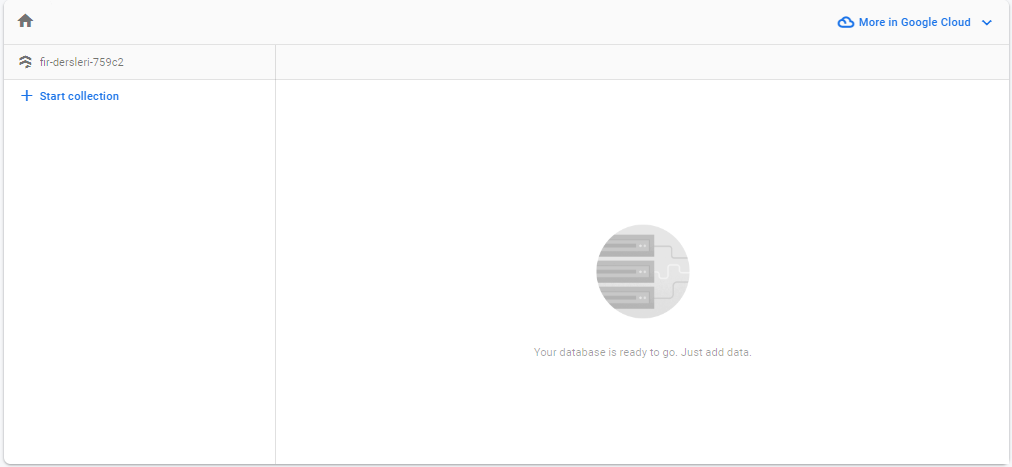
Bölüm 2 – Firebase Storage İşlemleri

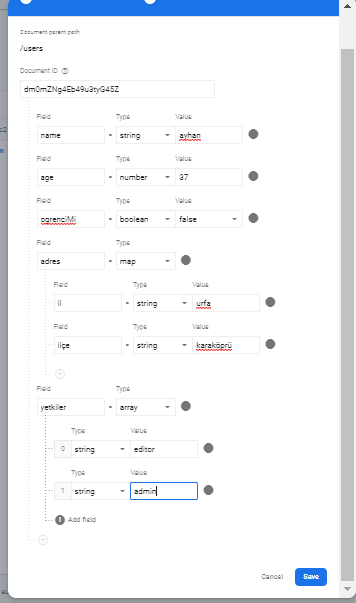
**Ders 1 : FireStore Collection ve Document Kavramları**

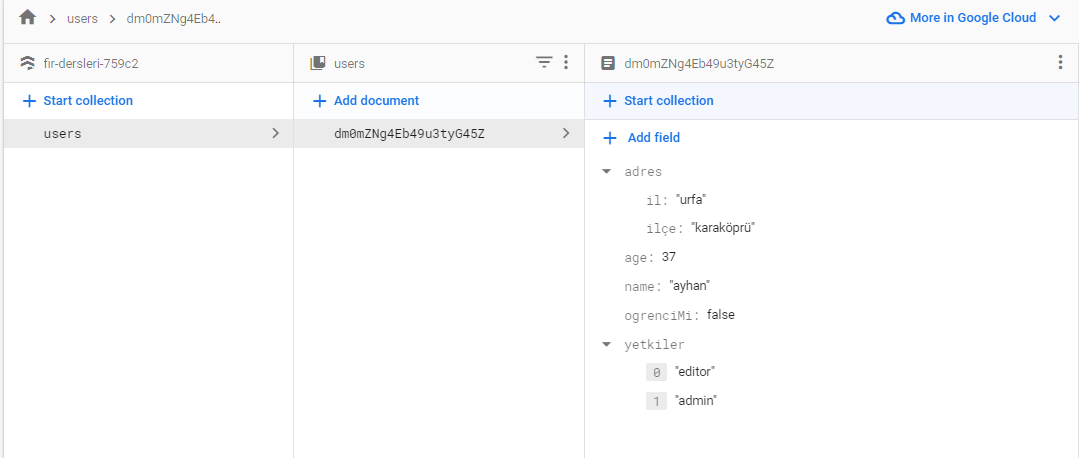
<https://console.firebase.google.com/u/0/project/fir-dersleri-759c2/overview> adresindeki projemize girelim.

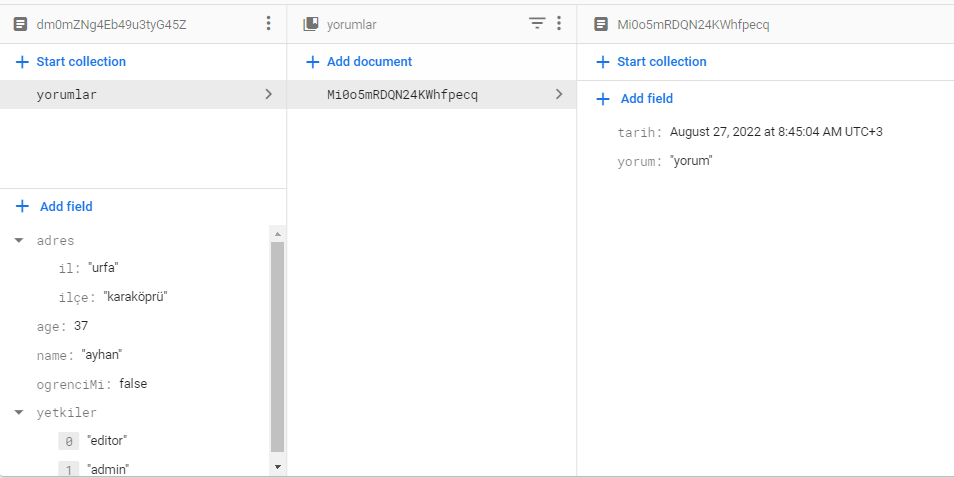
Build🡪Firestore Database🡪Create Database🡪Start in test mode(okuma ve yazmaya izinleri ver)🡪Cloud Firestore Location ( eur3 (europe-west) ) seçilir. Alttaki alana ulaşılır.



Veritabanından sonra collection oluşturabiliriz.







**Ders 2 : Firestore veri ekleme Set Data ve Add Metotları,FieldValue Örnekleri**

Projemize firestore\_islemleri.dart isminde bir dosya ekleyelim.

import 'package:flutter/material.dart';

import 'dart:ffi';

class FireStoreIslemleri extends StatelessWidget {

  const FireStoreIslemleri({Key? key}) : super(key: key);

  @override

  Widget build(BuildContext context) {

    return Scaffold(

      appBar: AppBar(

        title: const Text("Firestore İşelmeleri"),

      ),

      body: Center(

          child: Column(

        children: [

          ElevatedButton(

            onPressed: () => veriEklemeAdd(),

            child: const Text("Veri Ekle Add"),

          ),

        ],

      )),

    );

  }

  veriEklemeAdd() {}

}

Dart dependency ye gidilerek cloud\_firestore projeye eklenir.

<https://firebase.flutter.dev/docs/firestore/usage> adresine gidilir.

import 'package:cloud\_firestore/cloud\_firestore.dart';

class FireStoreIslemleri extends StatelessWidget {

   FireStoreIslemleri({Key? key}) : super(key: key);

  final FirebaseFirestore firestore = FirebaseFirestore.instance;

**not:** minSdk 21 olmalı (android/app/build.gradle)

Veri Ekleme Add ile verileri ekleyelim.(veriEklemeAdd() metoduna );

body: Center(

          child: Column(

        children: [

          ElevatedButton(

            onPressed: () => veriEklemeAdd(),

            child: const Text("Veri Ekle Add"),

          ),

veriEklemeAdd() async {

    Map<String, dynamic> \_eklenecekUser = <String, dynamic>{};

    \_eklenecekUser['isim'] = 'meral';

    \_eklenecekUser['yas'] = 29;

    \_eklenecekUser['ogrenciMİ'] = false;

    \_eklenecekUser['adres'] = {'il': 'ankara', 'ilce': 'mamak'};

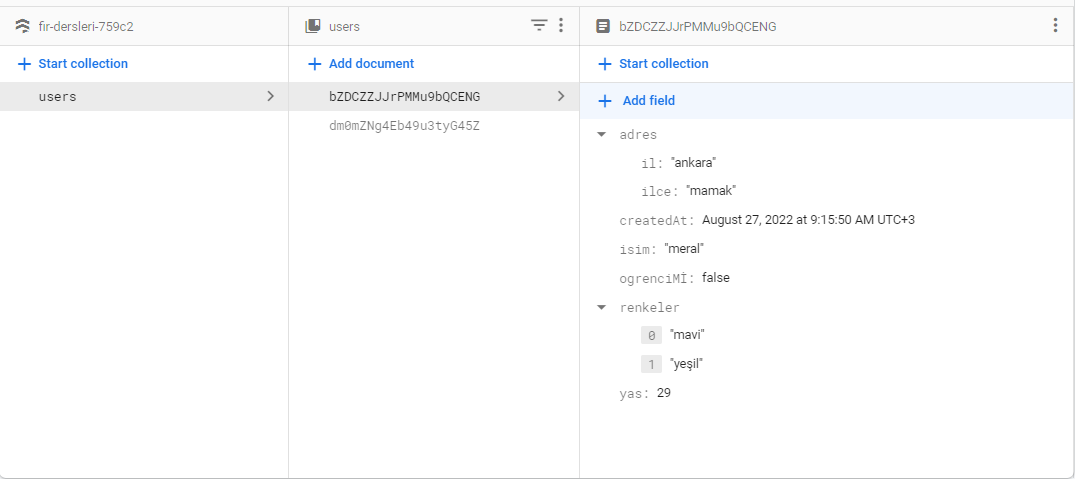
    \_eklenecekUser['renkeler'] = FieldValue.arrayUnion(['mavi', 'yeşil']);

    \_eklenecekUser['createdAt'] = FieldValue.serverTimestamp();

    await \_firestore.collection('users').add(\_eklenecekUser);

  }

}



veriEklemeSet adında bir buton ekleyelim ve metodu tıklanma olayına verelim.

ElevatedButton(

              onPressed: () => veriEklemeSet(),

              style: ElevatedButton.styleFrom(primary: Colors.green),

              child: const Text('Veri Ekle Set'),

            ),

veriEklemeSet() async {

    //users kullanıcısına ait id bilgisini \_yeniDocID ye atalım

    var \_yeniDocID = \_firestore.collection('users').doc().id;

    await \_firestore.doc('users/$\_yeniDocID') .set({'isim': 'emre', 'userID': \_yeniDocID});

//SetOptions demesek set çalışınca kullanıcının önceki verilerinin üzerine yazar.

//merge:true diyerek önceki bilgilerle birleştir diyoruz.

//FieldValue.increment ile belirtilen içeriği belirtilen şekilde arttırır.

    await \_firestore.doc('users/sNdIJXGgPUPl5L3sZGQK').set(

        {'okul': 'Ege Üniversitesi', 'yas': FieldValue.increment(-5)},

        SetOptions(merge: true));

  }

**Ders 3 : FireStore Veri Güncelleme ve Silme (Update-Delete)**

**Veri Güncelleme**

ElevatedButton(

              onPressed: () => veriGuncelleme(),

              style: ElevatedButton.styleFrom(primary: Colors.yellow),

              child: const Text('Veri Güncelle'),

            ),

 veriGuncelleme() async {

    await \_firestore.doc('users/sNdIJXGgPUPl5L3sZGQK').update({'adres.ilce': 'yeni ilçe'});

  }

**Veri Silme**

ElevatedButton(

              onPressed: () => veriSil(),

              style: ElevatedButton.styleFrom(primary: Colors.red),

              child: const Text('Veri Sil'),

            ),

veriSil() async {

    await \_firestore.doc('users/sNdIJXGgPUPl5L3sZGQK').delete();

    //Belirtilen kullanıcı id sini siler

    //veri silme için aşağıdaki gibi update de kullanılır.FieldValue.delete() ile içerik silinir.

    await \_firestore.doc('users/sNdIJXGgPUPl5L3sZGQK').update({'okul': FieldValue.delete()});

  }

**Ders 4 : Verilerin One Time Okunması, Eleman Sayısının Bulunması**

**Veritabanından Verilerin Okunması**

**OneTime Okunması**

ElevatedButton(

              onPressed: () => veriOkuOneTime(),

              style: ElevatedButton.styleFrom(primary: Colors.pink),

              child: const Text('Veri Oku One Time'),

            ),

//Veritabanından Verilerin Okunması

  //oneTime ile veriler sadece çağrıldığında görüntülenir.

  veriOkuOneTime() async {

    var \_usersDocuments = await \_firestore.collection('users').get();

    debugPrint(\_usersDocuments.size.toString());

    debugPrint(\_usersDocuments.docs.length.toString());

    for (var eleman in \_usersDocuments.docs) {

      debugPrint('Döküman id ${eleman.id}');

      Map userMap = eleman.data();

      debugPrint(userMap['isim']);

    }

    var \_emreDoc = await \_firestore.doc('users/dm0mZNg4Eb49u3tyG45Z').get();

    debugPrint(\_emreDoc.data()!['adres']['ilce'].toString());

  }

**Ders 5 : Stream ile Verilerdeki Anlık Değişimlerin İncelenmesi**

Verilerin RealTime Okunması:

ElevatedButton(

            onPressed: () => veriOkuRealTime(),

            style: ElevatedButton.styleFrom(primary: Colors.purple),

            child: const Text('Veri Oku Real Time'),

          ),

import 'dart:async';

class FireStoreIslemleri extends StatelessWidget {

  FireStoreIslemleri({Key? key}) : super(key: key);

  final FirebaseFirestore \_firestore = FirebaseFirestore.instance;

  StreamSubscription? \_userSubscribe;

//alttaki metot ile veritabanında olan tüm değişiklikler otomatik

  //olarak dinlenir.veritabanında ilgili id li herhangi bir kısmı değiştirdiğimiz

  //zaman veriler anlık projede güncellenecektir.

  veriOkuRealTime() async {

    //var \_userStream = await \_firestore.collection('users').snapshots();

    var \_userDocStream =

        \_firestore.doc('users/RhGwic4jK6RTCJJyKFGr').snapshots();

        //\_userSubscribe ile abonelik başlatılır(dinleme olayı)

    \_userSubscribe = \_userDocStream.listen((event) { //dinleme işlemi

      /\*  event.docChanges.forEach((element) {

        debugPrint(element.doc.data().toString());

      }); \*/

      debugPrint(event.data().toString());

    });

  }

Dinlemeyi Durdurma :

 ElevatedButton(

            onPressed: () => streamDurdur(),

            style: ElevatedButton.styleFrom(primary: Colors.teal),

            child: const Text('Stream Durdur'),

          ),

//Dinleme olayını durdurmak için şağıdaki metot kullanılır.

  streamDurdur() async {

    await \_userSubscribe?.cancel();

  }

**Ders 6 : Batch ile Toplu Veri Ekleme Silme Güncelleme**

 ElevatedButton(

            onPressed: () => batchKavrami(),

            style: ElevatedButton.styleFrom(primary: Colors.indigo),

            child: const Text('Batch Kavramı'),

          ),

 batchKavrami() async {

    WriteBatch \_batch = \_firestore.batch();

    CollectionReference \_counterColRef = \_firestore.collection('counter');

/\*

    Batch ile ekleme işlemleri

    100 adet id oluşturuldu.

    for (int i = 0; i < 100; i++) {

      var \_yeniDoc = \_counterColRef.doc();

      \_batch.set(\_yeniDoc, {'sayac': ++i, 'id':\_yeniDoc.id});

    }

\*/

/\*

     Batch ile güncelleme işlemleri

    var \_counterDocs = await \_counterColRef.get();

    \_counterDocs.docs.forEach((element) {

      \_batch.update(

          element.reference, {'createdAt': FieldValue.serverTimestamp()});

    });

\*/

/\*

    Batch ile toplu silme işlemleri

    var \_counterDocs = await \_counterColRef.get();

    \_counterDocs.docs.forEach((element) {

      \_batch.delete(element.reference);

    });

\*/

    //commit ile toplu değişiklikleri ekleriz.

    await \_batch.commit();

  }

**Ders 7 : Transaction Kavramı ve Örnek**

ElevatedButton(

            onPressed: () => transactionKavrami(),

            style: ElevatedButton.styleFrom(primary: Colors.brown),

            child: const Text('Transaction Kavramı'),

          ),

transactionKavrami() async {

    \_firestore.runTransaction((transaction) async {

      //1emrenin bakiyesini öğren

      //emreden 100 lira düş

      //hasana 100 lira ekle

      DocumentReference<Map<String, dynamic>> emreRef =

          \_firestore.doc('users/lODl1rILhnEeqeiDjBbj');//id leri değiştir.

      DocumentReference<Map<String, dynamic>> hasanRef =

          \_firestore.doc('users/UdpwK3unAKMMciZWUjKc');

      var \_emreSnapshot = await transaction.get(emreRef);

      var \_emreBakiye = \_emreSnapshot.data()!['para'];

      if (\_emreBakiye > 100) {

        var \_yeniBakiye = \_emreSnapshot.data()!['para'] - 100;

        transaction.update(emreRef, {'para': \_yeniBakiye});

        transaction.update(hasanRef, {'para': FieldValue.increment(100)});

      }

    });

  }

**Ders 8 : Veritabanında Sorgulama Yapmak**

 ElevatedButton(

            onPressed: () => queryingData(),

            style: ElevatedButton.styleFrom(primary: Colors.cyan),

            child: const Text('Veri Sorgulama'),

          ),

queryingData() async {

    var \_userRef = \_firestore.collection('users').limit(5);

    var \_sonuc = await \_userRef.where('renkler', arrayContains: 'mavi').get();

/\*

    for (var user in \_sonuc.docs) {

      debugPrint(user.data().toString());

    } \*/

    var \_sirala = await \_userRef.orderBy('yas', descending: false).get();

    /\*  for (var user in \_sirala.docs) {

      debugPrint(user.data().toString());

    } \*/

    var \_stringSearch = await \_userRef

        .orderBy('email')

        .startAt(['emre']).endAt(['emre' + '\uf8ff']).get();

    for (var user in \_stringSearch.docs) {

      debugPrint(user.data().toString());

    }

  }

**Ders 9 : Firebase Storage Dosya Yükleme ve Download URL Alma**

Dart: Add Dependency den firebase\_storage ve image\_picker paketleri indirilir.

Firebase Storage kısmına gidilir. <https://console.firebase.google.com/u/0/project/fir-dersleri-759c2/storage/fir-dersleri-759c2.appspot.com/files>

Aşağıdaki kütüphaneler paketler inmişse projeye import edilir.

import 'dart:io';

import 'package:image\_picker/image\_picker.dart';

import 'package:firebase\_storage/firebase\_storage.dart';

ElevatedButton(

            onPressed: () => kameraGaleriImageUpload(),

            style: ElevatedButton.styleFrom(primary: Colors.blue),

            child: const Text('Kamera Galeri Image Upload'),

          ),

kameraGaleriImageUpload() async {

    print('fonksiyon çalıstı');

    final ImagePicker \_picker = ImagePicker();

    XFile? \_file = await \_picker.pickImage(source: ImageSource.camera);

    print('image secildi ');

    if (\_file == null) {

      print('image null ');

    }

    var \_profileRef =

        FirebaseStorage.instance.ref('users/profil\_resimleri/user\_id');

    print(\_file?.name.toString());

    var \_task = \_profileRef.putFile(File(\_file!.path));

    debugPrint('yükleme başlatılacak');

    \_task.whenComplete(() async {

      var \_url = await \_profileRef.getDownloadURL();

      debugPrint('yükleme bitti');

      \_firestore

          .doc('users/RhGwic4jK6RTCJJyKFGr')

          .set({'profile\_pic': \_url.toString()}, SetOptions(merge: true));

      debugPrint(\_url);

    });

  }