

Teknoloji Fakültesi
Bilgisayar Mühendisliği

3301409 MOBİL PROGRAMLAMA NÖ
3301456 MOBİL PROGRAMLAMA NÖ
3311409 MOBİL PROGRAMLAMA İÖ
3311456 MOBİL PROGRAMLAMA İÖ

Dr. Öğretim Üyesi Ahmet Cevahir ÇINAR

Selçuk Üniversitesi Teknoloji Fakültesi Bilgisayar Mühendisliği Bölümü

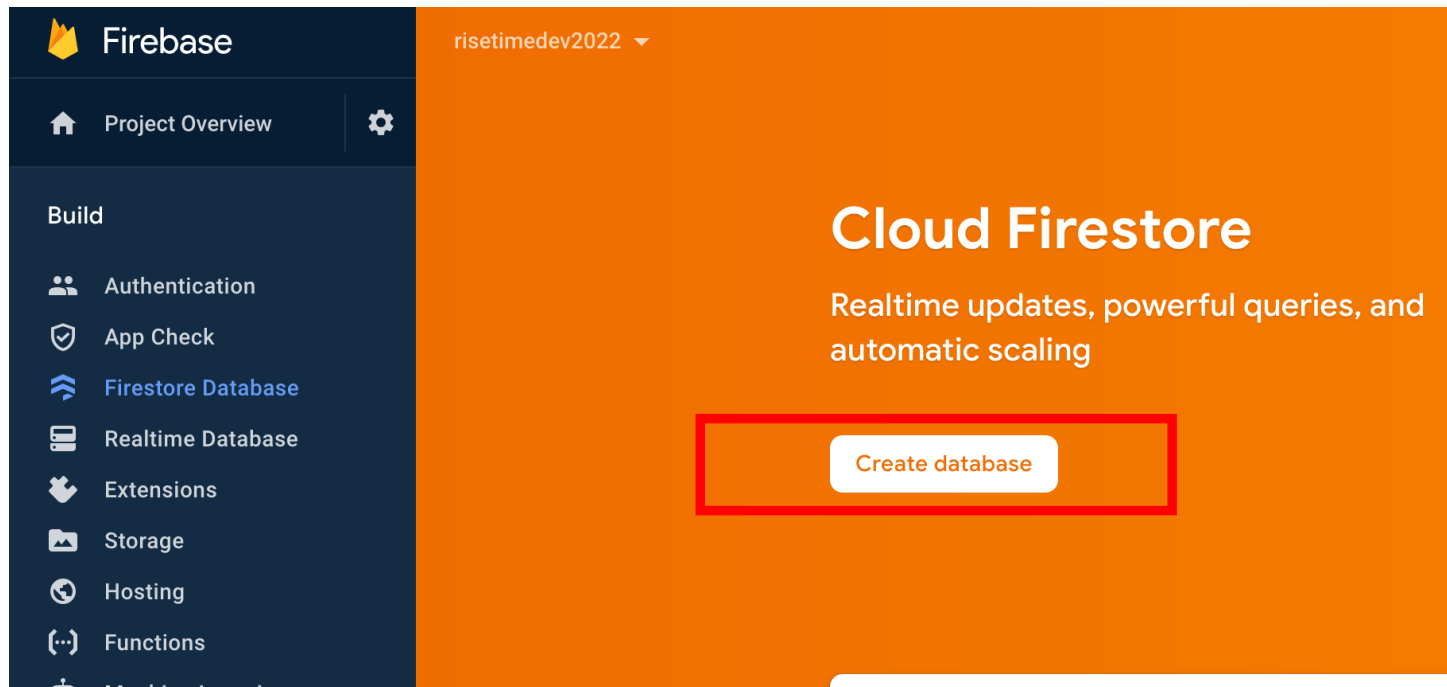
accinar@selcuk.edu.tr

HAFTA	TARİH	DERSİN İÇERİĞİ
1	9 Şubat 2022	Mobil Programlamaya genel bakış, tanıtım, derse giriş ve işlenişi ile ilgili yol haritasının belirlenmesi
2	16 Şubat 2022	Dart Programlama Dili
3	23 Şubat 2022	Geliştirme ortamının gereksinimleri ve kurulumlar (Flutter, Git, Android Studio vb.)
4	2 Mart 2022	Flutter uygulamalarının temelleri (klasör yapısı, pubspec.yaml dosyası, temel Flutter kütüphaneleri vb.), genel bileşenler, Flutter uygulamalarının yaşam döngüsü
5	9 Mart 2022	Flutter ile kullanıcı ara yüzü tasarlamak (MaterialApp, Scaffold, AppBar, SafeArea, SnackBar vb.)
6	16 Mart 2022	Widgetler (Value, Layout, Navigation, vb.) ve Widgetleri stillendirmek
7	23 Mart 2022	Geliştirilecek projelerin derste değerlendirilmesi, yönlendirme ve yardımcı olunması
8	30 Mart 2022	Projeyi Github'a yükleme ve Github Pages üzerinden yayına alma – Youtube'da var
	<u>1 Nisan 2022 20.00</u>	<u>Ara Sınav, Final ve Bütünleme sınavlarının nasıl yapılacağı ile ilgili YouTube toplantısı 😊</u>
	4-10 Nisan 2022	ARA SINAV SÜRECİ
9	13 Nisan 2022	Gestures ve Asenkron Programlama
10	20 Nisan 2022	Dosya işlemleri ve API ile çalışmak
11	27 Nisan 2022	SQLite ile veritabanı işlemleri
12	11 Mayıs 2022	Firestore ile veritabanı kullanımı
13	18 Mayıs 2022	Firestore ile Provider destekli veritabanı üzerinde CRUD işlemleri
14	25 Mayıs 2022	Animasyonlar ve Grafikler

12-13-14. derslerimizdeki konuları öğrenmek için RiseTime şirketinin kendi iç işleyişinde kullanmak üzere istemiş olduğu bir uygulamayı birlikte gerçekleştiriyorduk. 13.hafta kapsamında Domain Firmalarını ekleme, güncelleme, silme işlemlerini Provider destekli Firestore ile yapacağız.

Firestore entegrasyonunu yapalım

- <https://firebase.google.com/docs/firestore/quickstart> adresindeki adımları takip edelim



Create database



- 1 Secure rules for Cloud Firestore
- 2 Set Cloud Firestore location

After you define your data structure, **you will need to write rules to secure your data.**

[Learn more](#)

☐ **Start in production mode**

Your data is private by default. Client read/write access will only be granted as specified by your security rules.

☒ **Start in test mode**

Your data is open by default to enable quick setup. However, you must update your security rules within 30 days to enable long-term client read/write access.

```
rules_version = '2';
service cloud.firestore {
  match /databases/{database}/documents {
    match /{document=**} {
      allow read, write: if
        request.time < timestamp.date(2022, 6, 10);
    }
  }
}
```

The default security rules for test mode allow anyone with your database reference to view, edit and delete all data in your database for the next 30 days

Enabling Cloud Firestore will prevent you from using Cloud Datastore with this project, notably from the associated App Engine app

Cancel

Next

Create database



Secure rules for Cloud Firestore

2

Set Cloud Firestore location

Your location setting is where your Cloud Firestore data will be stored.



After you set this location, you cannot change it later. Also, this location setting will be the location for your default Cloud Storage bucket.

[Learn more](#)

Cloud Firestore location

eur3 (europe-west)



Enabling Cloud Firestore will prevent you from using Cloud Datastore with this project, notably from the associated App Engine app

Cancel

Enable

Create database



Secure rules for Cloud Firestore

2

Set Cloud Firestore location

Your location setting is where your Cloud Firestore data will be stored.



After you set this location, you cannot change it later. Also, this location setting will be the location for your default Cloud Storage bucket.

[Learn more](#)

Provisioning Cloud Firestore...

Cloud Firestore location

eur3 (europe-west)

Enabling Cloud Firestore will prevent you from using Cloud Datastore with this project, notably from the associated App Engine app

Cancel

Enable

How much will Cloud

Cloud Firestore

Data Rules Indexes Usage

 Protect your Cloud Firestore resources from abuse, such as billing fraud or phishing [Configure App Check](#) 

 Prototype and test end-to-end with the Local Emulator Suite, now with Firebase Authentication [Get started](#)  



 risetimedev2022

[+ Start collection](#)



Your database is ready to go. Just add data.

cloud_firestore 3.1.14

- https://pub.dev/packages/cloud_firestore kütüphanesini yükleyelim.
- Ayrıntılı kullanım bilgisi:
<https://firebase.google.com/docs/firestore/quickstart>

- <https://pub.dev/packages/uuid> paketi ile id üreteceğiz.

🏠 > users > cc3H0r3swvgjb...		
risetimedev-646e9	users	cc3H0r3swvgjb4uXGVwoYg6z4sk2
+ Start collection	+ Add document	+ Start collection
domainfirms	cc3H0r3swvgjb4uXGVwoYg6z4sk2 >	+ Add field
users >	krYwryCkTnVXKunKgBtgd5tWEZ52	user_email: "ccc@ccc.com"
	sWy6brHMTdhgws0URqZ9DttUDfB3	user_id: "cc3H0r3swvgjb4uXGVwoYg6z4sk2"
		user_name: "ccc"
		user_password: "ccccc"

🏠 > domainfirms > 9bd22bc0-6fe8-...		
risetimedev-646e9	domainfirms	9bd22bc0-6fe8-43dd-9eda-74ef3a23455c
+ Start collection	+ Add document	+ Start collection
domainfirms >	972add6-50d7-42b6-8602-8ea64f03f2f	+ Add field
users	9bd22bc0-6fe8-43dd-9eda-74ef3a23455	id: "9bd22bc0-6fe8-43dd-9eda-74ef3a23455c"
		loginname: "aa"
		loginpassword: "aaa"
		name: "a"
		web: "aaaa"

Hata ile ilgili hatırlatma

`import 'package:uuid/uuid.dart';` eklediğiniz halde hata alıyorsanız Terminal üzerinden **flutter pub get** komutu ile paketlerini yeniden çağırın.

Yeni kütüphane eklediğinizde projeniz kütüphaneyi tanımazsa **flutter pub get** komutunu denemeyi ihmal etmeyin!

Domain Firması modelini hatırlayalım ve /models/ klasörüne domain_firms.dart dosyasını oluşturalım

DomainFirms

id

name

loginname

loginpassword

web

```
class DomainFirms {  
  final String id;  
  final String name;  
  final String loginname;  
  final String loginpassword;  
  final String web;  
  
  DomainFirms(  
    {this.id, this.name, this.loginname, this.loginpassword, this.web});  
  
  Map<String, dynamic> toMap() {  
    return {  
      'id': id,  
      'name': name,  
      'loginname': loginname,  
      'loginpassword': loginpassword,  
      'web': web  
    };  
  }  
}  
  
DomainFirms.fromFirestore(Map<String, dynamic> firestore)  
  : id = firestore['id'],  
    name = firestore['name'],  
    loginname = firestore['loginname'],  
    loginpassword = firestore['loginpassword'],  
    web = firestore['web'];  
}
```

Firestore servislerini hazırlayalım

```
firestore_domainfirms_service.dart x
1  import 'package:cloud_firestore/cloud_firestore.dart';
2  import 'package:risetimedev/models/domain_firms.dart';
3
4  class FirestoreDomainFirmsService {
5      FirebaseFirestore _db = FirebaseFirestore.instance;
6
7      Future<void> saveDomainFirm(DomainFirms domainfirm) {
8          return _db
9              .collection('domainfirms')
10             .doc(domainfirm.id)
11             .set(domainfirm.toMap());
12     }
13
14     Stream<List<DomainFirms>> getDomainFirms() {
15         return _db.collection('domainfirms').snapshots().map((snapshot) => snapshot
16             .docs
17             .map((document) => DomainFirms.fromFirestore(document.data()))
18             .toList());
19     }
20
21     Future<void> removeProduct(String id) {
22         return _db.collection('domainfirms').doc(id).delete();
23     }
24 }
```

DomainFirmsProvider'ın Hazırlanması

- Değişkenler tanımlanır
- Getter'lar ayarlanır
- Setter'lar ayarlanır

```
changeName(String value) {  
  _name = value;  
  notifyListeners();  
}  
  
changeLoginName(String value) {  
  _loginname = value;  
  notifyListeners();  
}  
  
changeLoginPassword(String value) {  
  _loginpassword = value;  
  notifyListeners();  
}  
  
changeWeb(String value) {  
  _web = value;  
  notifyListeners();  
}
```

```
import 'package:flutter/material.dart';  
import 'package:risetimedev/models/domain_firms.dart';  
import 'package:risetimedev/services/domainfirms/firestore_domainfirms_service.dart';  
import 'package:uuid/uuid.dart';  
  
class DomainFirmsProvider with ChangeNotifier {  
  
  final firestoreService = FirestoreDomainFirmsService();  
  String _id;  
  String _name;  
  String _loginname;  
  String _loginpassword;  
  String _web;  
  
  var uuid = Uuid();  
  
  String get id => _id;  
  
  String get name => _name;  
  
  String get loginname => _loginname;  
  
  String get loginpassword => _loginpassword;  
  
  String get web => _web;
```

```

loadValues(DomainFirms domainfirm) {
    _id = domainfirm.id;
    _name = domainfirm.name;
    _loginname = domainfirm.loginname;
    _loginpassword = domainfirm.loginpassword;
    _web = domainfirm.web;
}

saveProduct() {
    print(_id);
    if (_id == null) {
        var newDomainFirm = DomainFirms(
            id: uuid.v4(),
            name: name,
            loginname: loginname,
            loginpassword: loginpassword,
            web: web);
        firestoreService.saveDomainFirm(newDomainFirm);
    } else {
        //Update
        var updatedDomainFirm = DomainFirms(
            id:id,
            name: name,
            loginname: loginname,
            loginpassword: loginpassword,
            web: web);
        firestoreService.saveDomainFirm(updatedDomainFirm);
    }
}

removeProduct(String id) {
    firestoreService.removeProduct(id);
}
}

```

Provider'ları ekleyelim

```
return MultiProvider(  
  providers: [  
    Provider<FlutterFireAuthService>(  
      create: (_) => FlutterFireAuthService(FirebaseAuth.instance),  
    ), // Provider  
    StreamProvider(  
      create: (context) =>  
        context.read<FlutterFireAuthService>().authStateChanges,  
      initialData: null,  
    ), // StreamProvider  
    ChangeNotifierProvider(create: (context) => DomainFirmsProvider()),  
    StreamProvider(create: (context) => FirestoreDomainFirmsService().getDomainFirms()),  
    ChangeNotifierProvider(create: (context) => MenuDrawerNotifier()),  
  ],  
);
```




baban
bb

bbbb



ahhh
aa

aaaa

domainFirmsView.dart



←

Add Domain Firm

Domain Firm Name

Domain Firm Login Name

Domain Firm Login Password

Domain Firm Web

Save

←

Edit Domain Firm

ahhh

aa

aaa

aaaa

Save

Delete

editDomainFirmsView.dart

Faydalı Kaynaklar

- <https://medium.com/flutter-community/flutter-crud-operations-using-firebase-cloud-firestore-a7ef38bbf027>
- <https://www.youtube.com/watch?v=YZt92I2V704> ve https://www.youtube.com/watch?v=dQHQ1_Jen6c videoları olabildiğince temiz bir şekilde Flutter+Firestore+Provider ekseninde örnek bir anlatım yapmaktadır.

Bu ders kapsamında anlatılan kodlara aşağıdaki adresten erişebilirsiniz:

<https://github.com/ahmetcevahircinar/risetimedev2022>

➤ Sorular?