

# IOS İLE MOBİL UYGULAMA GELİŞTİRME PROJESİ

Ad:Bünyamin

Soyad: HAMİDİOĞLU

Okul Numarası:19290247

Ders Kodu: BLM4538

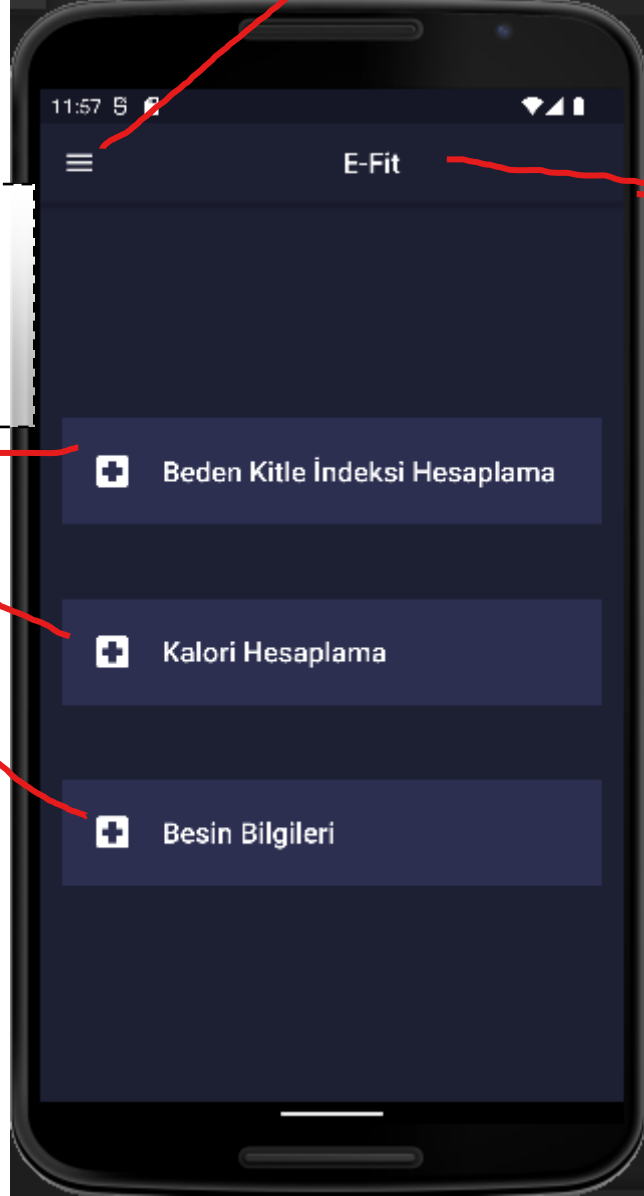
Proje Konusu: İnsanların bir günde yakması gereken kaloriyi ve vücut kitle endeksini hesaplıyor. Besinler hakkında kalori bilgisi veriyor.

## Anasayfa Ekranı

Giri ve kayıt ekranı için drawer ekledik

FlatButton kullanarak  
beden kitle endeksi  
hesaplama, kalori  
hesaplama ve besin  
bilgileri ekranlara  
yönlendirme yaptık.

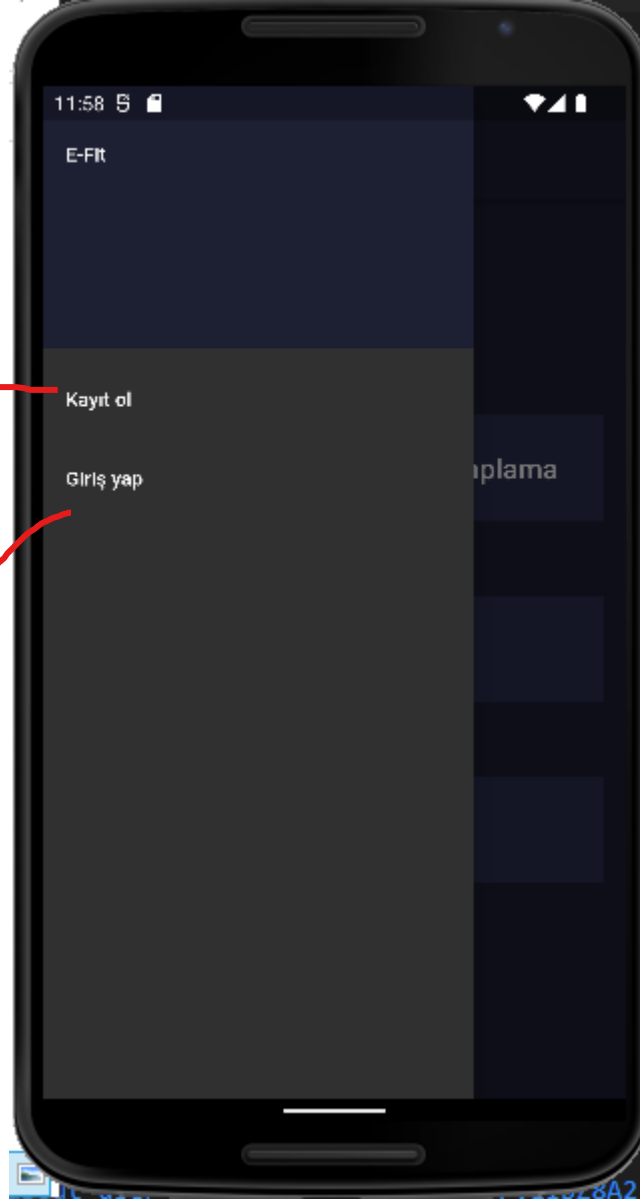
AppBar ekledik ve  
uygulamamızın ismi  
E-Fit yaptık



## Kullanıcı Kaydol ve Giriş Ekranı

ListTile kullanarak  
kayıt ol  
ekranına  
yönlendirme  
yaptık.

ListTile kullanarak  
kayıt ol ekranına  
yönlendirme yaptık.



## Kaydol Ekranı

The screenshot shows a mobile app interface for registration. At the top, there is a status bar with the time 11:58 and battery level 5. Below the status bar is a dark blue header with a back arrow on the left and the text "Kayıt ol" in the center. The main content area is a light blue rounded rectangle containing four input fields: "Adı" (with a person icon), "E-Mail" (with an envelope icon), "Parola" (with a key icon), and "Parola Tekrar" (with a key icon). At the bottom of this rectangle is a white rounded button with the text "Kayıt ol". Red annotations are present: a bracket on the left side of the input fields with the text "Firestore kullanarak veritabanı balantısı oluşturduk ve buradaki verileri firestore'e çektik.", an arrow pointing to the "Kayıt ol" text in the header with the text "AppBar ekledik", and an arrow pointing to the "Kayıt ol" button with the text "Kayıt ol butonuna basarak verileri firestore'e yolluyoruz".

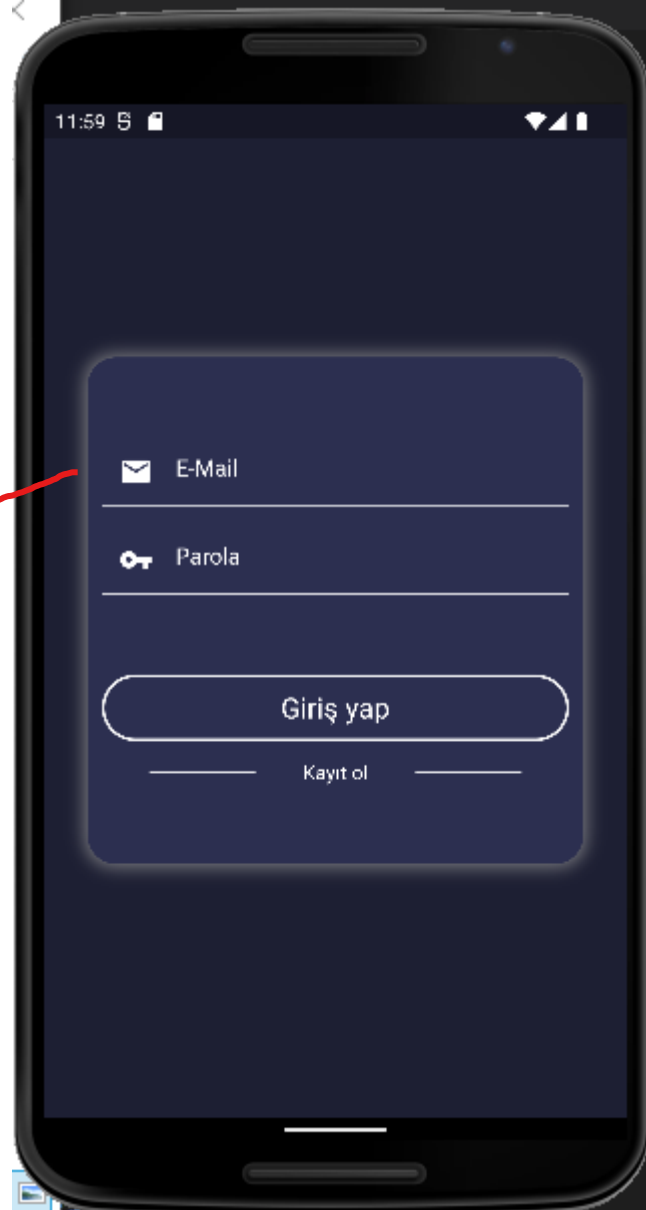
Firestore kullanarak veritabanı balantısı oluşturduk ve buradaki verileri firestore'e çektik.

AppBar ekledik

Kayıt ol butonuna basarak verileri firestore'e yolluyoruz

## Giriş Ekranı

Kayıt ol  
ekranında  
kullandığımız  
mail ve şifreyi  
girerek giriş  
yapıyoruz.



## Vücut Kitle Endeksi(BKİ) Hesaplama ve Sonuç Ekranı

Cinsiyet secimi yapıyoruz.

10:51 5

← BKİ HESAPLAMA

♂ ERKEK

♀ KADIN

BOY  
180 CM

KİLO 60 YAŞ 20

− + − +

HESAPLA

Boy secimi yapıyoruz.

Kilo secimi yapıyoruz

Yas secimi yapıyoruz.

Hesapla butonuna basarak girdigimiz verilere göre bki sonuç ekrana yönlendiriyor.

10:51 5

← Beden Kitle İndeksi Hesapla

Sonuç

Normal

18.5

Normal kilodasınız. Yediklerinize ve egzersizlerinize devam edebilirsiniz.

TEKRAR HESAPLA

Girdigimiz verilere göre vücut kitle endeksini hesaplayip endeksiniz hakkında bilgi veriyor.

Tekrar hesapla butonuna bastigimizda bki hesaplama ekranina yönlendiriliyoruz.

## Kalori Hesaplama ve Sonuç Ekranı

10:52

Kalori Hesabı

Erkek ☒ Kadın ☐

BOY  
180 CM

KİLO 60 YAŞ 20

Çalışma Biçimi: Lütfen Seçim Yapınız

HESAPLA

ButtonBar kullanarak cinsiyet secimini yaptık.

Expanded ile boy secimi ayarlıyoruz.

Kilo secimini

Yas secimini

DropDownButton kullanarak çalışma biçimini ayarladık.

10:52

Kalori Hesabı

Sonuç

Günlük minimum almanız gereken kalori değeri:

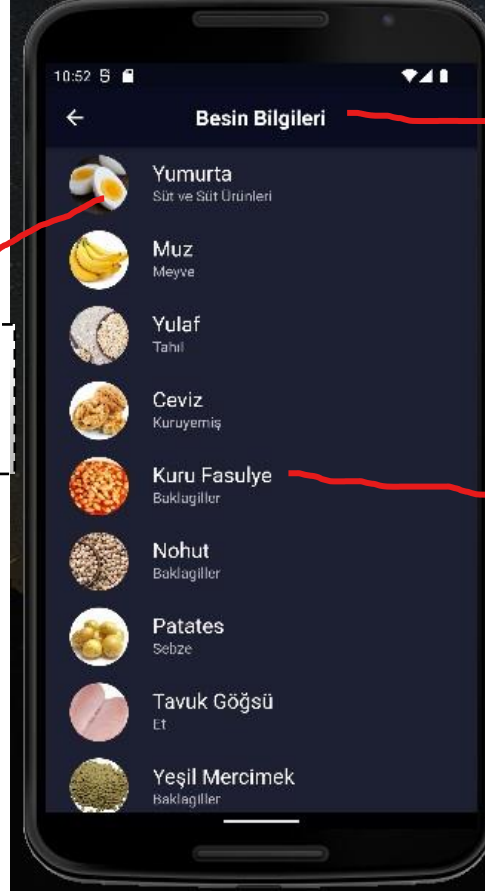
3735 Kalori

TEKRAR HESAPLA

Girdigimiz verilere bir günde almamız gereken kalori degerini veriyor.

Tekrar hesapla butonuna basarak kalori hesaplama ekranina yönlendiriliyoruz.

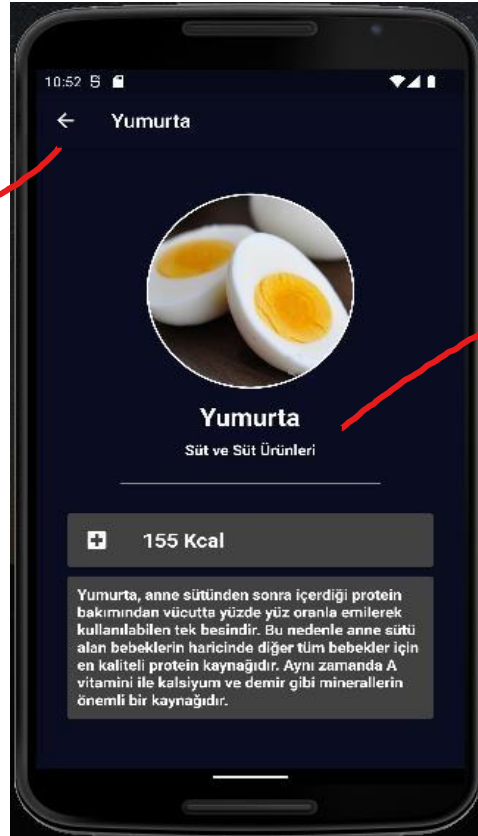
## Besin Bilgileri Ekranı



Besinlerin resimleri, isimleri ve türlerini ekledik.

AppBar ekledik. AppBar'a besin bilgileri textini ekledik.

Besinlerden birine tıkladığımızda besin ile ilgili ayrıntı veren ekrana yönlendiriliyoruz.



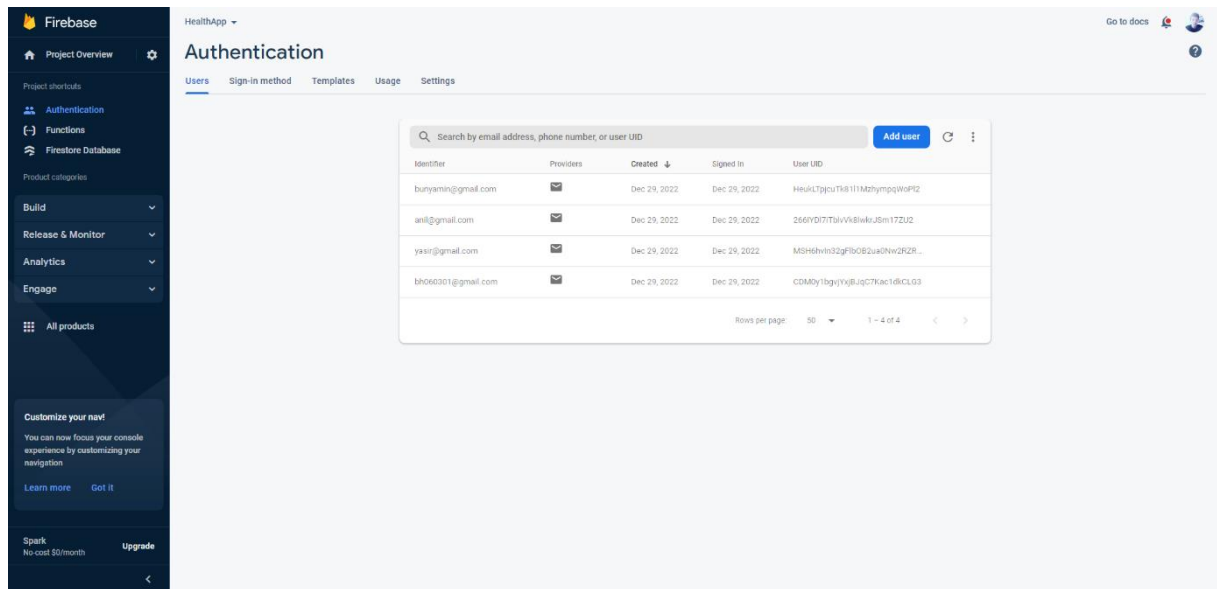
Geri gelme butonuna basarak besin bilgileri ekranına yönlendiriliyoruz

Secilen besinin ismi, resmi, türü ve besin hakkında bilgi verdik



```
login.dart x firebase_options.dart x auth_service.dart x
1 import 'package:cloud_firestore/cloud_firestore.dart';
2 import 'package:firebase_auth/firebase_auth.dart';
3
4 class AuthService {
5   final FirebaseAuth _auth = FirebaseAuth.instance;
6   final FirebaseFirestore _firestore = FirebaseFirestore.instance;
7
8   Future<User?> signIn(String email, String password) async {
9
10    var user = await _auth.signInWithEmailAndPassword(
11      email: email, password: password);
12
13    return user.user;
14  }
15
16  signOut() async {
17    return await _auth.signOut();
18  }
19
20  Future<User?> createPerson(String name, String email, String password) async {
21    var user = await _auth.createUserWithEmailAndPassword(
22      email: email, password: password);
23
24    await _firestore
25      .collection("Person")
26      .doc(user.user!.uid)
27      .set({'userName': name, 'email': email});
28
29    return user.user;
30  }
31 }
```

Firestore bağlantısını yapan kod parçası



The screenshot shows the Firebase Authentication console. On the left is a sidebar with navigation options: Project Overview, Authentication, Functions, and Firestore Database. The main area is titled 'Authentication' and has tabs for Users, Sign-in method, Templates, Usage, and Settings. The 'Users' tab is active, displaying a table of users. At the top of the table is a search bar and an 'Add user' button. The table has columns for Identifier, Providers, Created, Signed in, and User UID. There are four users listed, all created on Dec 29, 2022. At the bottom of the table, there is a 'Rows per page' dropdown set to 50 and a pagination indicator showing '1 - 4 of 4'.

Identifier	Providers	Created	Signed in	User UID
bunyamin@gmail.com	📧	Dec 29, 2022	Dec 29, 2022	HeuKLTpjuTks111MzhymqWwP02
anil@gmail.com	📧	Dec 29, 2022	Dec 29, 2022	266YD7ITuVvV8liksJSm17ZlQ
yasin@gmail.com	📧	Dec 29, 2022	Dec 29, 2022	MSH6hvk32gFb0BQa0NwQFZ6...
bh06001@gmail.com	📧	Dec 29, 2022	Dec 29, 2022	CDM0y1bgqYv8JqCTKac1dkCLG3

Firebasedeki kullanıcıların listesi

```

class _RemoteJsonState extends State<RemoteJson> {
  Future<List<Urunler>> _getUrunler() async{
    var empData=await http.get(Uri.parse("https://api.jsonbin.io/v3/qs/63acbb2e15ab31599e26f2a3"));
    var jsonData=json.decode(empData.body);
    List<Urunler> urunlerrr=[];
    for(var emp in jsonData)
    {
      Urunler urunler=Urunler(emp["urunID"], emp["urunAd"], emp["urunKategori"], emp["urunFoto"],emp["urunKalori"], emp["urunHakkinda"]);
      urunlerrr.add(urunler);
    }
    return urunlerrr;
  }
}

```

Ürünlerin geldiği api'nin entegrasyonunu yapan kod parçası

```

import 'package:flutter/material.dart';
import 'dart:math';
class BKiHesap{
  BKiHesap( {this.bireyBoy,this.bireyKilo} );
  final int? bireyBoy;
  final int? bireyKilo;
  late double _bki;

  String bkihesapla(){
    _bki = bireyKilo!/pow(bireyBoy!/100,2);
    return _bki.toStringAsFixed(1);
  }

  String sonucSiniflama(){
    if(_bki>=25)
      return 'Kilolu';
    else if(_bki>=18.5)
      return 'Normal';
    else
      return 'Zayıf';
  }

  String sonucAciklamasi(){
    if(_bki>=25)
      return 'Kilonuz olmanız gerekenden yüksek, Egzersiz yapmanızı tavsiye ederim.';
    else if(_bki>=18.5)
      return 'Normal kilodasınız. Yediklerinize ve egzersizlerinize devam edebilirsiniz.';
    else
      return 'Normal kilonuzun altındasınız. Yediklerinize dikkat edin.';
  }
}

```

Sonuçlarının hesaplandığı ve açıklandığı kod parçası