



DISCORD CLONE

UMUT KONAK - 210610010

[Proje Dosyaları](#)

```

1  @override
2  Widget build(BuildContext context) {
3      return MaterialApp(
4          title: 'Material App',
5          debugShowCheckedModeBanner: false,
6          theme: ThemeData.dark(),
7          home: AuthGate(),
8      );
9
10 }
11 }
12

```

Proje başlangıç home: AuthGate sayfası bu sayfayı inceleyelim

```

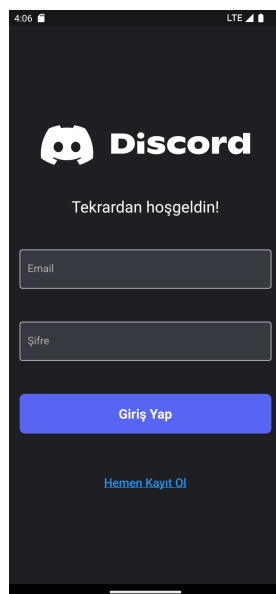
1  import 'package:firebase_auth/firebase_auth.dart';
2  import 'package:flutter/material.dart';
3  import 'package:flutter_whatsapp_clone/pages/home_page.dart';
4  import 'package:flutter_whatsapp_clone/services/auth/login_or_register.dart';
5
6  class AuthGate extends StatelessWidget {
7      const AuthGate({super.key});
8
9      @override
10     Widget build(BuildContext context) {
11         return Scaffold(
12             body: StreamBuilder(
13                 stream: FirebaseAuth.instance.authStateChanges(),
14                 builder: (context, snapshot) {
15                     // user is logged in
16                     if (snapshot.hasData) {
17                         return const Homepage();
18                     }
19
20                     // user is NOT log in
21                     else {
22                         return const LoginOrRegister();
23                     }
24                 },
25             ),
26         );
27     }
28 }
29

```

Firestore_auth kütüphanesinde bulunan authStateChanges fonksiyonu sayesinde kullanıcı giriş yapmış ise Anasayfaya sayfasına yönlendirilir. Kullanıcı giriş yapmamışsa Kayıt ol/Giriş Yap sayfasına yönlendirilir.



Eposta ve şifre kontrolü sağlayan iki adet controller bulunuyor. Bu iki controller email ve şifre alanlarına girilen değerleri değişken olarak tutuyor. signIn fonksiyonu ile de firebase giriş yapma isteği gönderilir. Giriş başarısız ise veya herhangi bir hata ile karşılaşırsa hata mesajı kullanıcıya gösterilir.



Login Sayfası

Authentication

Users Sign-in method Templates Usage Settings Extensions

Add user

Identifier	Providers	Created ↓	Signed in	User UID
test3@gmail.com	📧	Dec 17, 2023	Dec 29, 2023	A1CqQ9LD4kWyU8RcwGvE6N...
test2@gmail.com	📧	Dec 17, 2023	Dec 26, 2023	QGK2SG7RfZhyuFspAvpU...
test@gmail.com	📧	Dec 17, 2023	Jan 16, 2024	GKECN5s78oSExUwVR8siEXN...

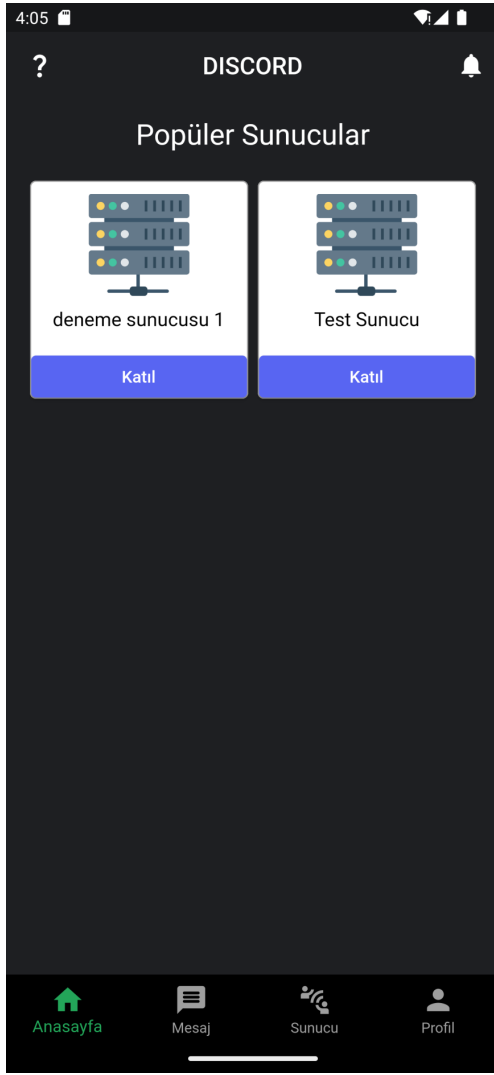
Rows per page: 50 1 - 3 of 3

Firebase'de kayıtlı kullanıcılar bu şekilde gözüküyor.

Kullanıcı giriş yaptıktan Anasayfa açılır. Bu sayfamızda popüler sunucular yer alıyor. Sunucu sayısı az olduğu için firebase'de servers collection altında bulunan dokümanlar gösteriliyor. Kodu ise şu şekildedir:

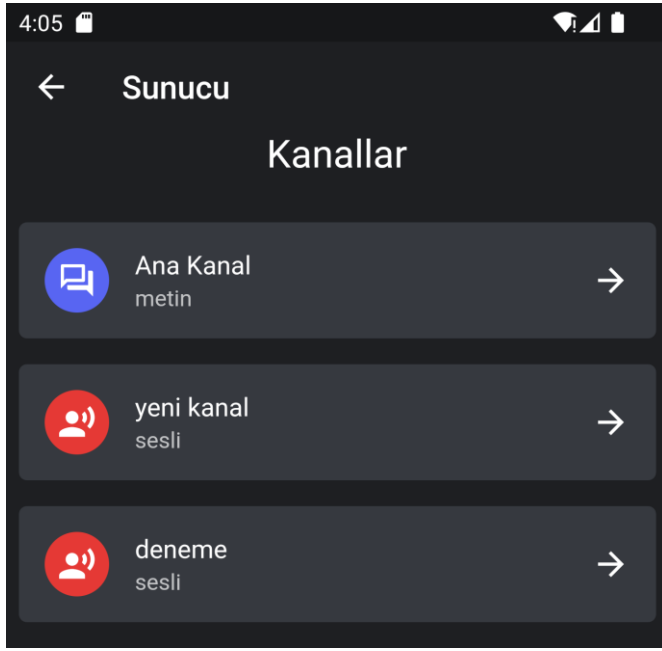
```
1  /* get all servers
2  Stream<QuerySnapshot> getAllServers() {
3    return _fireStore.collection('servers').snapshots();
4  }
```

Firebase'de ise server collection şu şekildedir:



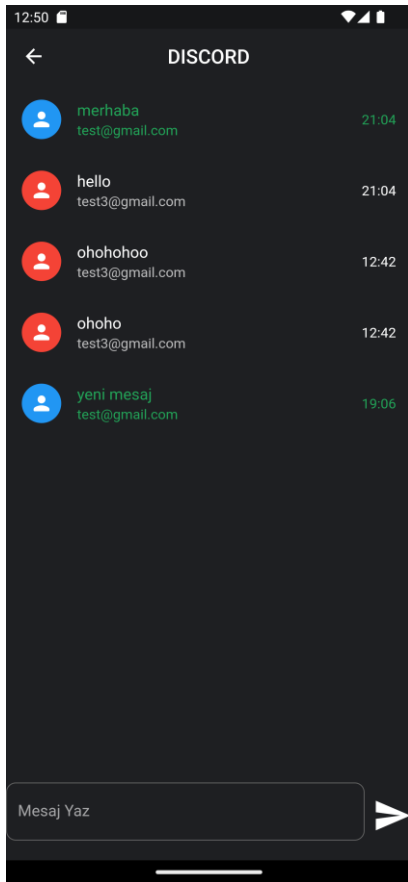
Projede mesaj bölümünde listelenen kullanıcılar ya da sunucudaki kanallar, Sunucuları listelemedeki koda benzer bir kullanıma sahiptir. Sadece collection ismi değişiyor.

Sunucularda iki farklı kanal türü bulunuyor. Metin ve ses kanalları;



1. Metin Kanalları

Metin kanallarında sunucuda bulunan kullanıcılar mesaj gönderebilir.

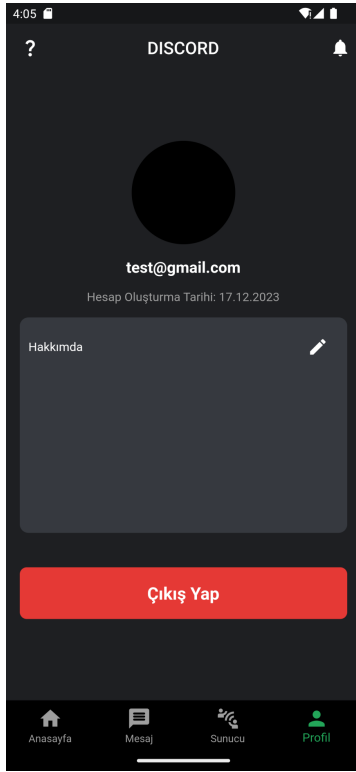


2. Ses Kanalları

Ses kanalına giriş yapıldığında flutter'ın Zegocloud kütüphanesi ile sesli görüntülü sohbet odası bağlanılıyor. (API girişi ile Zegocloud'un custom widgetleri kullanıldı)



Kullanıcının Profil sayfası ise şu şekildedir:



Bu sayfada bulunan bilgiler FirebaseAuth kütüphanesi ile şu kullanıcının uid bilgisi ile diğeri bilgileri görüntülenir. (Profil fotoğrafı, mail adres, hesap oluşturma tarihi, hakkında bilgileri)

Hakkımda kısmı düzenlenebilir şekilde ayarlandı. Sağ üstte bulunan düzenle butonuna tıklanınca bir textfield açılır buraya yazılan değer firebase users collection altında bulunan giriş yapmış kullanıcının dokümanın içerisindeki hakkında kısmını değiştirir.

Kodu:

```
1  /* Hakkında bölümünü güncelleme
2  Future<void> updateAboutText(String newText, userId) async {
3    try {
4      await FirebaseFirestore.instance
5        .collection('users')
6        .doc(userId)
7        .update({'aboutText': newText});
8    } catch (e) {
9      print('Hakkında kısmı güncellenirken bir sorun oluştu: $e');
10   }
11 }
```


Projede yer alan string ifadeleri farklı bir sınıfta tutuldu. Herhangi bir string ifadenin bulunması ve değiştirilmesine kolaylık sağlar, uygulamanın geliştirilmesi durumunda çoklu dil desteği için de avantaj sağlar.

Örneğin anasayfada yer alan birkaç string ifade:

```
1  return AlertDialog(  
2    title: Text(homeText.joinServer),  
3    content: Text(homeText.sureForJoinServer),  
4    actionsAlignment: MainAxisAlignment.center,  
5    actions: [  
6      ElevatedButton(  
7        style: ElevatedButton.styleFrom(  
8          backgroundColor: CustomColors().dcBlue,  
9        ),  
10       onPressed: () {  
11         joinAsServer(serverId, serverName, serverDesc, serverType);  
12         Navigator.of(context).pop();  
13       },  
14       child: Text(homeText.join),  
15     ),  
16     ElevatedButton(  
17       style: ElevatedButton.styleFrom(  
18         backgroundColor: CustomColors().dcBlue),  
19       onPressed: () {  
20         Navigator.of(context).pop();  
21       },  
22       child: Text(homeText.cancel),  
23     )  
24   ],  
25 );
```

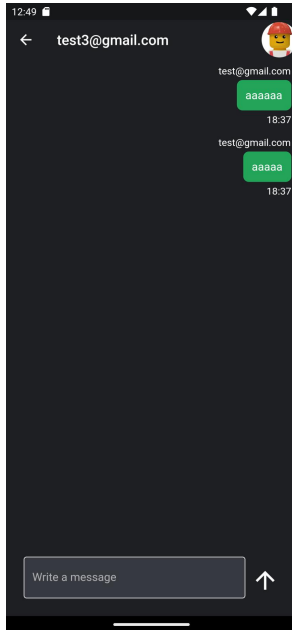
Home Text Class:

```
1  class HomeText {  
2    String popularServers = 'Popüler Sunucular';  
3    String joinServer = 'Sunucuya Katıl';  
4    String sureForJoinServer = 'Sunucuya katılmak istediğinizden emin misiniz?';  
5    String succesToJoinServer = 'Sunucuya katıldınız!';  
6    String join = 'Katıl';  
7    String cancel = 'İptal';  
8  }
```

Aynı zamanda projede bulunan renkler de ayrı bir class içerisinde tutuldu.



Projede mesaj sayfasında gönderilen mesajlar sol tarafta yer alırken alınan mesajlar ise sağ tarafta yer alıyor:



Kodu:

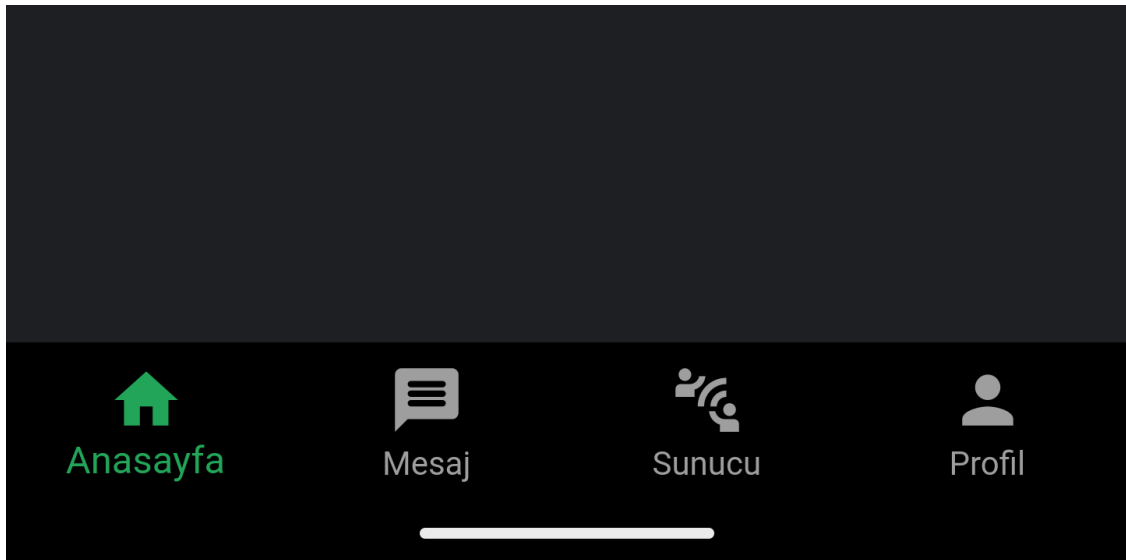
```

1
2 return Container(
3   alignment: alignment,
4   child: Padding(
5     padding: const EdgeInsets.all(8.0),
6     child: Column(
7       crossAxisAlignment:
8         (data['senderId'] == _firebaseAuth.currentUser!.uid
9           ? CrossAxisAlignment.end
10            : CrossAxisAlignment.start),
11       children: [
12         Text(data['senderEmail']),
13         SizedBox(
14           height: 5,
15         ),
16         ChatBubble(message: data['message']),
17         SizedBox(
18           height: 5,
19         ),
20         Text(formatTimestamp(data['timestamp'])),
21       ],
22     ),
23   ),
24 );

```

Data['senderId'] Firebase Auth ile şuanki kullanıcı olup olmadığı kontrol buna göre hizalama yapılır.

Son olarak bottom navigationbar'ı göstermek istiyorum.





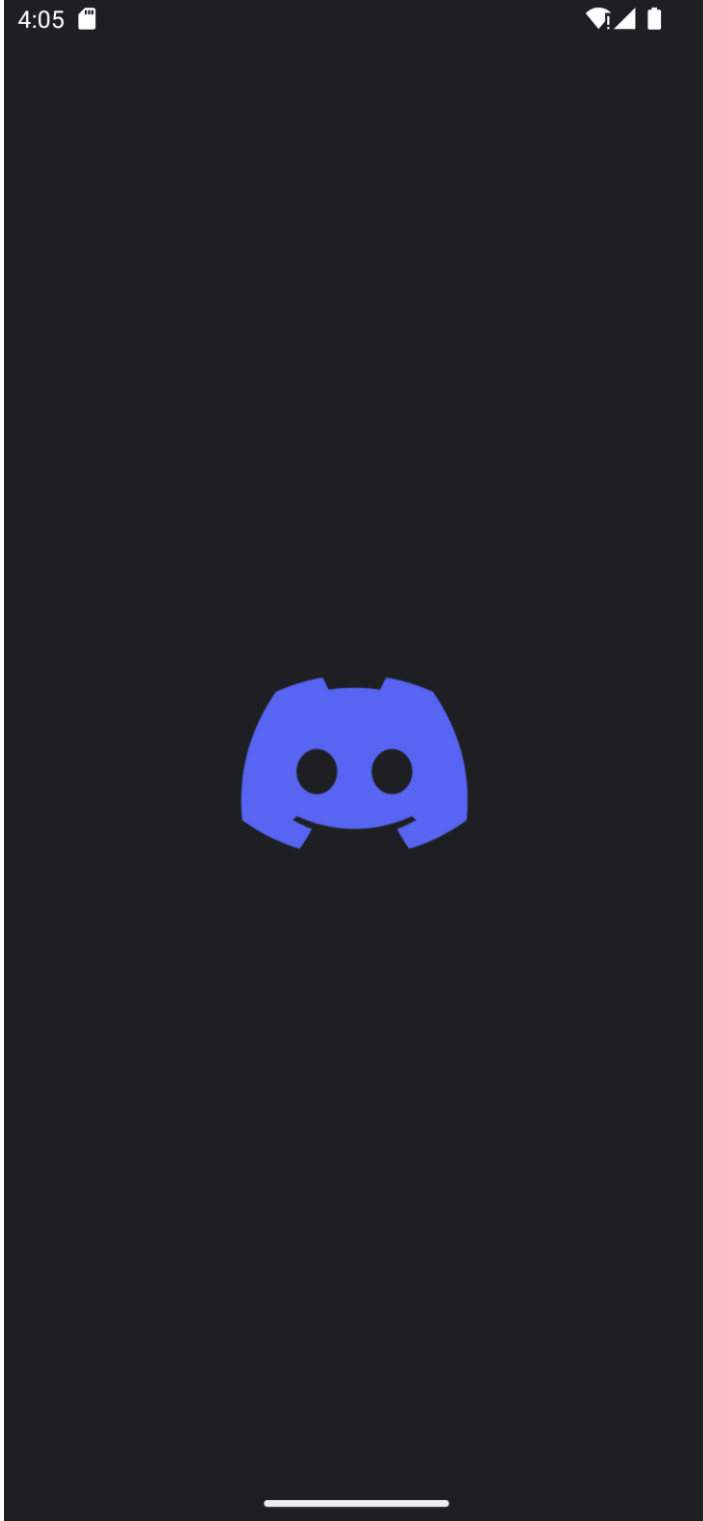
```

1  bottomNavigationBar: BottomNavigationBar(
2      type: BottomNavigationBarType.fixed,
3      iconSize: 28,
4      backgroundColor: Colors.black,
5      selectedItemColor: customColors.dcGreen,
6      unselectedItemColor: Colors.grey,
7      elevation: 0,
8      currentIndex: _currentIndex,
9      onTap: (int index) {
10         setState(() {
11             _currentIndex = index;
12         });
13     },
14     items: const [
15         BottomNavigationBarItem(
16             icon: Icon(Icons.home),
17             label: 'Anasayfa',
18         ),
19         BottomNavigationBarItem(
20             icon: Icon(Icons.message_rounded),
21             label: 'Mesaj',
22         ),
23         BottomNavigationBarItem(
24             icon: Icon(Icons.connect_without_contact),
25             label: 'Sunucu',
26         ),
27         BottomNavigationBarItem(
28             icon: Icon(Icons.person),
29             label: 'Profil',
30         ),
31     ]),
32     body: _pages[_currentIndex],

```

Bottom navbar ilk elemanın index 0'dır. İlk elemana tıklanıldığında `_pages[0]` olur ve burda `homeScreen` bulunur. Anasayfa gösterilir. Bu şekilde navbar ile pagination sağlanır.

Uygulamadan Görseller:





Discord

Hadi bir hesap oluřturalım!

Kayıt Ol

[Zaten bir hesabınız var mı?](#)

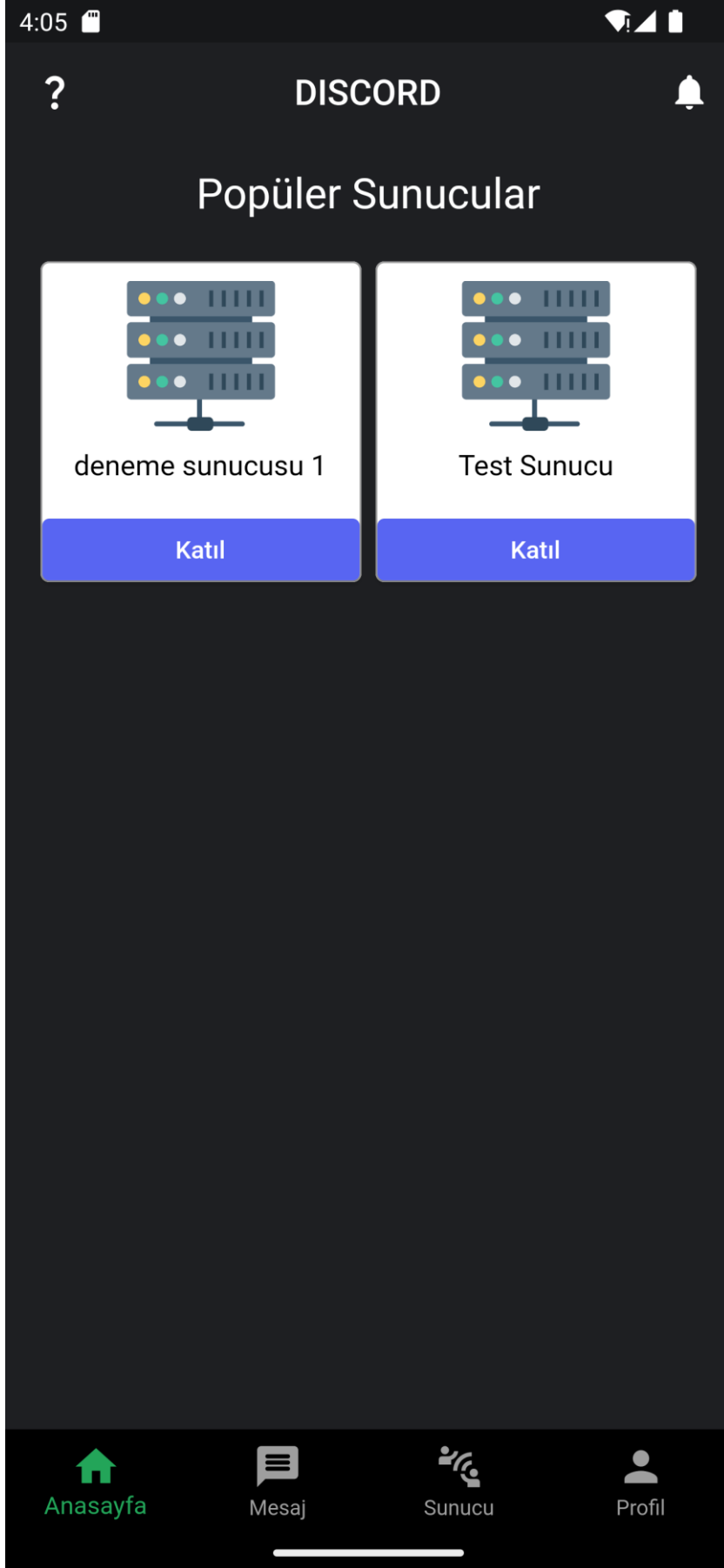


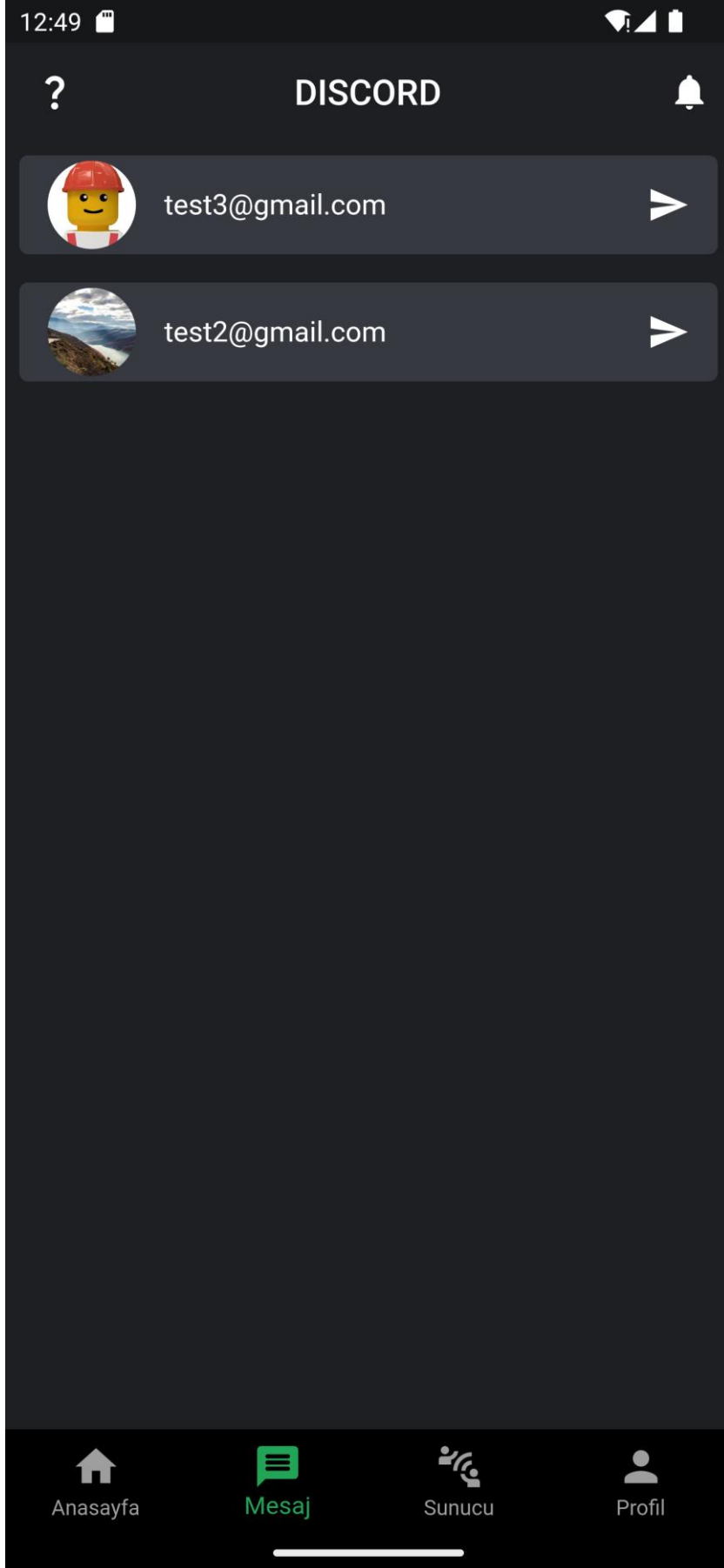
Discord

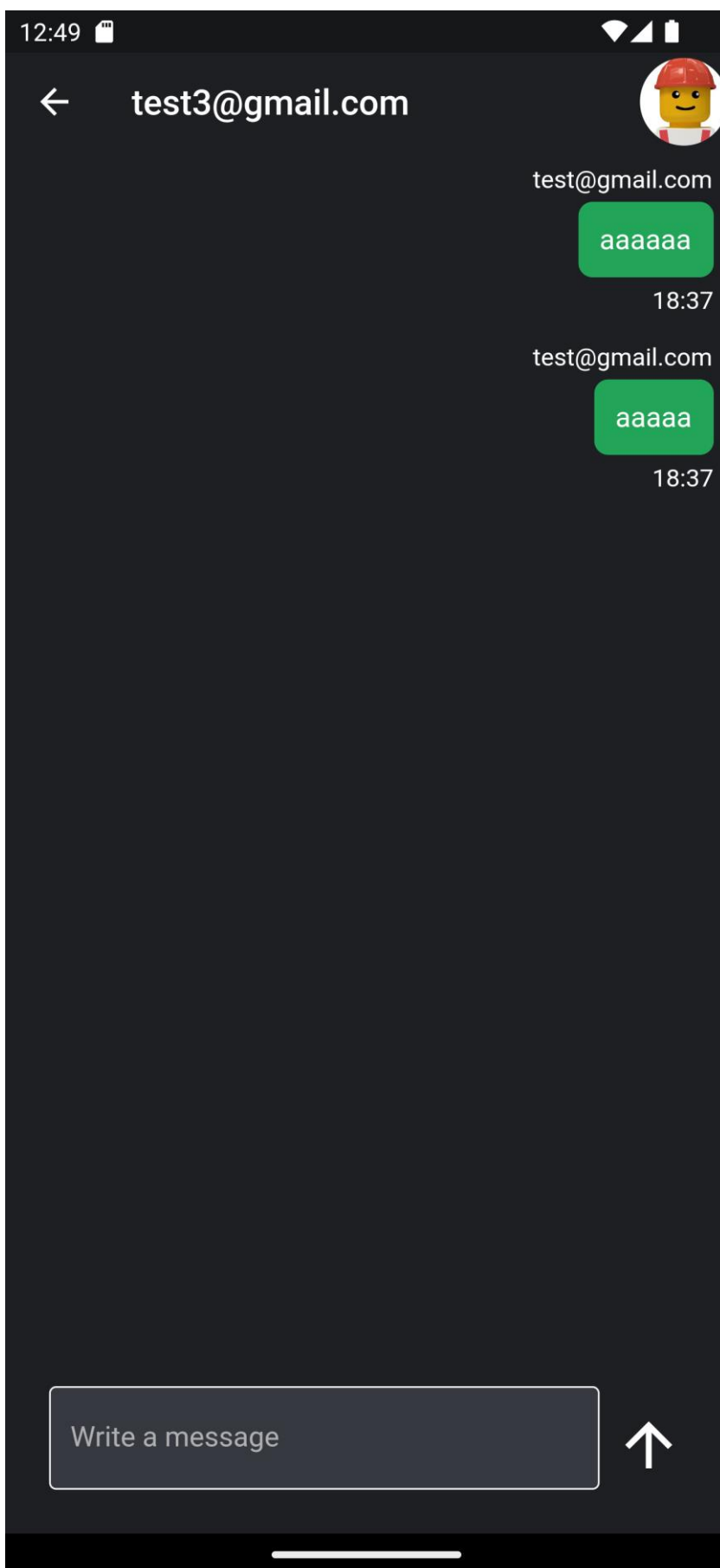
Tekrardan hoşgeldin!

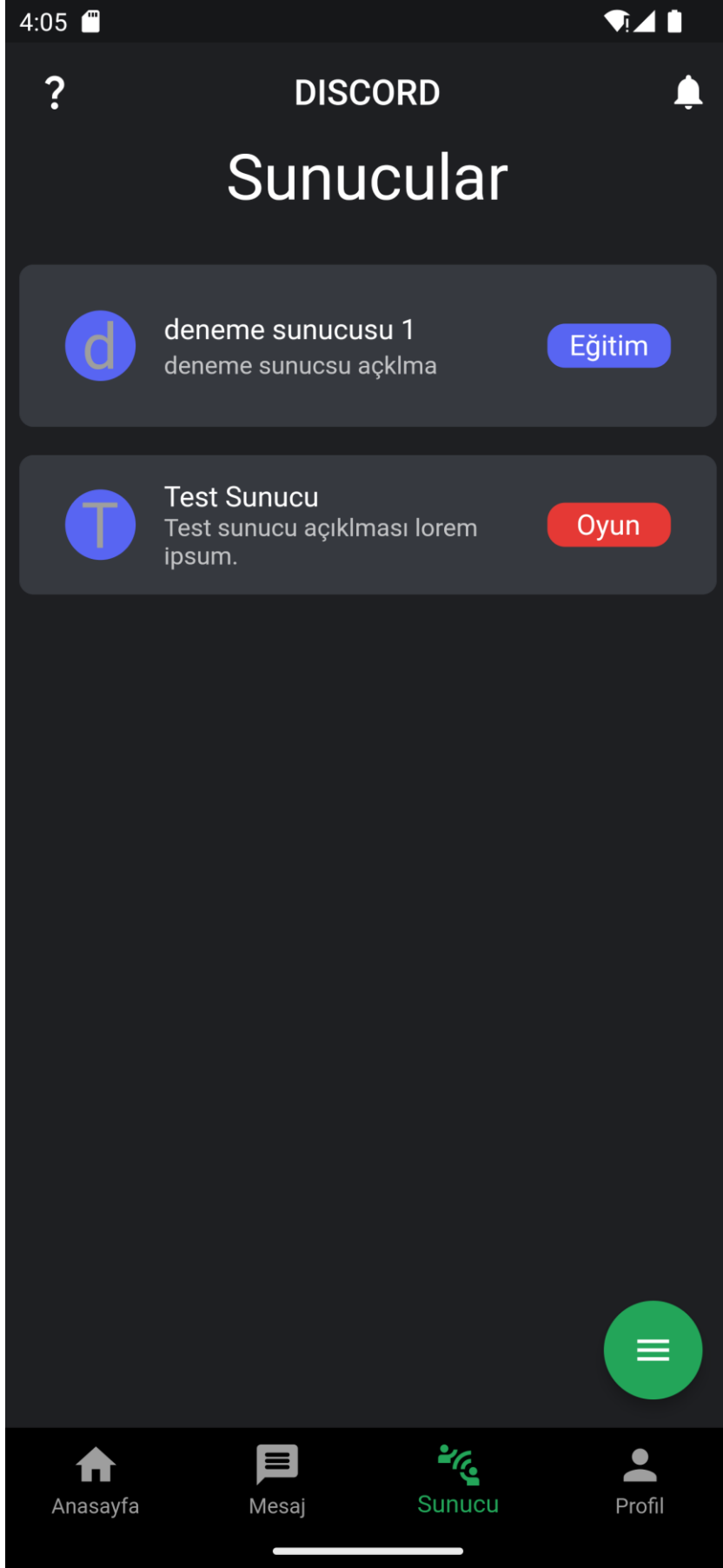
Giriş Yap

[Hemen Kayıt Ol](#)











Sunucu

Kanallar



Ana Kanal
metin



yeni kanal
sesli



4:06

LTE



DISCORD

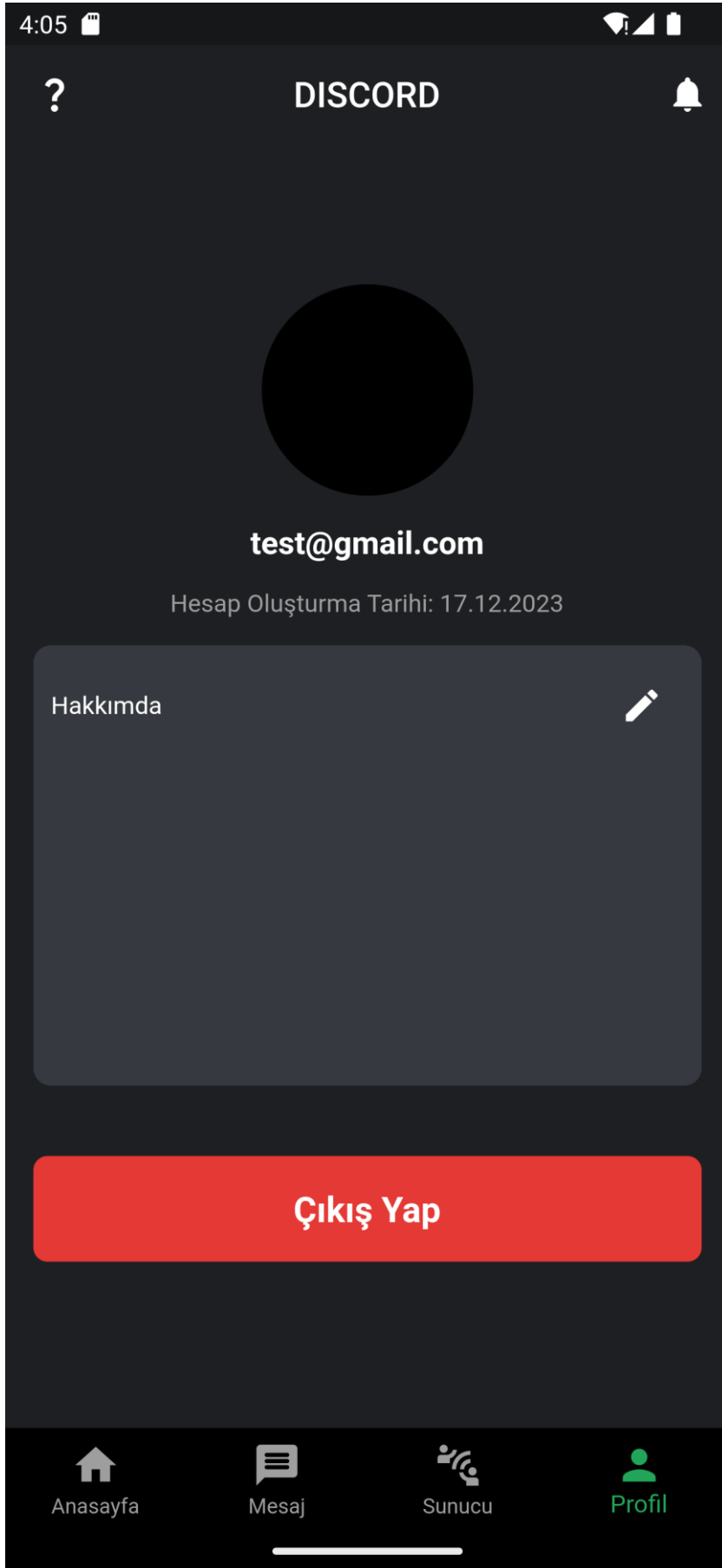


yeni mesaj
test@gmail.com

19:06

yeni mesaj





Users Sign-in method Templates Usage Settings Extensions

SAYFA 22