

Öğretim Görevlisi Enver BAĞCI

IOS ile Mobil Uygulama Geliştirme

- Müzik uygulaması

PROJE RAPORU

**Abdennasser
Romani**

Numara :

- 19290006

Uygulama Adı :

- Dinle Dinle



1. Giriş :



Başlangıç Sayfası :

=> Dinle Dinle internet ile çalışan bir müzik uygulaması'dır . Uygulamayı ilk açtığımızda Başlangıç ekranı'ı açılıyor .

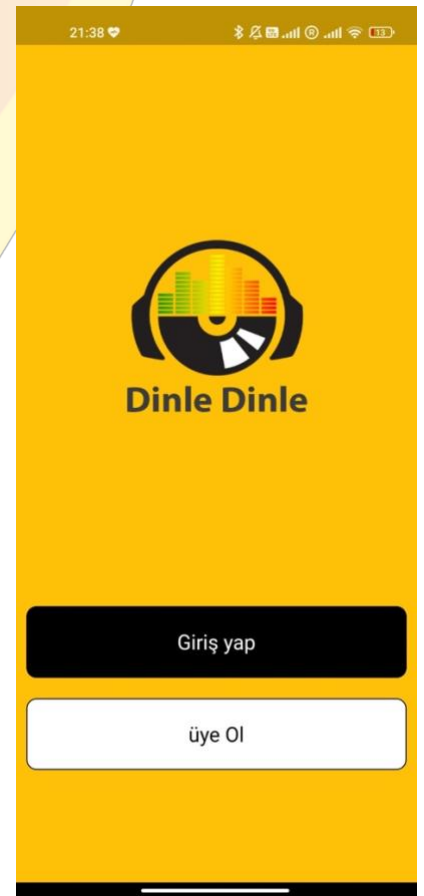
=> Telefonun sistemi uygulamanın problem olmadan çalıştırması için ve sistem hazır olması için Başlangıç ekranı'nda biraz bekliyoruz .

=> Ondan sonra eğer daha önce giriş yapsak direkt AnaSayfa'ya gidiyoruz aksi durumda Hoşgeldin Sayfaya gidiyoruz .

Hoşgeldin Sayfası :

=> Hoşgeldin Sayfasında iki buton var , daha önce hesabı oluşturursak giriş buton'a basarak giriş yapma sayfasına geçiyoruz .

=> Aksi durumda Kayıt olmak için ikinci buton'a basarak Kayıt olma sayfasına geçebiliriz .



17:09

Tekrar hoşgeldiniz!

abdo.romani1996@gmail.com

Parolanızı mı unuttunuz?

Giriş yap

Hesabınız yok mu? Şimdi üye Ol

21:38

Tekrar hoşgeldiniz!

abdo.romani1996@gmail.com

Parolanızı mı unuttunuz?

Geçersiz kullanıcı adı veya şifre. Lütfen tekrar kaydolun veya kullanıcı adı ve şifrenin doğru olduğundan emin olun

Şimdi üye Ol

Giriş Yapma Sayfası :

=> Giriş yapma sayfasında Mail ve Parola kullanarak giriş yapıp AnaSayfaya geçiyoruz .

=> Mailimizde ya da Parolamızda bir problem varsa ya da hesap yoksa bir dialog açılıyor, bir hata var olduğunu uyarmak için .

Kayıt Olma Sayfası :

=> Kayıt olma sayfasında Kullanıcı adı, Mail ve Parola kullanarak hesap oluşturabiliriz .

=> Hesap oluşturduktan sonra giriş yapma sayfasına direkt geçiyoruz .

16:56

Merhaba! Başlamak için kaydolun

Kullanıcı adı

E-posta

Parola

Tekrar Parola

Kayıt ol

Hesabınız var mı? Şimdi giriş yapınız

16:57

Merhaba! Başlamak için kaydolun

abdenasser

abr.romani1996@gmail.com

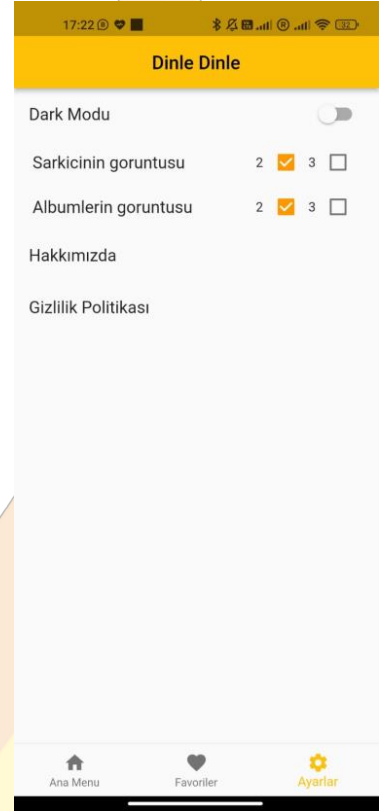
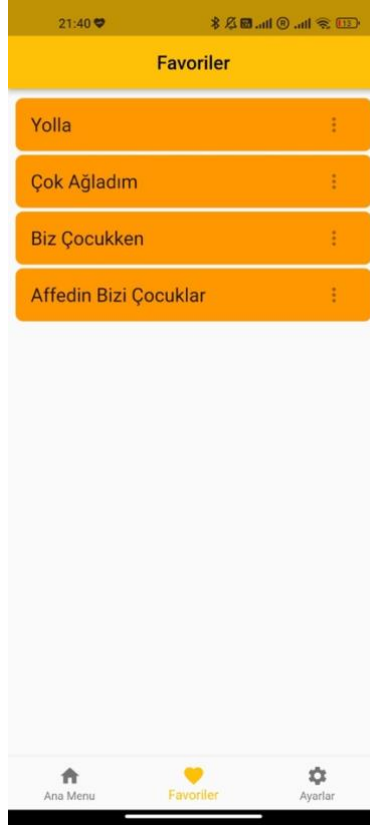
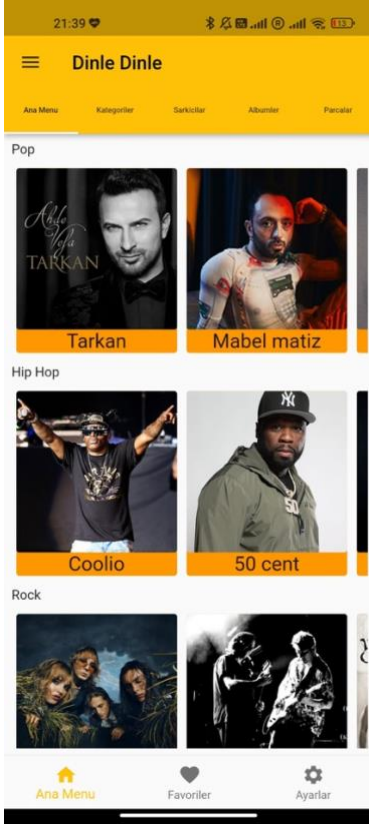
Parola

Tekrar Parola

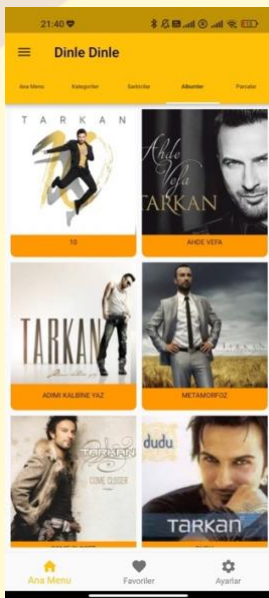
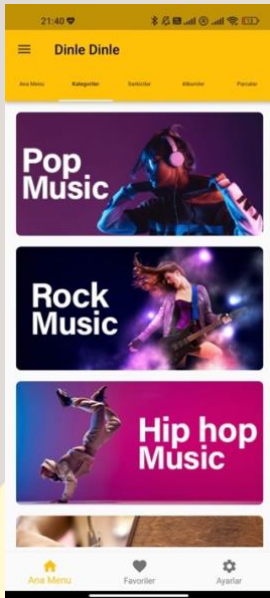
Kayıt ol

AnaSafası :

- => AnaSafasında BottomNavigationBar, Drawer , ve Tap var kullandık .
- => AnaMenu , Favoriler ve Ayarlar BottomNavigationBar olarak koyduk .
- => AnaMenu'da AnaMenu , Kategoriler , Şarkıcılar , Albümler ve Parçalar isimleri içerecek bir TapBar oluşturduk .
- => Favoriler'de ise favori olarak eklediğimiz Parçaları görebiliriz .



- => Kategorileri sayfasına girsek bütün kategorileri bulacağız.
- => Şarkıcılar'ın sayfasına girsek bütün Şarkıcıları bulacağız.
- => Albümler'in sayfasına girsek bütün Albümleri bulacağız.
- => Parçalar'ın sayfasına girsek bütün Parçaları bulacağız.



2. Firebase ile Giriş yapmak :

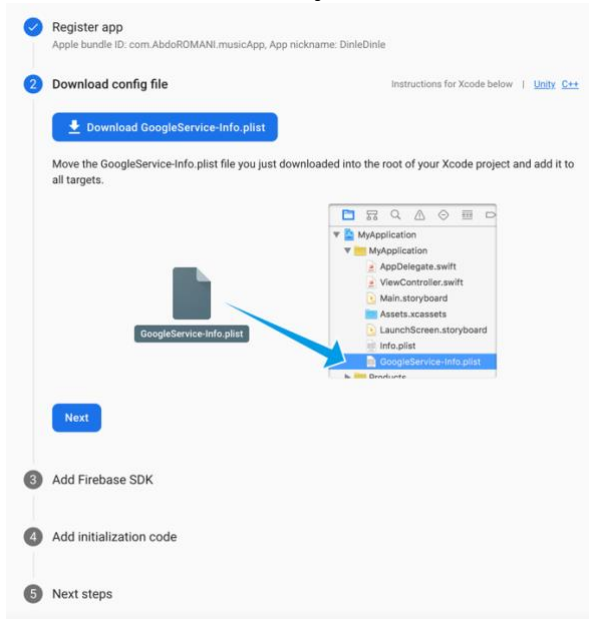
2.1. Android Studio'ye Firebese eklemek :

Android Studio'ya Firebase'i eklemek için Android Studio'daki termineli ve flutter komutları kullanarak ekleyebiliriz .

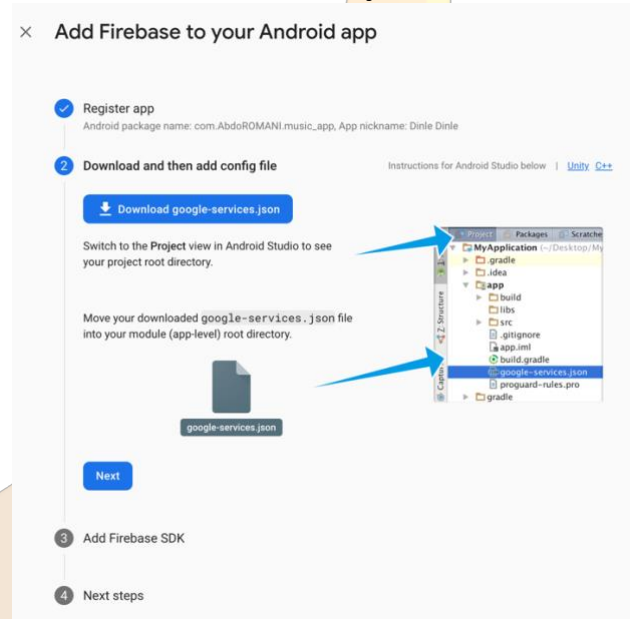
| | |
|---|--|
| <code>flutter pub add firebase_auth</code> | Eklenmeyi kurmak için |
| <code>flutter run</code> | Uygulamamızı yeniden oluşturmamak için |
| <code>import 'package:firebase_auth/firebase_auth.dart';</code> | Eklenmeyi Dart kodumuza aktarırız |

Üsteki adımları uygulayarak Firebase eklenmiş olur , ondan sonra firebase'te hesap oluşturup gerekli dosyaları ekleriz .

iOS için

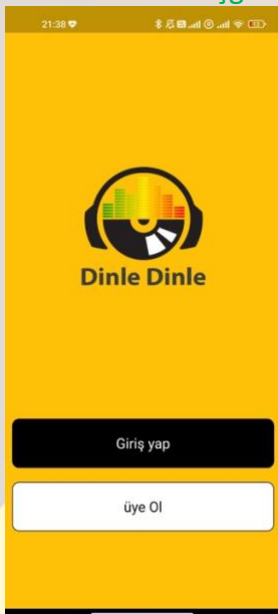


Android için



Firebase'i eklendikten sonra HoşGeldin Sayfasını tasarlamaya geçeriz

2.2. Hoşgeldin Sayfayı tasarlama :



=> Giriş yap ve Kayıt ol butonları oluşturmak için özel bir buton yeni bir Class'ta (CustomizedButton.dart) tanımlayarak yaptık .

=> (welcomescreen.dart)'te butonları tanımladık , Giriş yap butonu basarsak Giriş yapma sayfasına geçeceğiz yoksa Kayıt ol butonu basarsak Kayıt olma sayfasına geçeceğiz .

CustomizedButton.dart

```
import 'package:flutter/material.dart';

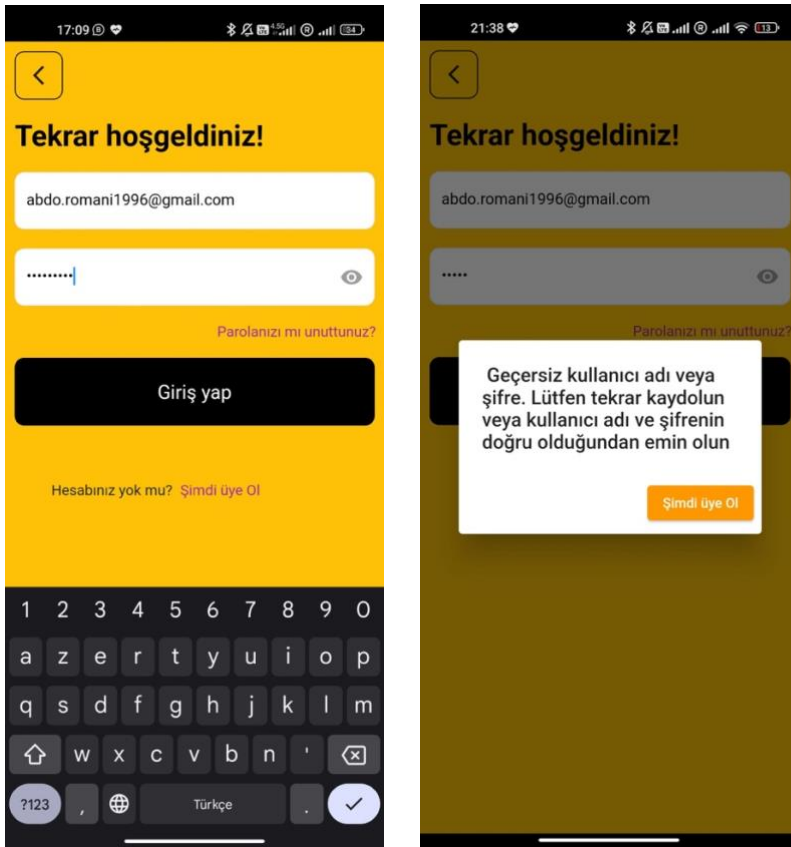
class CustomizedButton extends StatelessWidget {
  final String? buttonText;
  final Color? buttonColor;
  final Color? textColor;
  final VoidCallback? onPressed;
  const CustomizedButton(
    {Key? key,
    this.buttonText,
    this.buttonColor,
    this.onPressed,
    this.textColor})
    : super(key: key);

  @override
  Widget build(BuildContext context) {
    return Padding(
      padding: const EdgeInsets.all(10.0),
      child: InkWell(
        onTap: onPressed,
        child: Container(
          height: 70,
          width: MediaQuery.of(context).size.width,
          decoration: BoxDecoration(
            color: buttonColor,
            border: Border.all(width: 1, color: Colors.black),
            borderRadius: BorderRadius.circular(10)),
          child: Center(
            child: Text(
              buttonText!,
              style: TextStyle(
                color: textColor,
                fontSize: 20,
              ),
            ),
          ),
        ),
      ),
    );
  }
}
```

Butonları tanımlama ve çalıştırma kodları

```
CustomizedButton(
  buttonText: "Giriş yap",
  buttonColor: Colors.black,
  textColor: Colors.white,
  onPressed: () {
    Navigator.push(context,
      MaterialPageRoute(builder: (_) => const LoginScreen()));
  },
),
CustomizedButton(
  buttonText: "Üye Ol",
  buttonColor: Colors.white,
  textColor: Colors.black,
  onPressed: () {
    Navigator.push(context,
      MaterialPageRoute(builder: (_) => const SignUpScreen()));
  },
),
```


2.3. Giriş yapma Sayfayı tasarlama :



=> E-posta ve Parola girmek için özel bir textfield yeni bir Class'ta (customizedtextfield.dart) oluşturarak tanımladık .

=> Mailimizde ya da Parolamızda bir problem varsa ya da hesap yoksa bir dialog açılıyor, bir hata var olduğunu uyarmak için .

Customized_textfield.dart

```
import 'package:flutter/material.dart';

class CustomizedTextfield extends StatelessWidget {
  final TextEditingController myController;
  final String? hintText;
  final bool? isPassword;
  const CustomizedTextfield(
    {Key? key, required this.myController, this.hintText, this.isPassword})
    : super(key: key);

  @override
  Widget build(BuildContext context) {
    return Padding(
      padding: const EdgeInsets.all(10.0),
      child: TextField(
        style: TextStyle(color: Colors.black),
        keyboardType: isPassword!
          ? TextInputType.visiblePassword
          : TextInputType.emailAddress,
        enableSuggestions: isPassword! ? false : true,
        autocorrect: isPassword! ? false : true,
        obscureText: isPassword ?? true,
        controller: myController,
        decoration: InputDecoration(
          suffixIcon: isPassword!
            ? IconButton(
                icon: const Icon(Icons.remove_red_eye, color: Colors.grey),
                onPressed: () {},
              )
        )
      )
    );
  }
}
```

```

        : null,
        enabledBorder: OutlineInputBorder(
          borderSide: const BorderSide(color: Color(0xffE8ECF4), width: 1),
          borderRadius: BorderRadius.circular(10)),
        focusedBorder: OutlineInputBorder(
          borderSide: const BorderSide(color: Color(0xffE8ECF4), width: 1),
          borderRadius: BorderRadius.circular(10)),
        fillColor: Colors.white,
        filled: true,
        hintText: hintText,
        hintStyle: const TextStyle(color: Colors.grey),
        border: OutlineInputBorder(
          borderRadius: BorderRadius.circular(10),
        ),
      ),
    ),
  ),
);
}
}

```

Textfield tanımlama ve çalıştırma kodları :

```

CustomizedTextfield(
  myController: _emailController,
  hintText: "E-postanızı giriniz",
  isPassword: false,
),
CustomizedTextfield(
  myController: _passwordController,
  hintText: "Şifrenizi giriniz",
  isPassword: true,
),

```

Giriş yap butonu basınca E-posta ve parola kontrol etmek için bu kodu kullandık .

```

CustomizedButton(
  buttonText: "Giriş yap",
  buttonColor: Colors.black,
  textColor: Colors.white,
  onPressed: () async {
    try {
      await FirebaseAuthService().login(
        _emailController.text.trim(),
        _passwordController.text.trim());
      if (FirebaseAuth.instance.currentUser != null) {
        if (!mounted) return;
        Navigator.push(
          context,
          MaterialPageRoute(
            builder: (context) => HomePage()),
        );
      }
    } on FirebaseException catch (e) {
      debugPrint("error is ${e.message}");

      showDialog(
        context: context,
        builder: (context) => AlertDialog(
          title: const Text(
            " Geçersiz kullanıcı adı veya şifre. Lütfen tekrar kaydolun  
veya kullanıcı adı ve şifrenin doğru olduğundan emin olun"),
          actions: [
            ElevatedButton(
              child: const Text("Şimdi üye Ol"),
            ),
          ],
        ),
      );
    }
  },
)

```


2.4. Kayıt olma Sayfayı tasarlama :

[illegible]9

Firebese'e verileri eklemek için kullanılan kodları :

```
CustomizedButton(  
  buttonText: "Kayıt ol",  
  buttonColor: Colors.black,  
  textColor: Colors.white,  
  onPressed: () async {  
    try {  
      await FirebaseAuthService().signup(  
        _emailController.text.trim(),  
        _passwordController.text.trim());  
  
      if (!mounted) return;  
  
      Navigator.push(  
        context,  
        MaterialPageRoute(  
          builder: (context) => const LoginScreen()),  
      ) on FirebaseException catch (e) {  
        debugPrint(e.message);  
      }  
  
      // Navigator.push(context,  
      //   MaterialPageRoute(builder: (_) => const LoginScreen()));  
    },  
  ),  
),
```

2.5. Parola unutmama Sayfayı tasarlama :



=> Parolamızı unutmamızın halinde E-postamızı girerek şifremizi yenileyebiliriz .

3. Veri ekleme yöntemi :

(Lists.dart) adlı bir Dökümanda bütün verilerimi ekledim .

3.1. Fotoğrafları eklemek :

(assets) adlı bir Dökümana Kullanılan fotoğrafları eklenim . bu fotoğrafları kullanmak için (pubspec.yaml)'a şöyle gibi eklenilmeli .

```
assets:  
- assets/
```

Sonra kullanmak istenilen yerlerde şöyle çağırabiliriz .

```
image: AssetImage("assets/drawer.png"),
```

3.2. Kategorileri eklemek :

Önce bir class oluşturuldu

```
class kategorilerL{  
    int kid;//Kategorinin id  
    String isim;//Kategorinin ismi  
    String resim;//Kategorinin fotografi  
    List<sarkicilarL> sarkici;//Kategorinin sarkicilari  
    kategorilerL({required this.kid,required this.isim,required this.sarkici,required  
this.resim}); //Veri Eklemek icin  
}
```

Kategorilerin listesini oluşturmak ve doldurmak için :

```
List<kategorilerL> Kategoriler = [  
    kategorilerL(kid: 1, isim: 'Pop', sarkici: Sarkicilar.sublist(0,9), resim:  
'assets/pop.png'),  
    kategorilerL(kid: 2, isim: 'Hip Hop', sarkici: Sarkicilar.sublist(9,15), resim:  
'assets/rock.png'),  
    kategorilerL(kid: 3, isim: 'Rock', sarkici: Sarkicilar.sublist(15,20), resim:  
'assets/hiphop.png'),  
    kategorilerL(kid: 4, isim: 'Arabesk', sarkici: Sarkicilar.sublist(20,25), resim:  
'assets/arabesk.png'),  
];
```

3.3. Şarkıcıları eklemek :

Önce bir class oluşturuldu

```
class sarkicilarL{  
    int sid;  
    String isim;  
    String resim;  
    int kid;  
    List<albumlerL> album;  
    sarkicilarL({required this.sid, required this.isim, required this.resim, required  
this.kid, required this.album});  
}
```

Şarkıcıların listesini oluşturmak ve doldurmak için :

```
//Sarkici listesi  
List<sarkicilarL> Sarkicilar = [  
    //Pop  
    sarkicilarL(sid: 1, isim: 'Tarkan', resim: 'assets/tarkan.png', kid: 1, album:  
Albumler.sublist(0,6)),  
    sarkicilarL(sid: 2, isim: 'Mabel matiz', resim: 'assets/Mabelmatiz.png', kid: 1,  
album: Albumler.sublist(6,11)),  
    sarkicilarL(sid: 3, isim: 'Zeynep bastık', resim: 'assets/Zeynepbastik.png', kid:  
1, album: Albumler.sublist(11,15)),  
    sarkicilarL(sid: 4, isim: 'Kenan Doğulu', resim: 'assets/KenanDogulu.png', kid: 1,  
album: Albumler.sublist(15,19)),  
    //Hip Hop  
    sarkicilarL(sid: 10, isim: 'Coolio', resim: 'assets/Coolio.png', kid: 2, album:
```

```

Albumler sublist(40,42)),
    sarkicilarL(sid: 11, isim: '50 cent', resim: 'assets/fiftycent.png', kid: 2, album:
Albumler sublist(42,44)),
    //Rock
    sarkicilarL(sid: 16, isim: 'Maneskin', resim: 'assets/Maneskin.png', kid: 3, album:
Albumler sublist(57,59)),
    sarkicilarL(sid: 17, isim: 'Duman', resim: 'assets/Duman.png', kid: 3, album:
Albumler sublist(59,62)),
    //Arabesk
    sarkicilarL(sid: 21, isim: 'Müslüm gürses', resim: 'assets/Mgurses.png', kid: 4,
album: Albumler sublist(70,72)),
    sarkicilarL(sid: 22, isim: 'İbrahim tatlıses', resim: 'assets/Itatlıses.png', kid:
4, album: Albumler sublist(72,75)),
];

```

3.4. Albümleri eklemek :

Önce bir class oluşturuldu

```

class albumlerL{
    int aid;
    String isim;
    String resim;
    List<sarkilarL> sarki;
    int sid;
    albumlerL({required this.aid, required this.isim, required this.resim, required
this.sid, required this.sarki});
}
.kid, required this.album});
}

```

Albümleri listesini oluşturmak ve doldurmak için :

```

//Album listesi
List<albumlerL> Albumler = [
    //Tarkan
    albumlerL(aid: 1, isim: '10', resim: 'https://mp3semticdn.com/mp3_images/Tarkan-10-
2017.jpg', sid: 1, sarki: Sarkilar sublist(0,14)),
    albumlerL(aid: 2, isim: 'AHDE VEFA', resim:
'https://mp3semticdn.com/mp3_images/Tarkan-Ahde-Vefa-2016.jpg', sid: 1, sarki:
Sarkilar sublist(14,24)),
    albumlerL(aid: 3, isim: 'ADIMI KALBİNE YAZ', resim:
'https://mp3semticdn.com/mp3_images/Tarkan-Adimi-Kalbine-Yaz-2010.jpg', sid: 1,
sarki: Sarkilar sublist(24,32)),
    albumlerL(aid: 4, isim: 'METAMORFOZ', resim:
'https://mp3semticdn.com/mp3_images/Tarkan-Metamorfoz-2007-5.jpg', sid: 1, sarki:
Sarkilar sublist(32,35)),
    albumlerL(aid: 5, isim: 'COME CLOSER', resim:
'https://mp3semticdn.com/mp3_images/Tarkan-Come-Closer-2006-6.jpg', sid: 1, sarki:
Sarkilar sublist(35,36)),
    albumlerL(aid: 6, isim: 'DUDU', resim: 'https://mp3semticdn.com/mp3_images/Tarkan-
Dudu-2013.jpg', sid: 1, sarki: Sarkilar sublist(36,47)),
    //Mabel matiz
    albumlerL(aid: 7, isim: 'Maya', resim:
'https://i.scdn.co/image/ab67616d00001e02673669edc4c83a8ed2e2b20d', sid: 2, sarki:
Sarkilar sublist(47,56)),
    albumlerL(aid: 8, isim: 'Gök Nerede', resim:
'https://i.scdn.co/image/ab67616d00001e027b52fb76888fc05880db10c8', sid: 2, sarki:
Sarkilar sublist(56,66)),
    albumlerL(aid: 9, isim: 'Hanfendi', resim:

```

```
'https://i.scdn.co/image/ab67616d00001e0204a5819b31141c59915fe66b', sid: 2, sarki:
Sarkilar.sublist(66,69)),
    albumlerL(aid: 10, isim: 'Yaşım Çocuk', resim:
'https://i.scdn.co/image/ab67616d00001e0258d2bf7ae892171800d3b51c', sid: 2, sarki:
Sarkilar.sublist(69,78)),
    albumlerL(aid: 11, isim: 'Mabel Matiz', resim:
'https://i.scdn.co/image/ab67616d00001e024e6fafdc714c44e2e6c5ef92', sid: 2, sarki:
Sarkilar.sublist(78,87)),
    //Zeynep bastık
.
.
.
```

3.5. Parçaları eklemek :

Önce bir class oluşturuldu

```
class sarkilarL{
    int id;
    String isim;
    String url;
    int aid;
    sarkilarL({required this.id, required this.isim, required this.url, required
this.aid});
}
```

Albümleri listesini oluşturmak ve doldurmak için :

```
//sarki listesi
List<sarkilarL> Sarkilar = [
    //Tarkan 10
    sarkilarL(id: 1, isim: 'Yolla', url: 'https://mp3semticdn.com/mp3_files/Tarkan/10-
2017/Tarkan-Yolla.mp3', aid: 1),
    sarkilarL(id: 2, isim: 'Çay Simit', url:
'https://mp3semticdn.com/mp3_files/Tarkan/10-2017/Tarkan-Cay-Simit.mp3', aid: 1),
    sarkilarL(id: 3, isim: 'Beni Çok Sev', url:
'https://mp3semticdn.com/mp3_files/Tarkan/10-2017/Tarkan-Beni-Cok-Sev.mp3', aid: 1),
    sarkilarL(id: 4, isim: 'Ben Senin', url:
'https://mp3semticdn.com/mp3_files/Tarkan/10-2017/Tarkan-Ben-Senin.mp3', aid: 1),
    sarkilarL(id: 5, isim: 'Çok Ağladım', url:
'https://mp3semticdn.com/mp3_files/Tarkan/10-2017/Tarkan-Cok-Agladim.mp3', aid: 1),
    sarkilarL(id: 6, isim: 'Her Şey Fani', url:
'https://mp3semticdn.com/mp3_files/Tarkan/10-2017/Tarkan-Her-Sey-Fani.mp3', aid: 1),
    sarkilarL(id: 7, isim: 'Kedi Gibi', url:
'https://mp3semticdn.com/mp3_files/Tarkan/10-2017/Tarkan-Kedi-Gibi.mp3', aid: 1),
    sarkilarL(id: 8, isim: 'Hodri Meydan', url:
'https://mp3semticdn.com/mp3_files/Tarkan/10-2017/Tarkan-Hodri-Meydan.mp3', aid: 1),
    sarkilarL(id: 9, isim: 'Biz Çocukken', url:
'https://mp3semticdn.com/mp3_files/Tarkan/10-2017/Tarkan-Biz-Cocukken.mp3', aid: 1),
    sarkilarL(id: 10, isim: 'Bal Küpü', url:
'https://mp3semticdn.com/mp3_files/Tarkan/10-2017/Tarkan-Bal-Kupu.mp3', aid: 1),
    sarkilarL(id: 11, isim: 'Acımasız', url:
'https://mp3semticdn.com/mp3_files/Tarkan/10-2017/Tarkan-Acimasiz.mp3', aid: 1),
    sarkilarL(id: 12, isim: 'Sevdam Tek Nefes', url:
'https://mp3semticdn.com/mp3_files/Tarkan/10-2017/Tarkan-Sevdam-Tek-Nefes.mp3', aid:
1),
    sarkilarL(id: 13, isim: 'O Sevişmeler', url:
'https://mp3semticdn.com/mp3_files/Tarkan/10-2017/Tarkan-O-Sevismeler.mp3', aid: 1),
    sarkilarL(id: 14, isim: 'Affedin Bizi Çocuklar', url:
'https://mp3semticdn.com/mp3_files/Tarkan/10-2017/Tarkan-Affedin-Bizi-Cocuklar.mp3',
aid: 1),
    //Tarkan AHDE VEFA
    sarkilarL(id: 15, isim: 'Akşam Oldu Hüzünlendim Ben Yine', url:
```

```

'https://mp3semticdn.com/mp3_files/Tarkan/Ahde-Vefa-2016/Tarkan-Aksam-Oldu-
Huzunlendim-Ben-Yine.mp3', aid: 2),
  sarkilarL(id: 16, isim: 'Aşk Bu Değil Mi', url:
'https://mp3semticdn.com/mp3_files/Tarkan/Ahde-Vefa-2016/Tarkan-Ask-Bu-Degil-Mi.mp3',
aid: 2),
  sarkilarL(id: 17, isim: 'Enginde Yavaş Yavaş', url:
'https://mp3semticdn.com/mp3_files/Tarkan/Ahde-Vefa-2016/Tarkan-Enginde-Yavas-
Yavas.mp3', aid: 2),
  sarkilarL(id: 18, isim: 'Islak Daha Islak Öp Beni', url:
'https://mp3semticdn.com/mp3_files/Tarkan/Ahde-Vefa-2016/Tarkan-Islak-Daha-Islak-Op-
Beni.mp3', aid: 2),
  sarkilarL(id: 19, isim: 'Kadehinde Zehir Olsa', url:
'https://mp3semticdn.com/mp3_files/Tarkan/Ahde-Vefa-2016/Tarkan-Kadehinde-Zehir-
Olsa.mp3', aid: 2),
  sarkilarL(id: 20, isim: 'Kara Bulutları Kaldır Aradan', url:
'https://mp3semticdn.com/mp3_files/Tarkan/Ahde-Vefa-2016/Tarkan-Kara-Bulutlari-
Kaldir-Aradan.mp3 ', aid: 2),
  sarkilarL(id: 21, isim: 'Nasıl Geçti Habersiz', url:
'https://mp3semticdn.com/mp3_files/Tarkan/Ahde-Vefa-2016/Tarkan-Nasil-Gecti-
Habersiz.mp3', aid: 2),
  sarkilarL(id: 22, isim: 'Olmaz İlaç Sine-i Sad Pareme', url:
'https://mp3semticdn.com/mp3_files/Tarkan/Ahde-Vefa-2016/Tarkan-Olmaz-Ilac-Sine-i-
Sad-Pareme.mp3', aid: 2),
  sarkilarL(id: 23, isim: 'Rindlerin Akşamı', url:
'https://mp3semticdn.com/mp3_files/Tarkan/Ahde-Vefa-2016/Tarkan-Rindlerin-
Aksami.mp3', aid: 2),
  sarkilarL(id: 24, isim: 'Sevmekten Kim Usanır', url:
'https://mp3semticdn.com/mp3_files/Tarkan/Ahde-Vefa-2016/Tarkan-Sevmekten-Kim-
Usanir.mp3', aid: 2),
  //Adımı Kalbine Yaz
.
.
.

```

4. AnaMenu tasarlama:

4.1. BottomNavigationBar :

BottomNavigationBar oluşturmak için (main.dart)'ta aşağıdaki kodu kullanıldı :

```

bottomNavigationBar: BottomNavigationBar(
  onTap: (index) {
    setState(() {
      myIndex = index;
    });
  },
  currentIndex: myIndex,
  items: const [
    BottomNavigationBarItem(
      icon: Icon(Icons.home),
      label: "Ana Menu",
    ),
    BottomNavigationBarItem(
      icon: Icon(Icons.favorite),
      label: "Favoriler",
    ),
    BottomNavigationBarItem(
      icon: Icon(Icons.settings),
      label: "Ayarlar",
    ),
  ],
),

```


4.2. Drawer :

Drawer oluşturmak için (AnaMenu.dart)'teki AppBar içinde aşağıdaki kodu kullanıldı :

```
drawer: Drawer(  
  child: ListView(  
    padding: EdgeInsets.zero,  
    children: <Widget>[  
      DrawerHeader(  
        decoration: const BoxDecoration(  
          image: DecorationImage(  
            image: AssetImage("assets/drawer.png"),  
            fit: BoxFit.cover,  
          ),  
        ),  
        child: Container(  
          alignment: Alignment.bottomLeft, child: Text(userName!)),  
      ),  
      SwitchListTile(  
        //activeTrackColor: Colors.orange,  
        activeColor: Colors.orange,  
        title: const Text('Dark Modu'),  
        value: dark,  
        onChanged: (state) {  
          setState(() {  
            dark = state;  
          });  
          setTheme();  
          Navigator.push(  
            context,  
            MaterialPageRoute(builder: (context) => HomePage()),  
          );  
        },  
      ),  
      ListTile(  
        leading: const Icon(Icons.info),  
        title: const Text('Hakkımızda'),  
        onTap: () {  
          Navigator.push(  
            context,  
            MaterialPageRoute(builder: (context) => aboutUs()),  
          );  
        },  
      ),  
      ListTile(  
        leading: const Icon(Icons.contact_phone),  
        title: const Text('Bize Ulaşın'),  
        onTap: () {  
          Navigator.push(  
            context,  
            MaterialPageRoute(builder: (context) => contactUs()),  
          );  
        },  
      ),  
      ListTile(  
        leading: const Icon(Icons.privacy_tip),  
        title: const Text('Gizlilik Politikası'),  
        onTap: () {  
          Navigator.push(  
            context,  
            MaterialPageRoute(builder: (context) => privacyPolicy()),  
          );  
        },  
      ),  
    ],  
  ),  
),
```

```

    ),
    ListTile(
      leading: const Icon(Icons.outbond),
      title: const Text('Çıkış yap'),
      onTap: () async {
        await FirebaseAuth.instance.signOut();
        if (!mounted) return;
        Navigator.push(context,
          MaterialPageRoute(builder: (context) {
            return const WelcomeScreen();
          }));
      },
    ),
  ],
),
), //AppBar

```

4.3. TabBar :

```

bottom: const TabBar(
  tabs: <Widget>[
    Tab(
      icon: Text(
        "Ana Menu",
        style: TextStyle(fontSize: 11),
      ),
    ),
    Tab(
      icon: Text(
        "Kategoriler",
        style: TextStyle(fontSize: 10),
      ),
    ),
    Tab(
      icon: Text(
        "Sarkicilar",
        style: TextStyle(fontSize: 11),
      ),
    ),
    Tab(
      icon: Text(
        "Albumler",
        style: TextStyle(fontSize: 11),
      ),
    ),
    Tab(
      icon: Text(
        "Parcalar",
        style: TextStyle(fontSize: 11),
      ),
    ),
  ], //tabs
),

```

4.4. AnaMenu :



=> AnaMenu'yu böyle göstermek için iki listView kullanıldı , 1. olan kategorileri göstermek için 2. olan 1.nin içinde eklenildi şarkıcıları göstermek için

=> itemleri göstermek için Farklı Grid'leri oluşturuldu (Gridler.dart)'ta .

Bir Grid'in örneği :

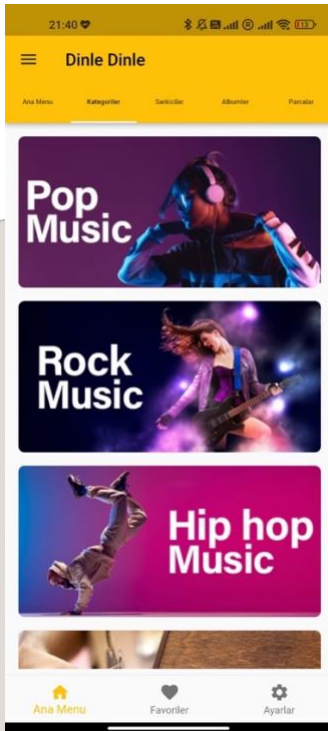
```
class Grid2 extends StatelessWidget {
  final String isim, resim;

  Grid2({required this.isim, required this.resim});
  @override
  Widget build(BuildContext context) {
    return FittedBox(
      child: Container(
        margin: const EdgeInsets.all(5),
        decoration: BoxDecoration(
          borderRadius: BorderRadius.circular(
            8.0,
          ),
          color: Colors.orange,
        ),
        child: Column(
          children: [
            ClipRRect(
              borderRadius: BorderRadius.circular(
                8.0,
              ),
              child: Image.asset(resim)),
            Text(
              isim,
              style: const TextStyle(fontSize: 50),
            ),
          ],
        ),
      ),
    );
  }
}
```

ListView örneği :

```
ListView.builder(  
  shrinkWrap: true,  
  physics: const ScrollPhysics(),  
  scrollDirection: Axis.horizontal,  
  itemCount: Kategoriler[i].sarkici.length,  
  itemBuilder: (context, index) {  
    return GestureDetector(  
      child: Container(  
        width: 180.0,  
        height: 180.0,  
        child: Grid2(  
          isim: Kategoriler[i].sarkici[index].isim,  
          resim: Kategoriler[i].sarkici[index].resim,  
        ),  
      ),  
      onTap: () {  
        Navigator.of(context, rootNavigator: true).push(  
          MaterialPageRoute(builder: (context) =>  
            Sarkiciler(  
              bartext:  
                Kategoriler[i].sarkici[index].isim,  
              mualbum: Kategoriler[i]  
                .sarkici[index]  
                .album),  
            ),  
        );  
      },  
    );  
  },  
);
```

4.5. Kategoriler :



=> Kategorileri böyle göstermek için bir listView oluşturuldu .

=> itemleri göstermek için Farklı Grid'leri oluşturuldu (Gridler.dart)'ta .

Bir Grid'in örneği :

```

class Grid1 extends StatelessWidget {
  final String isim, resim;

  Grid1({required this.isim, required this.resim});
  @override
  Widget build(BuildContext context) {
    return Padding(
      padding: const EdgeInsets.all(8.0),
      child: FittedBox(
        child: Container(
          decoration: BoxDecoration(
            borderRadius: BorderRadius.circular(
              8.0,
            ),
          ),
          child: Column(
            children: [
              ClipRRect(
                borderRadius: BorderRadius.circular(
                  8.0,
                ),
                child: Image.asset(
                  resim,
                ),
              ),
              //Text(isim, style: const TextStyle(fontSize: 40)),
            ],
          ),
        ),
      ),
    );
  }
}

```

ListView örneği :

```

ListView.builder(
  itemCount: Kategoriler.length,
  itemBuilder: (context, index) {
    return GestureDetector(
      child: Container(
        child: Grid1(
          isim: Kategoriler[index].isim,
          resim: Kategoriler[index].resim,
        ),
      ),
      onTap: () {
        Navigator.of(context, rootNavigator: true).push(
          MaterialPageRoute(
            builder: (context) => Katego(
              bartext: Kategoriler[index].isim,
              musarki: Kategoriler[index].sarkici),
          );
      },
    );
  },
);

```

4.6. Şarkıcılar :



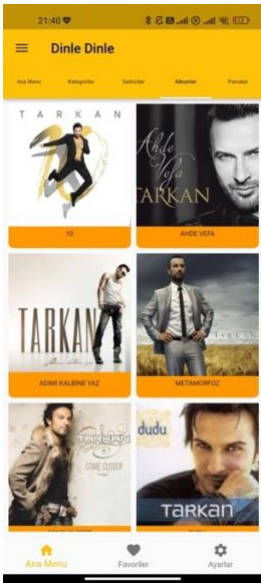
=> Şarkıcılar böyle göstermek için bir listView oluşturuldu .

=> itemleri göstermek için Farklı Grid'leri oluşturuldu (Gridler.dart)'ta .

ListView örneği :

```
GridView.builder(  
  gridDelegate: const SliverGridDelegateWithFixedCrossAxisCount(  
    crossAxisCount: 2,  
    crossAxisSpacing: 8.0,  
    mainAxisSpacing: 8.0,  
    childAspectRatio: 0.85,  
  ),  
  itemCount: Sarkicilar.length,  
  itemBuilder: (context, index) {  
    return GestureDetector(  
      child: Container(  
        child: Grid2(  
          isim: Sarkicilar[index].isim,  
          resim: Sarkicilar[index].resim,  
        ),  
      ),  
      onTap: () {  
        Navigator.of(context, rootNavigator: true).push(  
          MaterialPageRoute(  
            builder: (context) => Sarkiciler(  
              bartext: Sarkicilar[index].isim,  
              mualbum: Sarkicilar[index].album),  
            );  
          });  
    },  
  ));
```


4.7. Albümler :



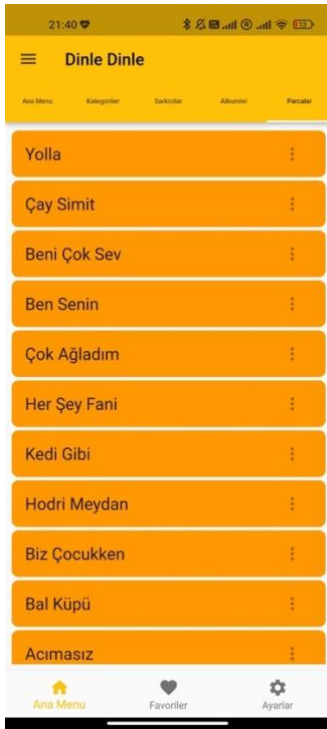
=> Albümler böyle göstermek için bir listView oluşturuldu .

=> itemleri göstermek için Farklı Grid'leri oluşturuldu (Gridler.dart)'ta .

ListView örneği :

```
GridView.builder(  
  gridDelegate: const SliverGridDelegateWithFixedCrossAxisCount(  
    crossAxisCount: 2,  
    crossAxisSpacing: 8.0,  
    mainAxisSpacing: 8.0,  
    childAspectRatio: 0.85,  
  ),  
  itemCount: Albumler.length,  
  itemBuilder: (context, index) {  
    return GestureDetector(  
      child: Container(  
        child: Grid3(  
          isim: Albumler[index].isim,  
          resim: Albumler[index].resim,  
        ),  
      onTap: () {  
        Navigator.of(context, rootNavigator: true).push(  
          MaterialPageRoute(  
            builder: (context) => Albums(  
              bartext: Albumler[index].isim,  
              musarki: Albumler[index].sarki),  
          );  
        });  
      },  
    );  
  },  
);
```

4.8. Parçalar :



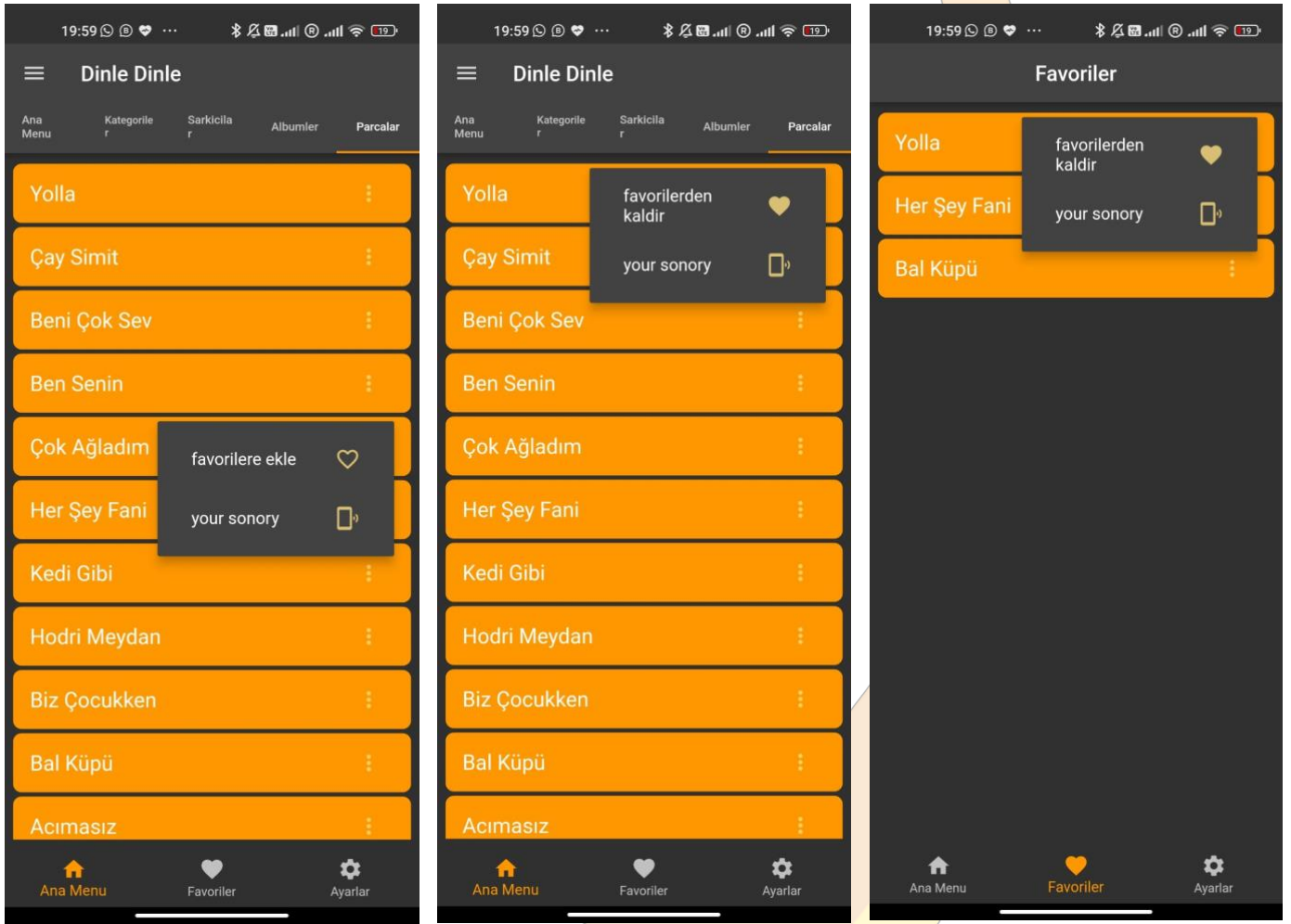
=> Parçalar böyle göstermek için bir listView oluşturuldu .

=> itemleri göstermek için Farklı Grid'leri oluşturuldu (Gridler.dart)'ta .

ListView örneği :

```
ListView.builder(  
  itemCount: Sarkilar.length,  
  itemBuilder: (context, index) {  
    return GestureDetector(  
      child: Grid5(  
        isim: Sarkilar[index].isim,  
        index: Sarkilar[index].id - 1,  
        id: Sarkilar[index].id,  
        url: Sarkilar[index].url),  
      onTap: () {  
  
        Navigator.of(context, rootNavigator: true).push(  
          MaterialPageRoute(  
            builder: (context) =>  
              MediaPlayer(musarki: Sarkilar, index: index)),  
          );  
      });  
    },  
  ),
```

5. Favoriler :



5.1. Favorileri eklemek :

Favorileri eklemek için shared preferences Kullanıldı .

İlk başta (pubspec.yaml)'da kütüphanesini eklemeliyiz, böyle yapabiliriz :

```
shared_preferences: ^2.0.15
```

favorileri eklemek için aşağıdaki kodu kullanıldı (true koyarak) :

```
Future<void> setFavorite(int index) async {
  final prefs = await SharedPreferences.getInstance();
  setState(() {
    if (isFavo[index] == false) {
      prefs.setBool(index.toString(), false);
    } else {
      prefs.setBool(index.toString(), true); //ekle
    }
  });
  Favoriler = [];
  for (int i = 0; i < Sarkilar.length; i++) {
    isFavo.add(prefs.getBool(i.toString()) ?? false);
    if (isFavo[i]) {
      Favoriler.add(sarkilarL(
        id: Sarkilar[i].id,
```

```

        isim: Sarkilar[i].isim,
        url: Sarkilar[i].url,
        aid: Sarkilar[i].aid));
    }
}
});
}

```

5.2. Favorileri eklemek :

favorileri silmek için aşağıdaki kodu kullanıldı (false koyarak) :

```

Future<void> setFavorite(int index) async {
    final prefs = await SharedPreferences.getInstance();
    setState(() {
        if (isFavo[index] == false) {
            prefs.setBool(index.toString(), false); //sil
        } else {
            prefs.setBool(index.toString(), true);
        }
        Favoriler = [];
        for (int i = 0; i < Sarkilar.length; i++) {
            isFavo.add(prefs.getBool(i.toString()) ?? false);
            if (isFavo[i]) {
                Favoriler.add(sarkilarL(
                    id: Sarkilar[i].id,
                    isim: Sarkilar[i].isim,
                    url: Sarkilar[i].url,
                    aid: Sarkilar[i].aid));
            }
        }
    });
}

```

5.3. Favorileri eklemek :

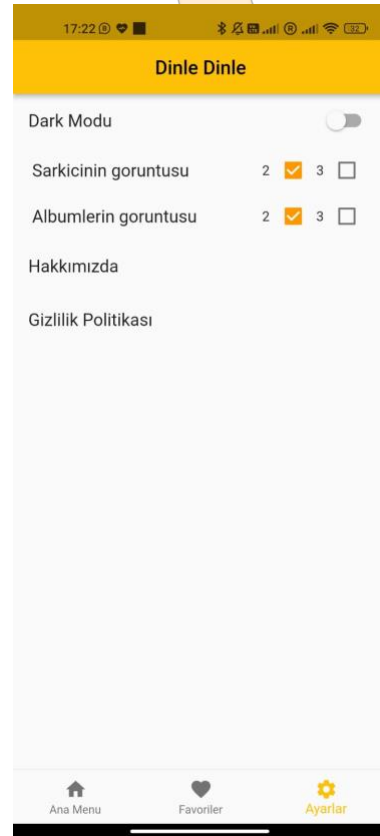
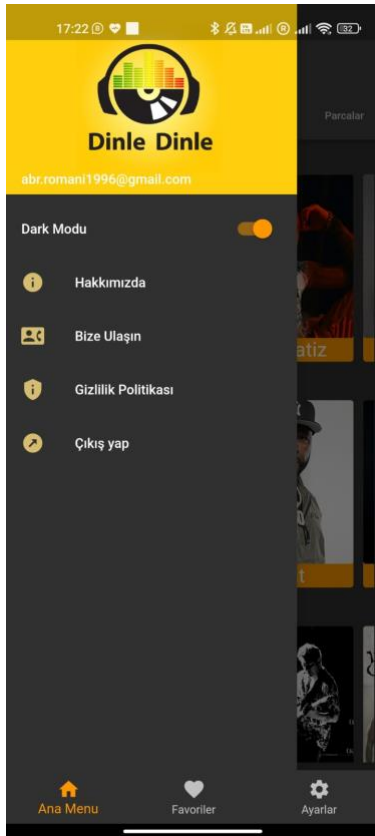
Favorilerin listesi doldurmak için :

```

final prefs = await SharedPreferences.getInstance();
setState(() {
    for (int i = 0; i < Sarkilar.length; i++) {
        isFavo.add(prefs.getBool(i.toString()) ?? false);
        if (isFavo[i]) {
            Favoriler.add(sarkilarL(
                id: Sarkilar[i].id,
                isim: Sarkilar[i].isim,
                url: Sarkilar[i].url,
                aid: Sarkilar[i].aid));
        }
    }
});

```

6. Ayarlar :



6.1. Dark modu (Shared preferences kullanıldı) :

Dark Light modun kodları :

```
ThemeData darkTheme = ThemeData(  
  // scaffoldBackgroundColor: Colors.purple.shade900,  
  //colorScheme: const ColorScheme.dark(),  
  accentColor: Colors.orange,  
  brightness: Brightness.dark,  
  //primaryColor: Colors.amber,  
  primarySwatch: Colors.amber,  
  iconTheme: IconThemeData(color: Colors.amber.shade200,opacity: 0.8),  
);  
  
ThemeData lightTheme = ThemeData(  
  // scaffoldBackgroundColor: Colors.purple.shade50,  
  //colorScheme: const ColorScheme.light(),  
  //accentColor: Colors.pink,  
  brightness: Brightness.light,  
  //primaryColor: Colors.amber,  
  primarySwatch: Colors.amber,  
  iconTheme: IconThemeData(color: Colors.amber.shade200,opacity: 0.8),  
);
```

Dark Modu aktif etmek için (Shared preferences kullanıldı):

```
SwitchListTile(  
  //activeTrackColor: Colors.orange,  
  activeColor: Colors.orange,  
  title: const Text('Dark Modu'),
```

```

value: dark,
onChanged: (state) {
  setState(() {
    dark = state;
  });
  setTheme();
  Navigator.push(
    context,
    MaterialPageRoute(builder: (context) => HomePage()),
  );
},
),

```

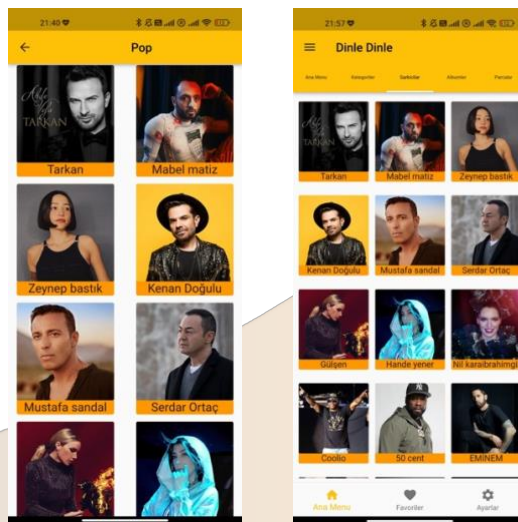
Uygulamaya ilk girdiğimizde var olduğunu theme modunu nasıl bilinir ?

```

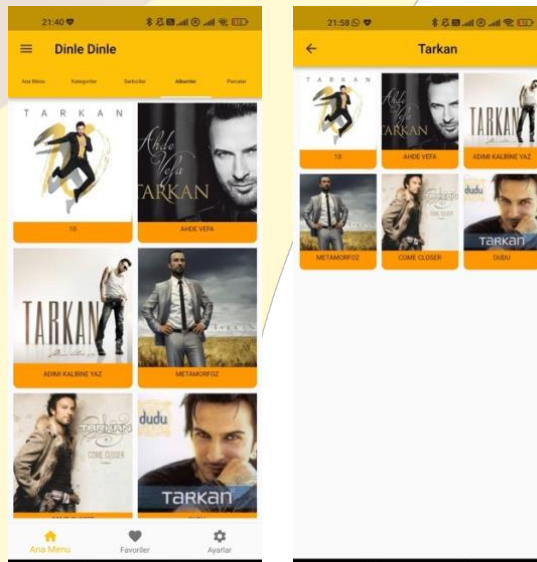
Future<void> getTheme() async {
  final prefs = await SharedPreferences.getInstance();
  setState(() {
    dark = (prefs.getBool('dark') ?? false);
  });
}

```

6.2. Şarkıcının görüntüsü :



6.3. Albümlerin görüntüsü :



Görüntüsü kodlari :

2 ise :

```
gridDelegate: const SliverGridDelegateWithFixedCrossAxisCount(  
  crossAxisCount: 2,  
  crossAxisSpacing: 8.0,  
  mainAxisSpacing: 8.0,  
  childAspectRatio: 0.85,  
),
```

3 ise :

```
gridDelegate: const SliverGridDelegateWithFixedCrossAxisCount(  
  crossAxisCount: 3,  
  crossAxisSpacing: 8.0,  
  mainAxisSpacing: 8.0,  
  childAspectRatio: 0.85,  
),
```

7. Sayfalar :

7.1. Kategoriler Sayfası :



=> Her Hangi bir Kategoriyeye bastığımızda , basıldığı kategoriyeye ait şarkıcıları gözükecek .

=> Class'ın içinde Gridiew Kullanıldı >

“Katego.dart”tan şarkıcıları göstermek için :

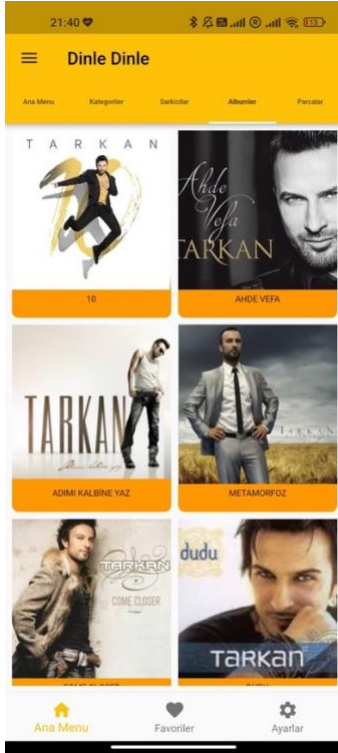
```
@override  
Widget build(BuildContext context) {  
  return MaterialApp(  
    theme: dark ? darkTheme : lightTheme,  
    home: Scaffold(  
      appBar: AppBar(  
        title: Center(child: Text(widget.bartext)),  
        leading: IconButton(  
          icon: const Icon(Icons.arrow_back),  
          onPressed: () {  
            Navigator.of(context).pop(true);  
          }  
        ),  
      ),  
    ),  
  );  
}
```

```

    },
  ),
),
body: (sarkiciii == 2) ? GridK22(widget.musarki) : GridK33(widget.musarki),
),
);
}

```

7.2. Şarkıcılar Sayfası :



=> Her Hangi bir Şarkıcıya bastığımızda , basıldığı Şarkıcıya ait albümleri gözükecek .

=> Class'ın içinde Gridiew Kullanıldı >

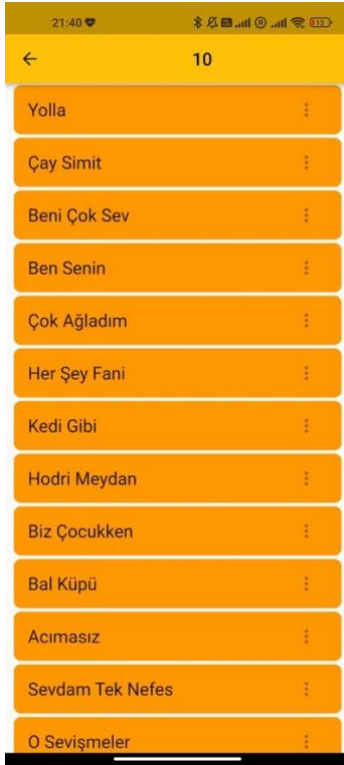
“Sarkiciler.dart” ’tan albümleri göstermek için :

```

@override
Widget build(BuildContext context) {
  return MaterialApp(
    theme: dark ? darkTheme : lightTheme,
    home: Scaffold(
      appBar: AppBar(
        title: Center(child: Text(widget.bartext)),
        leading: IconButton(
          icon: const Icon(Icons.arrow_back),
          onPressed: () {
            Navigator.of(context).pop(true);
          },
        ),
      ),
      body: Container(
        margin: const EdgeInsets.all(10.0),
        child: (albummm == 2) ? GridS22(widget.mualbum) : GridS33(widget.mualbum),
      ),
    ),
  );
}

```

7.3. Albümler Sayfası :



=> Her Hangi bir Albümlerine bastığımızda , basıldığı Albümlerine ait şarkıları gözükecek .

=> Class'ın içinde Gridiew Kullanıldı >

“Albums.dart” ’tan şarkıları göstermek için :

```
@override
Widget build(BuildContext context) {
  return MaterialApp(
    theme: dark ? darkTheme : lightTheme,
    home: Scaffold(
      appBar: AppBar(
        title: Center(child: Text(widget.bartext)),
        leading: IconButton(
          icon: const Icon(Icons.arrow_back),
          onPressed: () {
            Navigator.of(context).pop(true);
          },
        ),
      ),
      body: ListView.builder(
        itemCount: widget.musarki.length,
        itemBuilder: (context, index) {
          return GestureDetector(
            child: Container(
              margin: const EdgeInsets.only(left: 10.0, right: 10.0),
              child: Grid5(
                isim: widget.musarki[index].isim,
                index: widget.musarki[index].id - 1,
                id: widget.musarki[index].id,
                url: widget.musarki[index].url,
              ),
            ),
            onTap: () {
              Navigator.push(
                context,
                MaterialPageRoute(
                  builder: (context) =>
                    MediaPlayer(musarki: widget.musarki, index: index)),
              );
            },
          );
        },
      ),
    ),
  );
}
```

```

    );
  },
);
}

```

7.4. MediaPlayer Sayfası :



=> Her Hangi bir şarkıyı bastığımızda , basıldığı şarkıya ait müziğini açılacak.

=> (MediaPlayer) Kullanıldı .

MediaPlayer kütüphanesini eklemek : (pubspec.yaml)'da

```
audioplayers: ^0.20.1
```

MediaPlayer tanımlamak :

```
final audioPlayer = MediaPlayer();
```

URL'yi ayarla :

```
Future setAudio(String url) async {
  await globals.audioPlayer.setUrl(url);
  globals.audioPlayer.resume();
}
```

Müziği açmak :

```
globals.audioPlayer.onPlayerStateChanged.listen((state) {
  setState(() {
    isPlaying = state == PlayerState.PLAYING;
  });
});
```

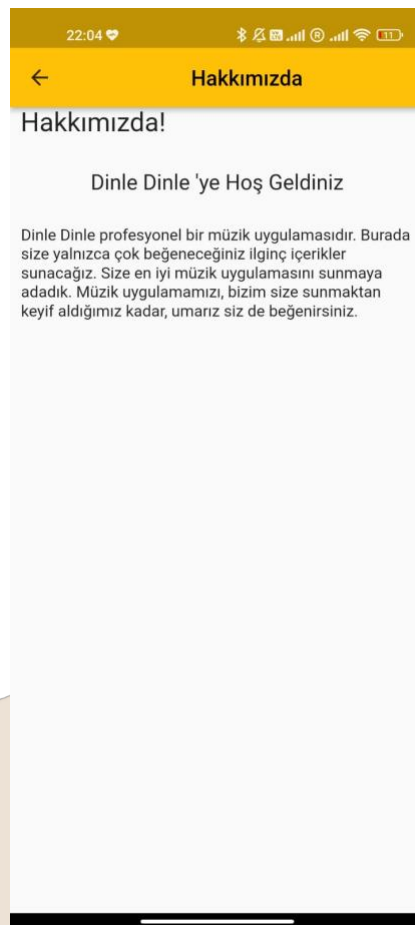
Slider'in değeri değiştirmek :

```
Slider(
  min: 0,
  max: duration.inSeconds.toDouble(),
  value: position.inSeconds.toDouble(),
  onChanged: (value) async {
    final position = Duration(seconds: value.toInt());
    await globals.audioPlayer.seek(position);
    await globals.audioPlayer.resume();
  },
)
```

```
globals.audioPlayer.onDurationChanged.listen((newDuration) {
  setState(() {
    duration = newDuration;
  });
});

globals.audioPlayer.onAudioPositionChanged.listen((newPosition) {
  setState(() {
    position = newPosition;
  });
});
```

7.5. Hakkımızda Sayfası :



```
class aboutUs extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      theme: dark ? darkTheme : lightTheme,
      home: Scaffold(
        appBar: AppBar(
          title: const Center(child: Text("Hakkımızda")),
          leading: IconButton(
            icon: const Icon(Icons.arrow_back),
            onPressed: () {
              Navigator.of(context).pop(true);
            },
          ),
        ),
        body: Column(
          children: [
```

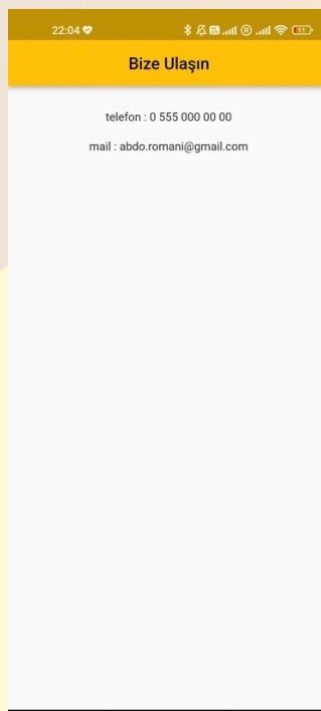
```

Container(
  alignment: Alignment.centerLeft,
  margin: const EdgeInsets.only(left: 10),
  child: const Text(
    "Hakkımızda!",
    style: TextStyle(fontSize: 25),
  ),
),
Container(
  margin: const EdgeInsets.only(left: 10.0, right: 10, top: 30),
  child: const Center(
    child: Text(
      "Dinle Dinle 'ye Hoş Geldiniz",
      style: TextStyle(fontSize: 20),
    ),
  ),
),
Container(
  margin: const EdgeInsets.only(left: 10.0, right: 10, top: 30),
  child: const Center(
    child: Text(
      "Dinle Dinle profesyonel bir müzik uygulamasıdır. "
      "Burada size yalnızca çok beğeneceğiniz ilginç içerikler sunacağız.

      "Size en iyi müzik uygulamasını sunmaya adadık. "
      "Müzik uygulamamızı, bizim size sunmaktan keyif aldığımız kadar, "
      "umamız siz de beğenirsiniz.",
      style: TextStyle(fontSize: 15),
    ),
  ),
),
),
),
),
);
}

```

7.6. Bize Ulaşın Sayfası :



```

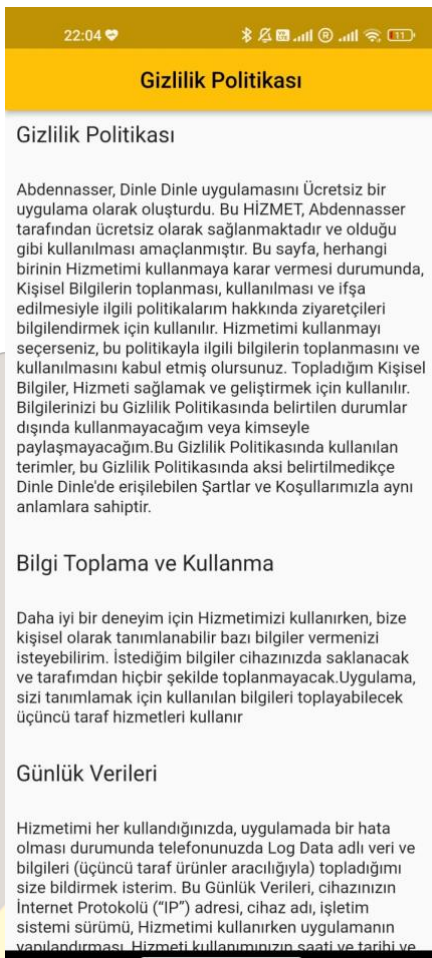
class ContactUs extends StatelessWidget {
  @override

```



```
Widget build(BuildContext context) {
  return MaterialApp(
    theme: dark ? darkTheme : lightTheme,
    home: Scaffold(
      appBar: AppBar(
        title: const Center(child: Text("Bize Ulaşın")),
      ),
      body: Center(
        child: Column(
          children: [
            Container(
              margin: const EdgeInsets.only(top: 30, bottom: 10),
              child: const Text("telefon : 0 555 000 00 00"),
            ),
            Container(
              margin: const EdgeInsets.all(10.0),
              child: const Text("mail : abdo.romani@gmail.com"),
            ),
          ],
        ),
      ),
    ),
  );
}
```

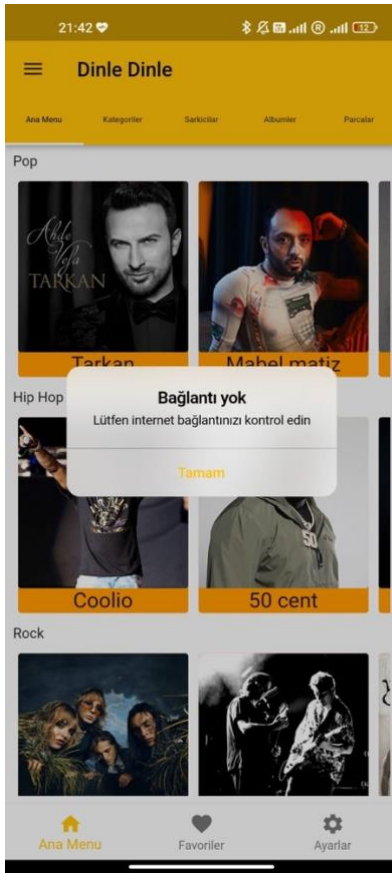
7.7. Gizlilik Politikası Sayfası :



=> Gizlilik Politikası "<https://app-privacy-policy-generator.firebaseapp.com>" web sitesi yardımıyla yapıldı .

8. Ek olarak :

8.1. internet var mı yok mu?



=> Uygulamamızı internet ile çalıştığı için internet var olup olmadığını kontrol etmeliyiz .

=> internet_connection_checker ve connectivity_plus kullanarak yapıldı .

kütüphaneleri eklemek için : (pubspec.yaml)'da

```
internet_connection_checker: ^0.0.1+4  
connectivity_plus: ^2.3.5
```

interneti kontrol etme fonksiyonu :

```
late StreamSubscription subscription;  
var isDeviceConnected = false;  
bool isAlertSet = false;  
  
getConnectivity() =>  
  subscription = Connectivity().onConnectivityChanged.listen((ConnectivityResult  
  result) async {  
  
    isDeviceConnected = await InternetConnectionChecker().hasConnection;  
    if(!isDeviceConnected && isAlertSet == false){  
  
      showDialogBox();  
      setState(() => isAlertSet = true);  
  
    }  
  });
```

interneti yok ise bir dialog gözüktüyor :

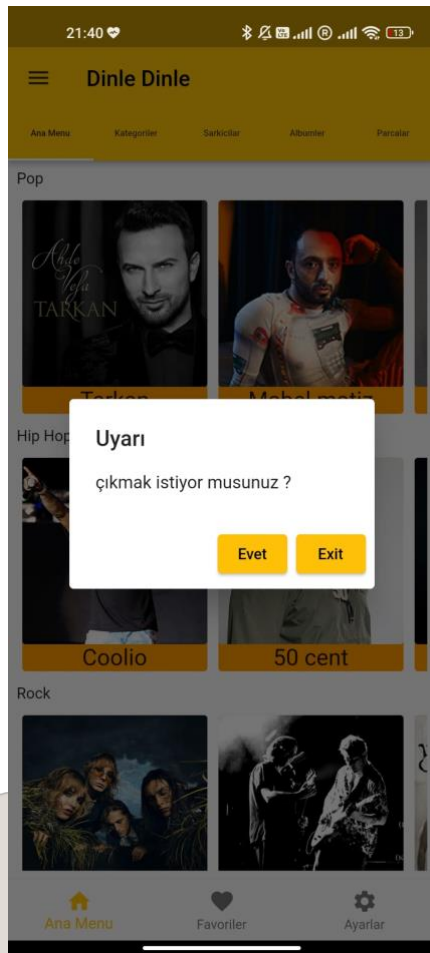
```
showDialogBox() => showCupertinoDialog<String>(context: context, builder:  
(BuildContext context) => CupertinoAlertDialog(  
  title: const Text("Bağlantı yok"),  
  content: const Text("Lütfen internet bağlantınızı kontrol edin"),  
  actions: <Widget>[
```

```

    TextButton(
      onPressed: () async{
        Navigator.pop(context,"Cancel");
        setState(() => isAlertSet = false);
        isDeviceConnected = await InternetConnectionChecker().hasConnection;
        if(!isDeviceConnected){
          showDialogBox();
          setState(() => isAlertSet = true);
        }
      },
      child: const Text("Tamam"))
    ],
  ));

```

8.2. Uygulama'dan çıkmak istediğimizde.



=> Uygulamadan çıkmak istediğimizde bir dialog gözükmeli , kullanıcı emin olmak için .

"onWillPop" kullanıldı :

```

WillPopScope(
  onWillPop: () async {
    final value = await showDialog<bool>(
      context: context,
      builder: (context) {
        return AlertDialog(
          title: Text("Uyarı"),
          content: Text("çıkmaq istiyor musunuz ?"),
          actions: [
            ElevatedButton(
              onPressed: () => Navigator.of(context).pop(false),
              child: Text("Evet")),
            ElevatedButton(
              onPressed: () => Navigator.of(context).pop(true),

```

```

        child: Text("Exit")),
      ],
    );
  });
if (value != null) {
  return Future.value(value);
} else {
  return Future.value(false);
}
},

```

8.3. Dark Modu fotoğrafları :

