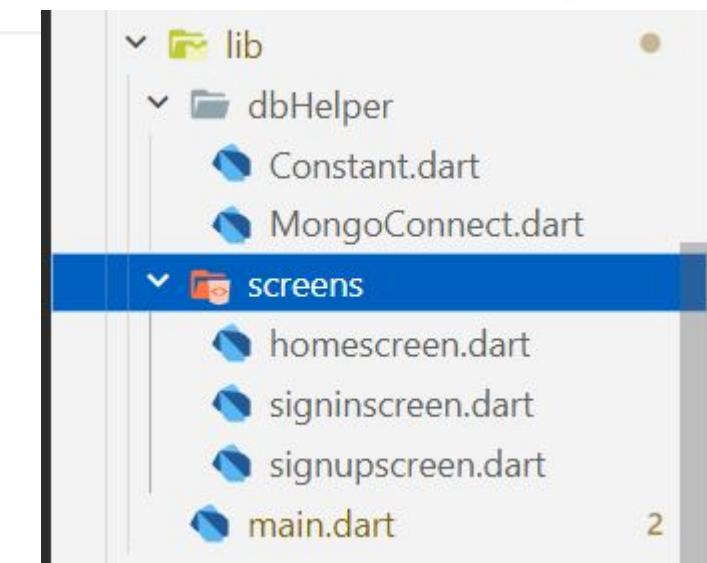
▶ ▾ ⌂ ⋮

signup_signin_mongodb > lib > main.dart > ...

```
1 import 'package:flutter/material.dart';
2 import 'package:mongo_dart/mongo_dart.dart';
3 import 'package:signup_signin_mongodb/dbHelper/MongoConnect.dart';
4 import 'package:signup_signin_mongodb/screens/signupscreen.dart';
5 import 'screens/signinscreen.dart';
6
7
8 void main() async {
9     WidgetsFlutterBinding.ensureInitialized();
10    MongoDBService ms = MongoDBService();
11    await ms.connect();
12    runApp(const MyApp());
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             home: SignUpScreen(),
23         ); // MaterialApp
24     }
25 }
```

signup_signin_mongodb > lib > dbHelper > MongoConnect.dart > MongoDBService > connect

```
1 import 'package:mongo_dart/mongo_dart.dart';
2 import 'Constant.dart';
3
4 class MongoDBService {
5     static late var _db;
6
7     Future<void> connect() async {
8         final String dbUrl = db_url;
9         _db = await Db.create(dbUrl);
10        await _db.open();
11        print('Connected to MongoDB');
12    }
13
14    Future<void> close() async {
15        await _db.close();
16        print('Connection to MongoDB closed');
17    }
18
19    Future<void> registerUser(Map<String, dynamic> user) async {
20        final usersCollection = _db.collection('users');
21        await usersCollection.insert(user);
22        print("Register User Called");
23    }
24
25
26    Future<Map<String, dynamic>?> findUserByEmail(String email) async {
27        final usersCollection = _db.collection('users');
28        final user = await usersCollection.findOne(where.eq('email', email));
29        return user;
30    }
31 }
```



Constant.dart

The screenshot shows a code editor interface with two tabs: "MongoConnect.dart" and "Constant.dart". The "Constant.dart" tab is active, indicated by a blue icon and the word "Constant.dart" followed by an "X". The file path "signup_signin_mongodb > lib > dbHelper > Constant.dart > ..." is displayed above the code area. The code itself contains two lines:

```
1 const db_url = "mongodb+srv://new_user:LB0ehdvltg1yb7QC@tourismmanagement.vapyc9c.mongodb.net/?r  
2 const collection_name = "users";|
```

The "db_url" variable is underlined with a yellow squiggle, indicating a potential error or warning. The "collection_name" variable is underlined with a blue squiggle.

MongoConnect.dart

signupscreen.dart X

...
E

signup_signin_mongodb > lib > screens > signupscreen.dart > _SignUpScreenState > build

```
83     ), // AppBar
84     body: Center(
85       child: Padding(
86         padding: const EdgeInsets.all(20.0),
87         child: Column(
88           mainAxisAlignment: MainAxisAlignment.center,
89           children: <Widget>[
90             TextFormField(
91               controller: emailController,
92               decoration: InputDecoration(
93                 labelText: 'Email Address',
94                 border: OutlineInputBorder(),
95                 errorText: isEmpty ? 'Please enter an email' : null,
96               ), // InputDecoration
97             ), // TextFormField
98             SizedBox(height: 20.0),
99             TextFormField(
100               controller: passwordController,
101               decoration: InputDecoration(
102                 labelText: 'Password',
103                 border: OutlineInputBorder(),
104                 errorText: isPasswordEmpty ? 'Please enter a password' : null,
105               ), // InputDecoration
106               obscureText: true,
107             ), // TextFormField
108             SizedBox(height: 20.0),
109             ElevatedButton(
110               onPressed: isLoading
111                 ? null
112                 : () {
113                   createAccount(context);
114                 },
115               child: Padding(
116                 padding: const EdgeInsets.symmetric(
```

```
30 void createAccount(BuildContext context) async {
31   String email = emailController.text.trim();
32   String password = passwordController.text.trim();
33
34   setState(() {
35     isEmpty = false;
36     isPasswordEmpty = false;
37   });
38
39   if (email.isEmpty) {
40     setState(() {
41       isEmpty = true;
42     });
43     showSnackbar('Please enter an email');
44     return;
45   }
46   if (password.isEmpty) {
47     setState(() {
48       isPasswordEmpty = true;
49     });
45   }
46   showSnackbar('Please enter a password');
47   return;
48 }
49
50 setState(() {
51   isLoading = true; // Set loading state to true
52 });
53
54 try {
55   Map<String, dynamic> user = {"email": email, "password": password};
56   MongoDBService ms = MongoDBService();
57   // await ms.connect();
58   await ms.registerUser(user);
59   Navigator.pushReplacement(
60     context,
61     MaterialPageRoute(builder: (context) => homepage()),
62   );
63   showSnackbar('Registration Successful');
64 } catch (err) {
65   showSnackbar(err.toString());
66 } finally {
67 }
```

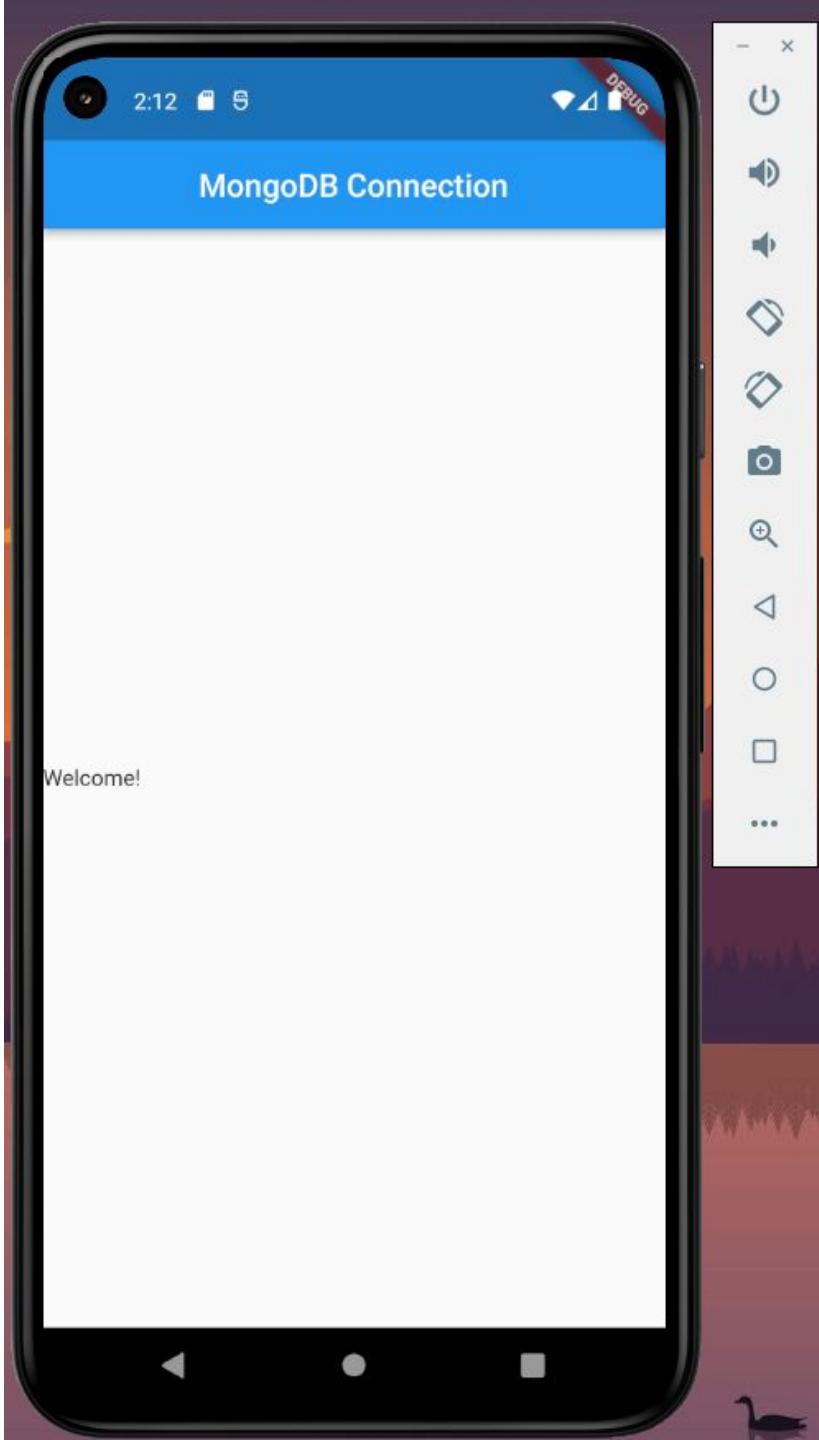
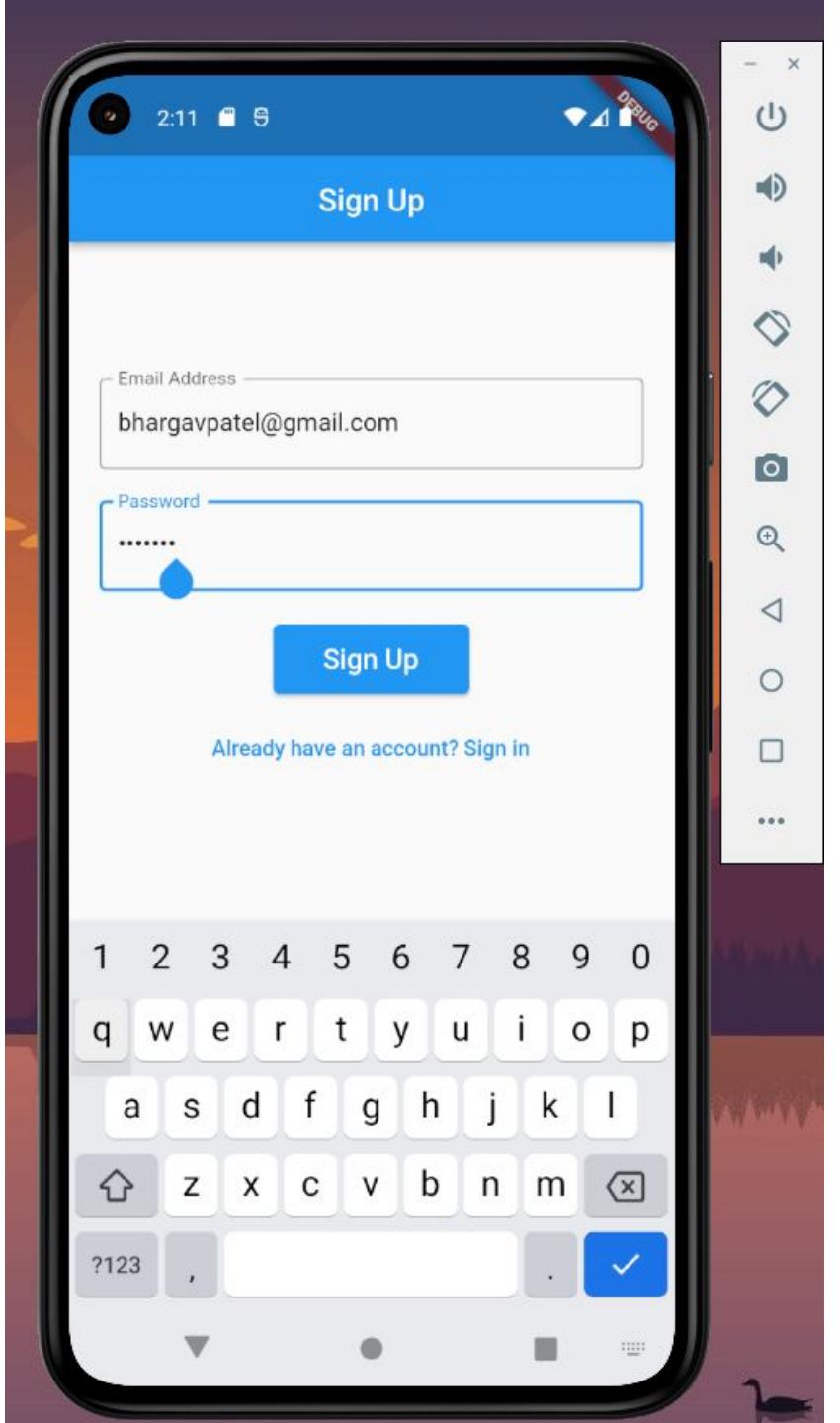
```
94         decoration: InputDecoration(  
95             labelText: 'Email Address',  
96             errorText: isEmpty ? 'Please enter an email' : null,  
97         ), // InputDecoration  
98     ), // TextField  
99     SizedBox(height: 20.0),  
100  
101     TextField(  
102         controller: passwordController,  
103         decoration: InputDecoration(  
104             labelText: 'Password',  
105             errorText: isPasswordEmpty ? 'Please Enter Password' : null,  
106         ), // InputDecoration  
107         obscureText: true,  
108     ), // TextField  
109     SizedBox(height: 20.0),  
110     // sign in Button  
111     ElevatedButton(  
112         onPressed: isLoading  
113             ? null  
114             : () {  
115                 SignIn(context);  
116             },  
117             child: isLoading  
118                 ? CircularProgressIndicator()  
119                 : Text('Sign In'),  
120     ), // ElevatedButton  
121     CupertinoButton(  
122         child: Text('Create an Account'),  
123         onPressed: () {  
124             Navigator.push(  
125                 context,  
126                 CupertinoPageRoute(  
127                     builder: (context) => SignUpScreen())); // CupertinoPageRoute
```

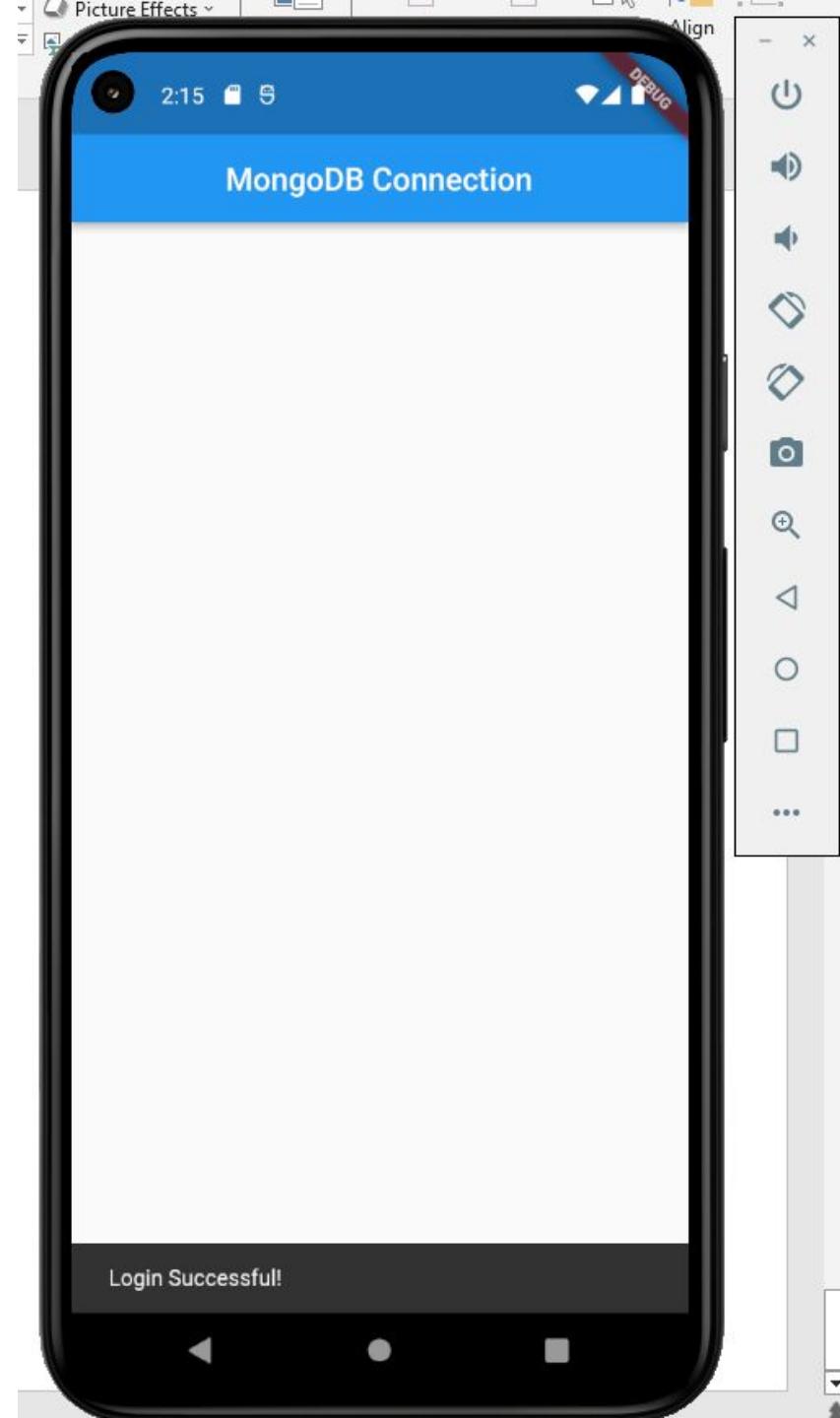
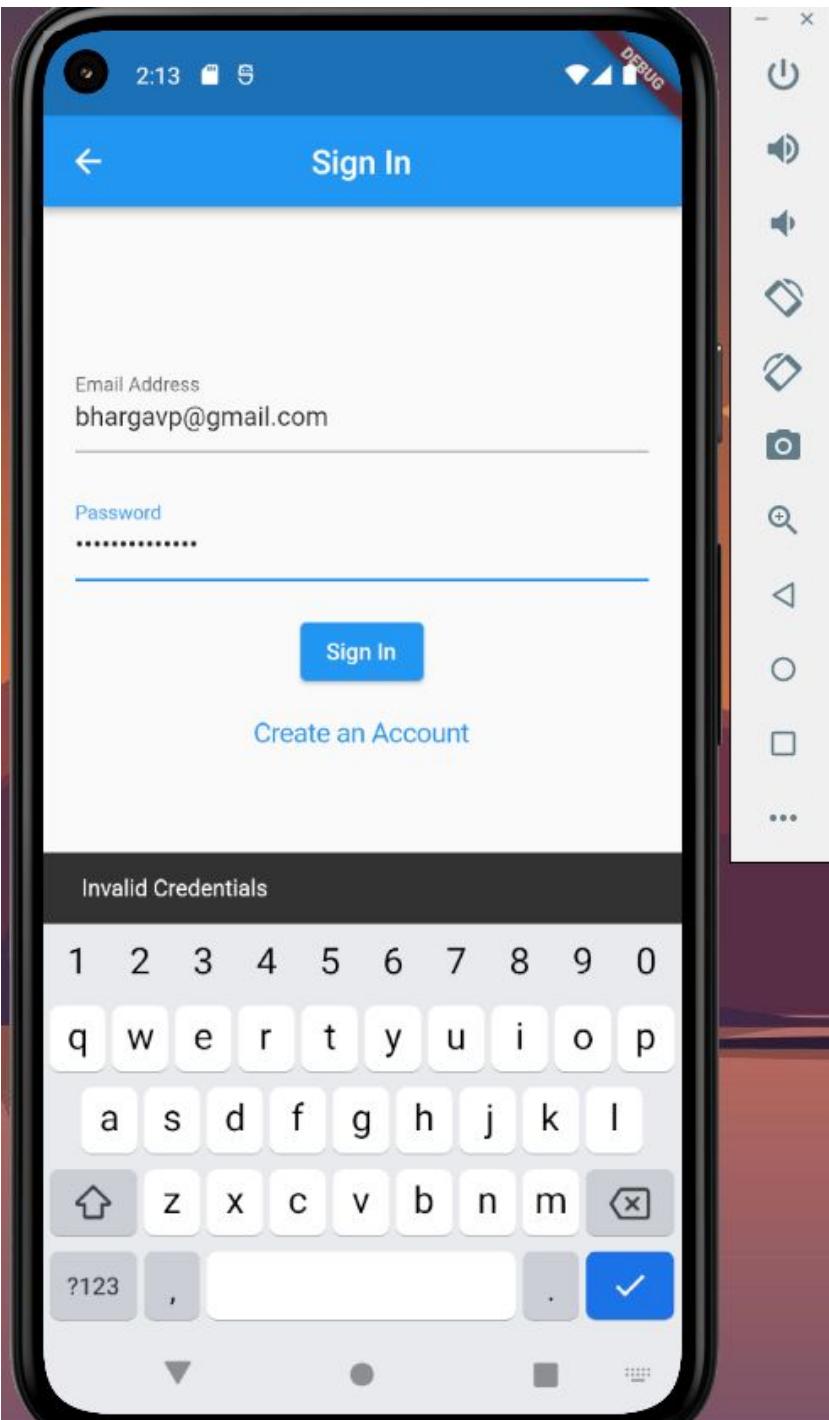
```
void SignIn(context) async {
  String email = emailController.text;
  String password = passwordController.text;

  setState(() {
    isEmptyEmail = false;
    isEmptyPassword = false;
  });

  if (email == "") {
    setState(() {
      isEmptyEmail = true;
    });
    showSnackbar('Please enter an email');
  }
  if (password == "") {
    setState(() {
      isEmptyPassword = true;
    });
    showSnackbar('Please enter password');
  } else {
    try {
      MongoDBService ms = MongoDBService();
      Map<String, dynamic>? user = await ms.findUserByEmail(email);
    
```

```
      if(user != null) {
        if(user["password"] == password)
        {
          Navigator.popUntil(context, (route) => route.isFirst);
          Navigator.pushReplacement(
            context, CupertinoPageRoute(builder: (context) => homepage()));
          showSnackbar('Login Successful!');
        }
        else {
          showSnackbar("Invalid Credentials");
        }
      }
      else
      {
        showSnackbar("Invalid Credentials");
      }
    
```





test.users

STORAGE SIZE: 36KB LOGICAL DATA SIZE: 298B TOTAL DOCUMENTS: 4 INDEXES TOTAL SIZE: 36KB

[Find](#)[Indexes](#)[Schema Anti-Patterns](#)[Aggregation](#)[Search Indexes](#)[INSERT DOCUMENT](#)[Filter](#)

Type a query: { field: 'value' }

[Reset](#)[Apply](#)[More Options](#)

QUERY RESULTS: 1-3 OF 3

```
_id: ObjectId('6504710ab460ab26abcaf1b9')
email: "devmehta@dmail.com"
password: "devmwhta"
```

```
_id: ObjectId('650494f537f1760d0828ffbe')
email: "shreya@gmail.com"
password: "shreya"
```

```
_id: ObjectId('651147dd68ca1f539d78a41d')
email: "bhargavpatel@gmail.com"
password: "bhargav"
```





Flutter

Accessing Hardware Components Using Flutter (accessing microphone)

main.dart

pubspec.yaml X

Download Refresh Sync Minimize Maximize

```
pubspec.yaml
22  SUK: >=2.1.0 <4.0.0
23
24 # Dependencies specify other packages that your package needs in order to work.
25 # To automatically upgrade your package dependencies to the latest versions
26 # consider running `flutter pub upgrade --major-versions`. Alternatively,
27 # dependencies can be manually updated by changing the version numbers below to
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 CupertinoIcons font to your application.
36 # Use with the CupertinoIcons class for iOS style icons.
37 cupertino_icons: ^1.0.2
38 avatar_glow: ^2.0.2
39 speech_to_text: ^6.3.0
40
41 dev_dependencies:
42   flutter_test:
43     | sdk: flutter
44
45 # The "flutter_lints" package below contains a set of recommended lints to
46 # encourage good coding practices. The lint set provided by the package is
47 # activated in the `analysis_options.yaml` file located at the root of your
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.
```

EXPLORER ... main.dart pubspec.yaml SpeechScreen.dart AndroidManifest.xml

MICROPHONE... + C F

android > app > src > main > AndroidManifest.xml

```
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|fontScale|size|density|dpi|uiMode|densitySwapped"
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

The screenshot shows a code editor window with the file `main.dart` open. The file path is `lib > main.dart > MyApp > build`. The code implements a simple Flutter application structure:

```
1 import 'package:flutter/material.dart';
2 import '../screens/SpeechScreen.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       home: SpeechScreen(),
16     ); // MaterialApp
17   }
18 }
19
```

The code uses Dart syntax, including imports for `flutter/material.dart` and `SpeechScreen.dart`. It defines a `void main()` function to run the application and a `MyApp` class that extends `StatelessWidget`. The `build` method returns a `MaterialApp` widget with `SpeechScreen` as the home screen.

speech_to_text | Flutter Package

pub.dev/packages/speech_to_text

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: errorListener );
if ( available ) {
    speech.listen( onResult: resultListener );
}
else {
    print("The user has denied the use of speech recognition.");
}
// some time later...
speech.stop()
```

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

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
    Widget build(BuildContext context) {
```

lib > screens >  SpeechScreen.dart >  _SpeechScreenState >  speechToText

```
1 import 'package:avatar_glow/avatar_glow.dart';
2 import 'package:flutter/material.dart';
3 import 'package:speech_to_text/speech_to_text.dart';
4
5 class SpeechScreen extends StatefulWidget {
6     const SpeechScreen({super.key});
7
8     @override
9     State<SpeechScreen> createState() => _SpeechScreenState();
10 }
11
12 class _SpeechScreenState extends State<SpeechScreen> {
13     SpeechToText speechToText = SpeechToText();
14     var text = "Hold the button and start speaking";
15     var isListening = false;
16
17     @override
18     Widget build(BuildContext context) {
19         return Scaffold(
20             floatingActionButtonLocation: FloatingActionButtonLocation.centerFloat,
21             floatingActionButton: AvatarGlow(
22                 endRadius: 75.0,
23                 animate : isListening,
24                 duration:Duration(milliseconds:2000),
25                 glowColor : Colors.red,
26                 repeat : true,
```

main.dart

pubspec.yaml

SpeechScreen.dart

...
E

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

```
55     |         isListening = false;
56     |     );
57     |     speechToText.stop();
58     },
59
60     child: CircleAvatar(
61         backgroundColor: Colors.blue.shade800,
62         radius : 35,
63         child : Icon(isListening ? Icons.mic : Icons.mic_none, color : Colors.white),
64
65     ), // CircleAvatar
66     ), // GestureDetector
67     ), // AvatarGlow
68     appBar : AppBar(
69         backgroundColor: Colors.blue.shade800,
70         leading : const Icon(Icons.sort_rounded, color : Colors.white),
71         title : Text("Speech to Text"),
72         centerTitle: true,
73     ), // AppBar
74     body : Container(
75
76         padding : const EdgeInsets.symmetric(horizontal: 24, vertical : 16),
77         margin: const EdgeInsets.only(bottom:150.0),
78         child : Text(
79             text,
80             style : const TextStyle(fontSize: 24, color : Colors.black),
81         ) // Text
82     ) // Container
83 ); // Scaffold
84 }
85 }
86 }
```

7:19

0.1
LTE KB/s 4G+ .11 70%

7:20

0
LTE KB/s 4G+ .11 70%

7:20

2
LTE KB/s 4G+ .11 70%

7:21

0.36
LTE KB/s 4G+ .11 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



Speech to Text

Hold the button and start speaking



Allow microphone_access to find, connect to, and determine the relative position of nearby devices?

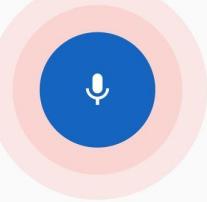
ALLOW

DON' T ALLOW



Speech to Text

Hold the button and start speaking



Speech to Text

this is speech to text Application in which whatever you are speaking will be converted to text



Thank You



Flutter

Accessing Hardware Components Using Flutter (accessing camera)

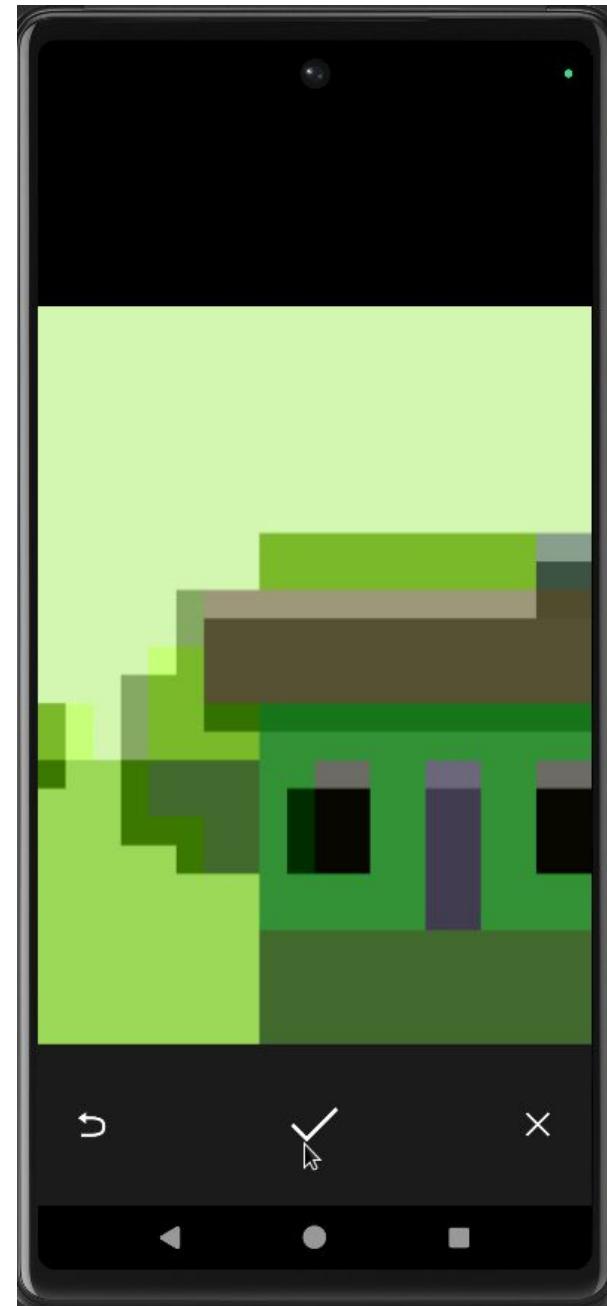
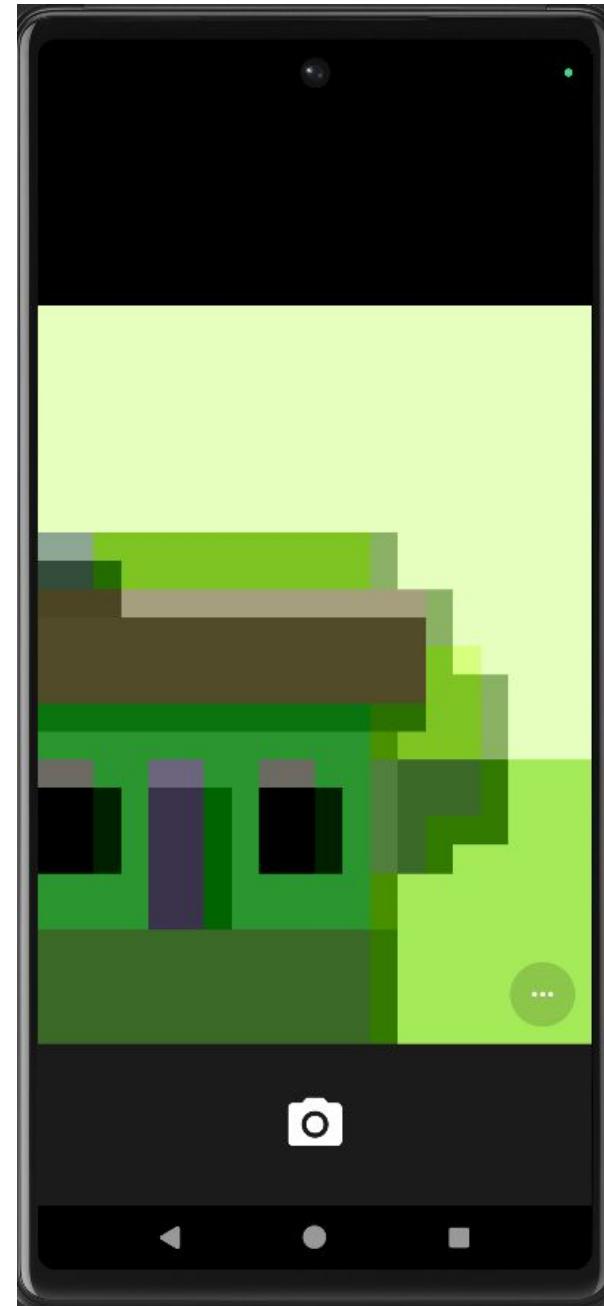
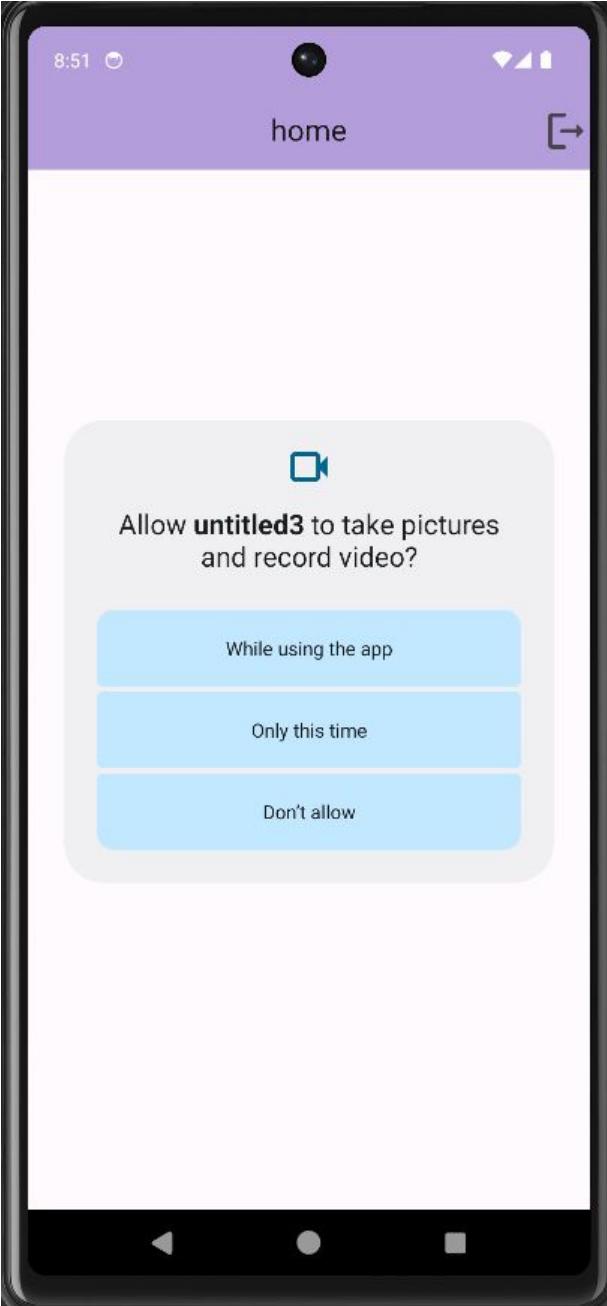
```
dependencies:  
  flutter:  
    sdk: flutter  
  firebase_core:  
  cloud_firestore:  
  firebase_auth:  
  google_sign_in:  
  firebase_crashlytics:  
  firebase_analytics:  
  firebase_performance:  
  flutter_spinkit:  
  google_nav_bar:  
  animated_notch_bottom_bar:  
  lottie:  
  image_picker: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

```
<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|smallestScreenSize|locale|layoutDirection|fontScale|density|ActionBar|color"
            android:hardwareAccelerated="true"
            android:windowSoftInputMode="adjustResize">
```

app/src/main/AndroidManifest.xml

```
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,
    ),
  ],
),
),
),
); // Scaffold
}
}
```



Thank You