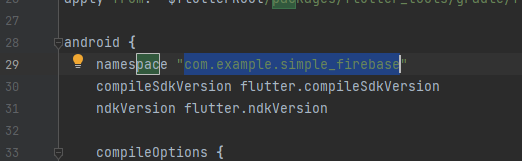
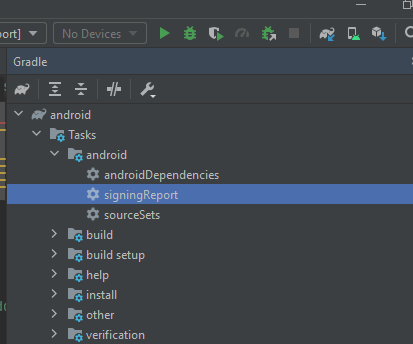
* Firstly, create a simple flutter project.
* Now go to google and search for firebase console.
* Click create project.
* Enter project name. Maximum projects of which you can create is 12.
* Accept terms and click continue.
* Go with the flow until your project is created.
* Now click on android.
* Now we have to register our project in firebase of we have created on step 1 above.
* To do that, open gradle.build file and copy this and put it in the place of android package name. 
* Open AndriodManifest.xml file from Android>app>src>main
* Click on “open for Editing in android studio” located at the top right corner in new window.
* In that new window click on gradle located at the right side bar. In that click on android>Tasks>android>signing report.



* Copy SHA1 key from there and paste it in the firebase project registration page and click on register.
* Now download the google service.jason file and paste it in the app. Then click on next. Make sure file name does not change. After this click next.
* Now copy lines and paste them in the perspective locations.
* Before copying lines of java or kotlin check that google play service is installed or not. To do that click on sdk manger , from there go to system settings>Android SDk and click on SDK tools. Then check that, google play services and google play licensing library, is checked or not if not then check them and click apply.
* You are ready now.

If you like you can see documentation for firebase from here: [Fundamentals  |  Firebase Documentation (google.com)](https://firebase.google.com/docs/guides?hl=en&authuser=0&_gl=1*q0ouk4*_ga*MTkxMzM3MTIzMi4xNjg3NTI5Njcw*_ga_CW55HF8NVT*MTY4NzY2NzQ1NS4zLjAuMTY4NzY2NzQ1NS4wLjAuMA..) or simple examples from here: [Firebase Documentation (google.com)](https://firebase.google.com/docs/samples?hl=en&authuser=0&_gl=1*1j0tdi9*_ga*MTkxMzM3MTIzMi4xNjg3NTI5Njcw*_ga_CW55HF8NVT*MTY4NzY2NzQ1NS4zLjEuMTY4NzY2NzU0MC4wLjAuMA..)

* At the end click on the return to console to back to console. Now you can see your app in the console.
* Click on your app and click on settings gear button.
* Click on build>realtime database from the menu located at the left.
* Click create database.
* Select respective option and click next after this select start in lock mode and click enable.
* Now click on rules.
* Remove false and type true in front of read and write and ignore the security message for now.
* Our work on firebase for now is complete, so let us go back to our android studio.
* Now run your app and check that no error occurs.
* Now go to the terminal to check flutter version. It should be stable.
* Write flutter –version to check flutter version. There should be no error.
* Now we have to put plugins for firebase. For that go to pubspec.yaml file.
* Normally at first time google-play.services file doesn’t build right so download it again and add in respective location.

Write these three lines in dependencies,

firebase\_auth: any  
firebase\_core: any  
firebase\_database: any // click pub get

**Writing data using app:**

https://firebase.google.com/docs/database/flutter/read-and-write

https://www.youtube.com/watch?v=8kUx9TFhXpQ

* Add packages: firebase database and firebase core.

Code:

import 'package:flutter/material.dart';  
import 'package:firebase\_database/firebase\_database.dart';  
import 'package:firebase\_core/firebase\_core.dart';  
import 'package:firebase\_auth/firebase\_auth.dart';  
  
void main() async {  
 WidgetsFlutterBinding.*ensureInitialized*();  
 await Firebase.*initializeApp*();  
 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: 'Flutter Demo',  
 home: const MyHomePage(title: 'Flutter Demo Home Page'),  
 );  
 }  
}

class MyHomePage extends StatefulWidget {  
 const MyHomePage({super.key, required this.title});  
 final String title;  
  
 @override  
 State<MyHomePage> createState() => \_MyHomePageState();  
}  
  
class \_MyHomePageState extends State<MyHomePage> {  
 int \_counter = 0;  
 late DatabaseReference dbRef;  
  
 void \_incrementCounter() {  
 setState(() {  
 \_counter++;  
 });  
 }  
  
 @override  
 void initState() {  
 super.initState();  
 setState(() {  
 dbRef = FirebaseDatabase.*instance*.ref().child("Detail");  
 });  
 }  
  
 @override  
 Widget build(BuildContext context) {  
 return Scaffold(  
 appBar: AppBar(

backgroundColor: Colors.*blue*,  
 title: Text(widget.title),  
 ),  
 body: Center(  
 child: Column(  
 mainAxisAlignment: MainAxisAlignment.center,  
 children: <Widget>[  
 const Text(  
 'You have pushed the button this many times:',  
 ),  
 Text(  
 '$\_counter',  
 style: Theme.*of*(context).textTheme.headlineMedium,  
 ),  
 ],  
 ),  
 ),  
 floatingActionButton: FloatingActionButton(  
 onPressed: () {  
 \_incrementCounter();  
 Map<String, String> Details = {"Number of counts": "$\_counter"};  
 dbRef.set(  
 Details); // we can also use dbRef.push().set(Detsils);// this will generate unique key on every entry.  
 },  
 tooltip: 'Increment',  
 child: const Icon(Icons.*add*),  
 ),   
 );  
 }  
}

**Reading data using App:**

<https://www.youtube.com/watch?v=y_CE0jm3Lb4>

import 'package:firebase\_database/ui/firebase\_animated\_list.dart';  
import 'package:flutter/material.dart';  
import 'package:firebase\_database/firebase\_database.dart';  
import 'package:firebase\_core/firebase\_core.dart';  
import 'dart:convert';  
import 'package:firebase\_auth/firebase\_auth.dart';  
  
void main() async {  
 WidgetsFlutterBinding.*ensureInitialized*();  
 await Firebase.*initializeApp*();  
 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: 'Flutter Demo',  
 home: const MyHomePage(title: 'Flutter Demo Home Page'),  
 );  
 }  
}  
  
class MyHomePage extends StatefulWidget {  
 const MyHomePage({super.key, required this.title});  
  
 final String title;  
  
 @override  
 State<MyHomePage> createState() => \_MyHomePageState();  
}  
  
class \_MyHomePageState extends State<MyHomePage> {  
 int \_counter = 0;  
 late DatabaseReference dbRef;  
  
 Widget listitem({required Map student}) {  
 return Container(  
 child: Column(  
 children: [  
 Text(  
 'No. of counts on firebase:${student['Number of counts']}',  
 )  
 ],  
 ),  
 );  
 }  
  
 void \_incrementCounter() {  
 setState(() {  
 \_counter++;  
 });  
 }  
  
 @override  
 void initState() {  
 super.initState();  
 setState(() {  
 dbRef = FirebaseDatabase.*instance*.ref().child("Detail");  
 });  
 }  
  
 @override  
 Widget build(BuildContext context) {  
 return Scaffold(  
 appBar: AppBar(  
 backgroundColor: Colors.*blue*,  
 title: Text(widget.title),  
 ),  
 body: Center(  
 child: Column(  
 mainAxisAlignment: MainAxisAlignment.center,  
 children: <Widget>[  
 Text(  
 'You have pushed the button this many times:',  
 ),  
 Text(  
 '$\_counter',  
 style: Theme.*of*(context).textTheme.headlineMedium,  
 ),  
 Container(  
 height: 300,  
 child: Expanded(  
 child: FirebaseAnimatedList(  
 query: dbRef,  
 itemBuilder: (BuildContext context, DataSnapshot snapshot,  
 Animation<double> animation, int index) {  
 return ListTile(  
 title: Center(  
 child: Text(  
 'No. of counts on firebase:${snapshot.child('Number of counts').value.toString()}'),  
 ),  
 );  
 }),  
 ),  
 )  
 ],  
 ),  
 ),  
 floatingActionButton: FloatingActionButton(  
 onPressed: () {  
 \_incrementCounter();  
 Map<String, String> Details = {"Number of counts": "$\_counter"};  
 dbRef.push().set(  
 Details); // we can also use dbRef.push().set(Detsils);// this will generate unique key on every entry.  
 },  
 tooltip: 'Increment',  
 child: const Icon(Icons.*add*),  
 ), // This trailing comma makes auto-formatting nicer for build methods.  
 );  
 }  
}

Also see this to stream data from firebase: <https://www.youtube.com/watch?v=C4XwkQcNCQs>

**Editing and deleting data:**

import 'package:firebase\_database/ui/firebase\_animated\_list.dart';  
import 'package:flutter/material.dart';  
import 'package:firebase\_database/firebase\_database.dart';  
import 'package:firebase\_core/firebase\_core.dart';  
import 'dart:convert';  
import 'package:firebase\_auth/firebase\_auth.dart';  
  
void main() async {  
 WidgetsFlutterBinding.*ensureInitialized*();  
 await Firebase.*initializeApp*();  
 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: 'Flutter Demo',  
 home: const MyHomePage(title: 'Flutter Demo Home Page'),  
 );  
 }  
}  
  
class MyHomePage extends StatefulWidget {  
 const MyHomePage({super.key, required this.title});  
  
 final String title;  
  
 @override  
 State<MyHomePage> createState() => \_MyHomePageState();  
}  
  
class \_MyHomePageState extends State<MyHomePage> {  
 var \_counter = TextEditingController();  
 var editcontroller = TextEditingController();  
 late DatabaseReference dbRef;  
  
 Widget listitem({required Map student}) {  
 return Container(  
 child: Column(  
 children: [  
 Text(  
 'No. of counts on firebase:${student['Number of counts']}',  
 )  
 ],  
 ),  
 );  
 }  
  
 // void \_incrementCounter() {  
 // setState(() {  
 // \_counter++;  
 // });  
 // }  
  
 @override  
 void initState() {  
 super.initState();  
 setState(() {  
 dbRef = FirebaseDatabase.*instance*.ref().child("Detail");  
 });  
 }  
  
 @override  
 Widget build(BuildContext context) {  
 return Scaffold(  
 appBar: AppBar(  
 backgroundColor: Colors.*blue*,  
 title: Text(widget.title),  
 ),  
 body: Center(  
 child: Column(  
 mainAxisAlignment: MainAxisAlignment.center,  
 children: <Widget>[  
 Container(  
 //flex: 3,  
 child: TextField(  
 controller: \_counter,  
 //keyboardType: TextInputType.text,  
 decoration: InputDecoration(  
 hintText: 'write here',  
 ),  
 ),  
 ),  
 Container(  
 //flex: 1,  
 child: TextButton(  
 style: TextButton.*styleFrom*(  
 foregroundColor: Colors.*white*,  
 backgroundColor: Colors.*blue*, // Background Color  
 ),  
 onPressed: () {  
 // Map<String, String> Details = {  
 // "title": "${\_counter.text}"  
 // "id": "${DateTime.now().}"  
 // };  
 if (\_counter.text == '') {//do nothing.  
 } else {  
 String id =  
 DateTime.now().microsecondsSinceEpoch.toString();  
 dbRef.child(id).set(  
 {"title": "${\_counter.text.toString()}", "id": "$id"});  
 // dbRef.push().set(  
 // Details); // we can also use dbRef.push().set(Details);// this will generate unique key on every entry.  
 \_counter.text = '';  
 }  
 },  
 child: Text('Save Data on firebase'),  
 ),  
 ),  
 Expanded(  
 child: Container(  
 height: 300,  
 child: Expanded(  
 child: FirebaseAnimatedList(  
 query: dbRef,  
 itemBuilder: (BuildContext context, DataSnapshot snapshot,  
 Animation<double> animation, int index) {  
 final title = snapshot.child('title').value.toString();  
 return ListTile(  
 title: Center(  
 child: Text(  
 'Title: ${snapshot.child('title').value.toString()}'),  
 ),  
 subtitle: Center(  
 child: Text(  
 'Id: ${snapshot.child('id').value.toString()}'),  
 ),  
 trailing: PopupMenuButton(  
 icon: Icon(Icons.*more\_vert*),  
 itemBuilder: (context) => [  
 PopupMenuItem(  
 value: 1,  
 child: ListTile(  
 onTap: () {  
 Navigator.*pop*(context);  
 showMyDialog(  
 title,  
 snapshot  
 .child('id')  
 .value  
 .toString());  
 },  
 leading: Icon(Icons.*edit*),  
 title: Text('Edit'),  
 )),  
 PopupMenuItem(  
 value: 1,  
 child: ListTile(  
 onTap: () {  
 Navigator.*pop*(context);  
 dbRef  
 .child(snapshot  
 .child('id')  
 .value  
 .toString())  
 .remove();  
 },  
 leading: Icon(Icons.*delete*),  
 title: Text('Delete'),  
 )),  
 ],  
 ),  
 );  
 }),  
 ),  
 ),  
 )  
 ],  
 ),  
 ),  
 // floatingActionButton: FloatingActionButton(  
 // onPressed: () {  
 // //\_incrementCounter();  
 // Map<String, String> Details = {  
 // "Number of counts": "${\_counter.text}"  
 // };  
 // dbRef.push().set(  
 // Details); // we can also use dbRef.push().set(Details);// this will generate unique key on every entry.  
 // },  
 // tooltip: 'Increment',  
 // child: const Icon(Icons.add),  
 // ), // This trailing comma makes auto-formatting nicer for build methods.  
 );  
 }  
  
 Future<void> showMyDialog(String title, String id) async {  
 editcontroller.text = title;  
 return showDialog(  
 context: context,  
 builder: (BuildContext context) {  
 return AlertDialog(  
 title: Text('update'),  
 content: Container(  
 child: TextField(  
 controller: editcontroller,  
 decoration: InputDecoration(  
 hintText: 'edit',  
 ),  
 ),  
 ),  
 actions: [  
 TextButton(  
 onPressed: () {  
 Navigator.*pop*(context);  
 },  
 child: Text('Cancel'),  
 ),  
 TextButton(  
 onPressed: () {  
 Navigator.*pop*(context);  
 dbRef.child(id).update({"title": editcontroller.text});  
 },  
 child: Text('Update'),  
 )  
 ],  
 );  
 });  
 }  
}

**Authentication using email and password:**

<https://www.youtube.com/watch?v=Dyu-tcX0H7M>

<https://www.youtube.com/watch?v=4vKiJZNPhss>

**Authentication using username:**