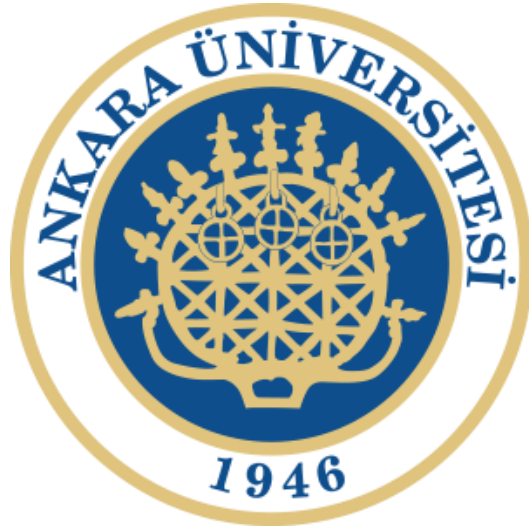


ANKARA ÜNİVERSİTESİ
MÜHENDİSLİK
FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ



BLM4537 PROJE SUNUMU
MALZEMEDEN MASAYA
UYGULAMASI
HALİME GİLDAN
20290252
ENVER BAĞCI

09-01-2024

ÖZET:

Ankara Üniversitesi Bilgisayar Mühendisliği bölümü derslerinden BLM4537 IOS ile Mobil Uygulama Geliştirme I dersi kapsamında aldığım “Malzemedен Masaya” isimli projem için çalışmalar yaptım. Gerekli kurulumlardan ve Figma tasarım programından hazırladığım tasarımdan sonra projenin mimarisini hazırlamaya başladım. Öncelikle Frontend kısmı ile ilgilenip gerekli görünümleri ayarladım. Daha sonrasında Backend tarafına başlayıp gerekli ayarlamaları yaptım. Gerekli API’leri yazıp bağlantıyı sağladıktan sonra uygulama hazır oldu.

Kodlar Githubımda bulunmaktadır : https://github.com/halimegldn/Flutter_Project

GİRİŞ:

FRONTEND KODLARI:

İlk olarak kullanıcı kaydı ve girişı sayfasının görünümünü oluşturdum. Görünümde giriş sayfası, kayıt sayfası, malzeme ekleme sayfası, yemek türü seçmek için sayfa, kalori değeri girmek için sayfa, pişirme süresi girme sayfası, ayarlar sayfası ve paylaşma sayfası bulunmaktadır.

Kayıtlar için Firebase Veri tabanında “Users” tablosu oluşturdum.

Frontend kayıt ekranında Kullanıcı Maili, Şifre, Tekrar Şifre olmak üzere 3 değişken isteniyor. Giriş sayfasında Firebase Veri tabanında kayıtlı kişilerin giriş yapması için Mail ve Şifre içeren değişkenler bulunmakta.

Kayıt Ekranı:

```
main.dart x
1  import 'package:flutter/material.dart';
2  import 'package:firebase_core/firebase_core.dart';
3  import 'package:firebase_auth/firebase_auth.dart';
4  import 'Login_Page.dart';
5  import 'firebase_options.dart';
6
7  //Firebase bağlantısı
8  void main() async {
9    WidgetsFlutterBinding.ensureInitialized();
10
11    try {
12      await Firebase.initializeApp(
13        options: DefaultFirebaseOptions.currentPlatform,
14      );
15    } catch (e) {
16      print("Firebase bağlantı hatası: $e");
17    }
18
19    runApp(MyApp());
20  }
21
22  class MyApp extends StatelessWidget {
23    @override
24    Widget build(BuildContext context) {
25      return MaterialApp(
26        debugShowCheckedModeBanner: false,
27        home: RegisterPage(),
28      ); // MaterialApp
29    }
30  }
31
32  //Kayıt sayfası
33  class RegisterPage extends StatefulWidget {
34    @override
35    _RegisterPageState createState() => _RegisterPageState();
36  }
37
38  //Kullanıcı bilgilerini alma işlemi
39  class _RegisterPageState extends State<RegisterPage> {
40    final FirebaseAuth _auth = FirebaseAuth.instance;
41
42    final TextEditingController _emailController = TextEditingController();
43    final TextEditingController _passwordController = TextEditingController();
44    final TextEditingController _confirmPasswordController =
45      TextEditingController();
46
47    Future<void> _register() async {
48      try {
49        if (_passwordController.text != _confirmPasswordController.text) {
50          print("Hata: Şifreler eşleşmiyor.");
51          return;
52        }
53
54        UserCredential userCredential =
55          await _auth.createUserWithEmailAndPassword(
56            email: _emailController.text,
57            password: _passwordController.text,
58          );
59
60        print("Kullanıcı kaydı başarılı: ${userCredential.user!.uid}");
61
62        _showLoginPage();
63      } on FirebaseAuthException catch (e) {
64        print("Kullanıcı kaydı hatası: ${e.message}");
65      }
66    }
67
68    //Giriş sayfasına yönlendirme kodu
69    void _showLoginPage() {
70      Navigator.pushReplacement(
71        context,
72        MaterialPageRoute(builder: (context) => LoginPage()),
73      );
74    }
75
76    // Giriş sayfasına gitmek için kullanılacak fonksiyon
77    void _goToLoginPage() {
78      Navigator.pushReplacement(
79        context,
80        MaterialPageRoute(builder: (context) => LoginPage()),
81      );
82    }
83  }
```

```

75 // Giriş sayfasına gitmek için kullanılacak fonksiyon
76 void _goToLoginPage() {
77   Navigator.pushReplacement(
78     context,
79     MaterialPageRoute(builder: (context) => LoginPage()),
80   );
81 }
82
83 //Kayıt sayfası stil özellikleri
84 @override
85 Widget build(BuildContext context) {
86   return Scaffold(
87     appBar: null,
88     body: Container(
89       //Sayfanın arkaplan resmi
90       decoration: BoxDecoration(
91         image: DecorationImage(
92           image: NetworkImage(
93             'https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcRuoboEK2anBgaa0W030_LdzwocT00r1QcM1SFbhACsXw&s'), // NetworkImage
94           fit: BoxFit.cover,
95         ), // DecorationImage
96         color: Colors.black.withOpacity(0.3),
97       ), // BoxDecoration
98     child: Center(
99       //Kullanıcı bilgilerini alma işlemi
100       child: Container(
101         padding: const EdgeInsets.all(16.0),
102         child: Column(
103           mainAxisAlignment: MainAxisAlignment.center,
104           children: [
105             TextField(
106               controller: _emailController,
107               style: TextStyle(color: Colors.white),
108               decoration: InputDecoration(
109                 labelText: 'E-posta',
110                 labelStyle: TextStyle(color: Colors.white),
111               ), // InputDecoration
112             ), // TextField
113             TextField(
114               controller: _passwordController,
115             ), // TextField
116             TextField(
117               controller: _passwordController,
118               style: TextStyle(color: Colors.white),
119               decoration: InputDecoration(
120                 labelText: 'Şifre',
121                 labelStyle: TextStyle(color: Colors.white),
122               ), // InputDecoration
123               obscureText: true,
124             ), // TextField
125             TextField(
126               controller: _confirmPasswordController,
127               style: TextStyle(color: Colors.white),
128               decoration: InputDecoration(
129                 labelText: 'Şifreyi Tekrar Girin',
130                 labelStyle: TextStyle(color: Colors.white),
131               ), // InputDecoration
132               obscureText: true,
133             ), // TextField
134             SizedBox(height: 16.0),
135             //Kayıt butonu
136             ElevatedButton(
137               onPressed: _register,
138               style: ElevatedButton.styleFrom(
139                 primary: Colors.black,
140                 shape: RoundedRectangleBorder(
141                   borderRadius: BorderRadius.circular(15.0),
142                 ), // RoundedRectangleBorder
143                 padding: EdgeInsets.symmetric(
144                   vertical: 18.0,
145                   horizontal: 32.0,
146                 ), // EdgeInsets.symmetric
147               ),
148               child: Text('Kayıt Ol'),
149             ), // ElevatedButton
150             SizedBox(height: 8.0),
151             //Kayıt butonuna tıklanınca yapılacak işlemler
152             TextButton(
153               onPressed: _goToLoginPage,
154               style: TextButton.styleFrom(
155                 primary: Colors.black,
156                 padding: EdgeInsets.symmetric(
157                   vertical: 18.0,
158                   horizontal: 32.0,
159                 ), // EdgeInsets.symmetric
160               ),
161               child: Text('Giriş Yap'),
162             ), // TextButton
163           ],
164         ), // Column
165       ), // Container
166     ), // Center
167   ), // Container
168 ); // Scaffold

```

E-posta

Şifre

Şifreyi Tekrar Girin

Kayıt Ol

Giriş Yap

Giriş Ekranı:

```
main.dart x Login_Page.dart x
1  import 'package:flutter/material.dart';
2  import 'Page1.dart';
3  import 'Auth_Service.dart';
4  import 'package:firebase_auth/firebase_auth.dart';
5
6  class LoginPage extends StatefulWidget {
7    //Login sayfası
8    @override
9    LoginPageState createState() => _LoginPageState();
10 }
11
12 class _LoginPageState extends State<LoginPage> {
13   //Kullanıcı bilgilerini alma işlemi
14   final TextEditingController _emailController = TextEditingController();
15   final TextEditingController _passwordController = TextEditingController();
16   final AuthService _authService = AuthService();
17
18   Future<void> _login(BuildContext context) async {
19     // Kullanıcı giriş işlemi
20     User? user = await _authService.signIn(
21       _emailController.text,
22       _passwordController.text,
23     );
24
25     if (user != null) {
26       // Giriş başarılı, Page1 sayfasına yönlendir
27       Navigator.pushReplacement(
28         context,
29         MaterialPageRoute(builder: (context) => Page1()),
30       );
31     } else {
32       // Giriş başarısızsa hata mesajı gösterilebilir
33       showDialog(
34         context: context,
35         builder: (context) {
36
37           return AlertDialog(
38             title: Text('Hata'),
39             content: Text('Giriş başarısız. Lütfen tekrar deneyin.'),
40             actions: [
41               TextButton(
42                 onPressed: () {
43                   Navigator.pop(context);
44                 },
45                 child: Text('Tamam'),
46               ), // TextButton
47             ],
48           ); // AlertDialog
49         },
50       );
51     }
52   }
53
54   //Giriş sayfası stil özellikleri
55   @override
56   Widget build(BuildContext context) {
57     return Scaffold(
58       appBar: null,
59       body: Container(
60         //Arkaplan rengi
61         decoration: BoxDecoration(
62           image: DecorationImage(
63             image: NetworkImage(
64               'https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcRUoboEK2anBgaa0W030_LdzwocT00r1QcM1SFbhACsXw&s'), // Network
65             fit: BoxFit.cover,
66           ), // DecorationImage
67           color: Colors.white.withOpacity(0.3),
68         ), // BoxDecoration
69         child: Center(
70           child: Container(
71             //Form oluşturma
```

```

70 //Form oluşturma
71 padding: const EdgeInsets.all(16.0),
72 child: Column(
73   mainAxisAlignment: MainAxisAlignment.center,
74   children: [
75     TextField(
76       controller: _emailController,
77       style: TextStyle(color: Colors.white),
78       decoration: InputDecoration(
79         labelText: 'E-posta',
80         labelStyle: TextStyle(color: Colors.white),
81       ), // InputDecoration
82     ), // TextField
83     TextField(
84       controller: _passwordController,
85       style: TextStyle(color: Colors.white),
86       decoration: InputDecoration(
87         labelText: 'Şifre',
88         labelStyle: TextStyle(color: Colors.white),
89       ), // InputDecoration
90       obscureText: true,
91     ), // TextField
92     SizedBox(height: 16.0),
93     //Buton style özellikleri
94     ElevatedButton(
95       onPressed: () => _login(context),
96       style: ElevatedButton.styleFrom(
97         primary: Colors.black,
98         shape: RoundedRectangleBorder(
99           borderRadius: BorderRadius.circular(15.0),
100         ), // RoundedRectangleBorder
101         padding: EdgeInsets.symmetric(
102           vertical: 18.0,
103           horizontal: 32.0,
104         ), // EdgeInsets.symmetric
105       ), // ElevatedButton.styleFrom
106       child: Text('Giriş Yap', style: TextStyle(color: Colors.white)),
107     ), // ElevatedButton
108   ],
109 ), // Column
110 ), // Container
111 ), // Center
112 ), // Container
113 ); // Scaffold
114 }
115 }
116 }

```

E-posta

Şifre

Giriş Yap

Malzeme Ekleme Sayfası:

Malzeme ekleme, yemek türü seçme, kalori aralığı belirleme, pişirme süresi seçme işlemleri için bir TabBar oluşturdum. 4 sayfayı tek bir sayfanın farklı bölümleri gibi göstermiş oldum.

TabBar:

```
main.dart x Login_Page.dart x Page1.dart x
1 import 'package:flutter/material.dart';
2 import 'package:cloud_firestore/cloud_firestore.dart';
3 import 'Share_Page.dart';
4 import 'Setting.dart';
5 import 'Page2.dart';
6 import 'Page3.dart';
7 import 'Page4.dart';
8
9 class Page1 extends StatefulWidget {
10   @override
11   Page1State createState() => _Page1State();
12 }
13
14 class _Page1State extends State<Page1> {
15   @override
16   Widget build(BuildContext context) {
17     return MaterialApp(
18       debugShowCheckedModeBanner: false,
19       home: Scaffold(
20         backgroundColor: Color(0xFFB0BEC5),
21         body: DefaultTabController(
22           length: 4,
23           child: Column(
24             children: [
25               TabBar(
26                 tabs: [
27                   Tab(text: 'Malzeme'),
28                   Tab(text: 'Yemek T'),
29                   Tab(text: 'Kalori'),
30                   Tab(text: 'Pişme S'),
31                 ],
32               ), // TabBar
33             Expanded(
34               child: TabBarView(
35                 children: [
36                   MealPage(),
37                   SecondPage(),
38                   ThirdPage(),
39                   FourthPage(),
40                 ],
41               ), // TabBarView
42             ), // Expanded
43           ],
44         ), // Column
45       ), // DefaultTabController
46     ), // Scaffold
47   ); // MaterialApp
48 }
49 }
```

Malzeme Ekleme Sayfası:

Kullanıcının elinde bulunan malzemeleri girip o malzemeleri içeren yemekleri sunmasını sağlayan bir sayfa oluşturdum. Kullanıcı elinde bulunan malzeme sayısına göre malzeme ekleyebilecek.

Girilen malzemeleri Firebase oluşturduğum “Meals” tablosundaki malzemeleri içeren yemeğe göre filtreleme yapıyor.

```
class MealPage extends StatefulWidget {  
  @override  
  MealPageState createState() => _MealPageState();  
}  
  
class _MealPageState extends State<MealPage> {  
  final FirebaseFirestore _firestore = FirebaseFirestore.instance;  
  List<TextEditingController> malzemeControllers = [];  
  
  //Başlangıçta 3 malzeme text i eklendi  
  @override  
  void initState() {  
    super.initState();  
    malzemeControllers.add(TextEditingController());  
    malzemeControllers.add(TextEditingController());  
    malzemeControllers.add(TextEditingController());  
  }  
  
  @override  
  Widget build(BuildContext context) {  
    return Scaffold(  
      backgroundColor: Color(0xFFB0BEC5),  
      body: Column(  
        children: [  
          SizedBox(height: 16),  
          //+ Butonu oluşturuldu  
          ElevatedButton(  
            style: ElevatedButton.styleFrom(  
              primary: Color(0xFF535252),  
              shape: RoundedRectangleBorder(),  
              padding: EdgeInsets.all(18.0),  
            ),  
            //+ Butonuna tıklanıldıkça text ekleme işlemi yapılıyor  
            onPressed: () {  
              setState(() {
```

```

        malzemeControllers.add(TextEditingController());
    });
},

    child: Text(
      '+',
      style: TextStyle(fontSize: 16.0),
    ), // Text
  ), // ElevatedButton
  for (int i = 0; i < malzemeControllers.length; i++)
    Padding(
      padding: const EdgeInsets.all(8.0),
      child: TextField(
        controller: malzemeControllers[i],
        decoration: InputDecoration(labelText: 'Malzeme ${i + 1}'),
      ), // TextField
    ), // Padding
  ), // ElevatedButton
  //Arama butonu
  ElevatedButton(
    style: ElevatedButton.styleFrom(
      primary: Color(0xFF535252),
      shape: RoundedRectangleBorder(),
      padding: EdgeInsets.all(18.0),
    ),
    onPressed: () => _onSearchButtonPressed(context),
    child: Text('Ara'),
  ), // ElevatedButton
],
), // Column
); // Scaffold
}

//Butona tıklanınca Firebase araması yapıyor
void _onSearchButtonPressed(BuildContext context) async {
  List<String> ingredients = [],
  for (var controller in malzemeControllers) {
    ingredients.add(controller.text);
  }

  var snapshot = await _firestore
    .collection('meals')
    .where('malzemeler', arrayContainsAny: ingredients)
    .get();

  List<DocumentSnapshot> meals = snapshot.docs;
  Navigator.push(
    context,
    MaterialPageRoute(
      builder: (context) => MeallistPage(meals),
    ), // MaterialPageRoute
  );
}
}

```

| Malzeme | Yemek T | Kalori | Pişme S |
|----------------|---------|--------|---------|
| <div>+</div> | | | |
| Malzeme 1 | | | |
| Malzeme 2 | | | |
| Malzeme 3 | | | |
| <div>Ara</div> | | | |

Yemek Türü Seçme Sayfası:

Kullanıcının görmek istedikleri yemek türünü seçmelerini sağlayan bir sayfa oluşturdum. Girilen yemek türü eşleştirmesini Firebase oluşturduğum “meals” tablosundaki yemek türüne göre filtreleme yapıyor.

```
main.dart x Login_Page.dart x Page1.dart x Page2.dart x
1 import 'package:flutter/material.dart';
2 import 'Page1.dart';
3 import 'package:flutter/material.dart';
4 import 'package:cloud_firestore/cloud_firestore.dart';
5
6 //Yemek türü sayfası
7 class SecondPage extends StatefulWidget {
8   @override
9   _SecondPageState createState() => _SecondPageState();
10 }
11
12 class _SecondPageState extends State<SecondPage> {
13   List<String> selectedTypes = [];
14   //checkbox ile yemeklerin tür adları ve radio butonlar oluşturuldu
15   @override
16   Widget build(BuildContext context) {
17     return Scaffold(
18       backgroundColor: Color(0xFF90A4AE),
19       body: Column(
20         children: [
21           CheckboxListTile(
22             title: Text('Çorba'),
23             value: selectedTypes.contains('Çorba'),
24             onChanged: (value) => _onTypeSelected('Çorba', value ?? false),
25           ), // CheckboxListTile
26           CheckboxListTile(
27             title: Text('Salata'),
28             value: selectedTypes.contains('Salata'),
29             onChanged: (value) => _onTypeSelected('Salata', value ?? false),
30           ), // CheckboxListTile
31           CheckboxListTile(
32             title: Text('Ana Yemek'),
33             value: selectedTypes.contains('Ana Yemek'),
34             onChanged: (value) => _onTypeSelected('Ana Yemek', value ?? false),
35           ), // CheckboxListTile
36           CheckboxListTile(
37             title: Text('Tatlı'),
38             value: selectedTypes.contains('Tatlı'),
39             onChanged: (value) => _onTypeSelected('Tatlı', value ?? false),
40           ), // CheckboxListTile
41
42           //Arama butonu özellikleri
43           ElevatedButton(
44             style: ElevatedButton.styleFrom(
45               primary: Color(0xFF535252),
46               shape: RoundedRectangleBorder(),
47               padding: EdgeInsets.all(18.0),
48             ),
49             onPressed: () => _onSearchButtonPressed(context),
50             child: Text('Ara'),
51           ), // ElevatedButton
52         ],
53       ), // Column
54     ); // Scaffold
55   }
56
57   //Yemek türü araniyor
58   void _onTypeSelected(String type, bool selected) {
59     setState(() {
60       if (selected) {
61         selectedTypes.add(type);
62       } else {
63         selectedTypes.remove(type);
64       }
65     });
66   }
67
68   //Firebase meals tablasında arama işlemi yapılıyor
69   void _onSearchButtonPressed(BuildContext context) async {
70     if (selectedTypes.isNotEmpty) {
71       var snapshot = await FirebaseFirestore.instance
72         .collection('meals')
73         .where('yemek_turu', whereIn: selectedTypes)
74         .get();
75
76       List<DocumentSnapshot> meals = snapshot.docs;
77       Navigator.push(
78         context,
79         MaterialPageRoute(
80           builder: (context) => MealListPage(meals),
81         ), // MaterialPageRoute
82       );
83     }
84   }
85 }
```

```
81         ), // MaterialPageRoute
82     );
83 } else {
84     print('Lütfen en az bir yemek türü seçin.');
```

| Malzeme | Yemek T | Kalori | Pişme S |
|---------|-----------|--------|--------------------------|
| | Çorba | | <input type="checkbox"/> |
| | Salata | | <input type="checkbox"/> |
| | Ana Yemek | | <input type="checkbox"/> |
| | Tatlı | | <input type="checkbox"/> |

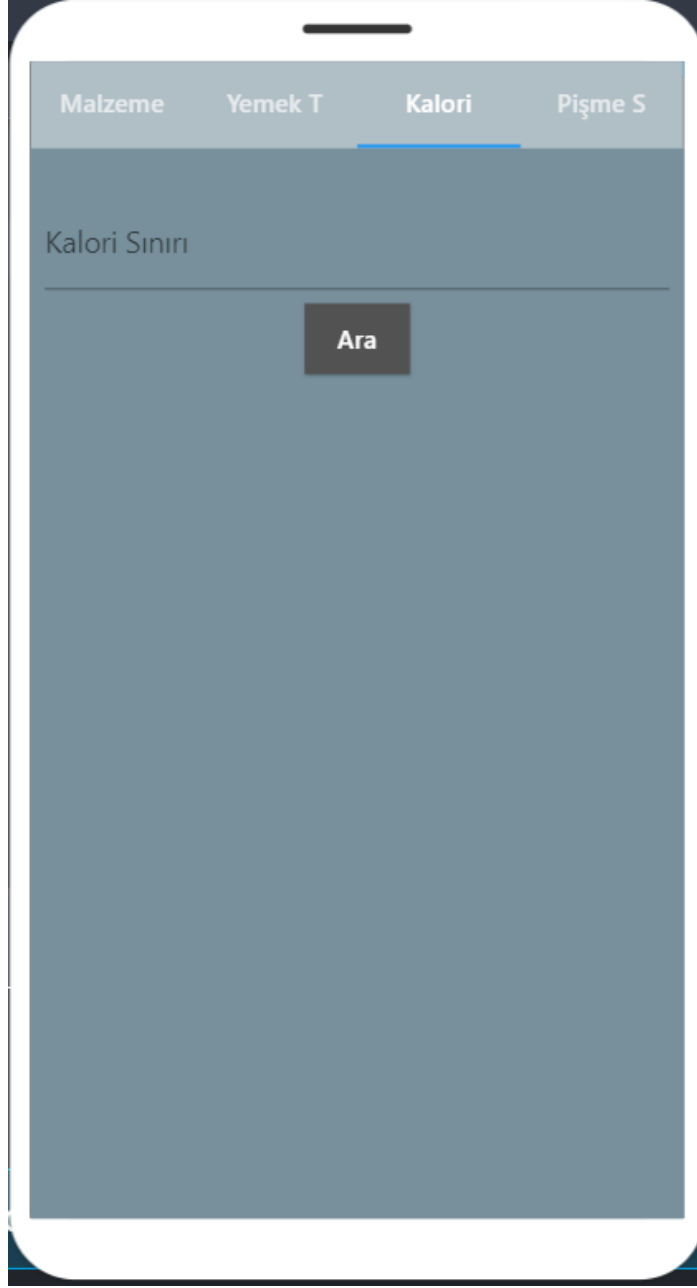
Ara

Kalori Sayfası:

Kullanıcının görmek istedikleri kalori aralığını sağlayan bir sayfa oluşturdum.

Girilen yemek türü eşleştirmesini Firebase oluşturduğum “meals” tablosundaki girilen kalori değerinden düşük kaloriye sahip yemeklerin filtrelenmesini yapıyor.

```
main.dart x Login_Page.dart x Page1.dart x Page2.dart x Page3.dart x
1 import 'package:flutter/material.dart';
2 import 'Page1.dart';
3 import 'package:flutter/material.dart';
4 import 'package:cloud_firestore/cloud_firestore.dart';
5
6 //Kalori sayfası
7 class ThirdPage extends StatefulWidget {
8   @override
9   ThirdPageState createState() => _ThirdPageState();
10 }
11
12 class _ThirdPageState extends State<ThirdPage> {
13   TextEditingController calorieController = TextEditingController();
14
15   @override
16   Widget build(BuildContext context) {
17     return Scaffold(
18       backgroundColor: Color(0xFF78909C),
19       body: Column(
20         children: [
21           SizedBox(height: 16),
22           //Kalori girmek için text oluşturma işlemi
23           Padding(
24             padding: const EdgeInsets.all(8.0),
25             child: TextField(
26               controller: calorieController,
27               keyboardType: TextInputType.number,
28               decoration: InputDecoration(labelText: 'Kalori Sınırı'),
29             ), // TextField
30           ), // Padding
31           //Arama butonu özellikleri
32           ElevatedButton(
33             style: ElevatedButton.styleFrom(
34               primary: Color(0xFF535252),
35               shape: RoundedRectangleBorder(),
36               padding: EdgeInsets.all(18.0),
37             ),
38             onPressed: () => _onSearchButtonPressed(context),
39             child: Text('Ara'),
40           ), // ElevatedButton
41         ],
42       ), // Column
43     ); // Scaffold
44
45     //Firebase meals tablosunda kalori değeri girilen aralığın altında olan yemekleri arama işlemi
46     void _onSearchButtonPressed(BuildContext context) async {
47       int enteredCalorie = int.tryParse(calorieController.text) ?? 0;
48
49       var snapshot = await FirebaseFirestore.instance
50         .collection('meals')
51         .where('kalori', isLessThanOrEqualTo: enteredCalorie)
52         .get();
53
54       List<DocumentSnapshot> meals = snapshot.docs;
55       Navigator.push(
56         context,
57         MaterialPageRoute(
58           builder: (context) => MeallistPage(meals),
59         ), // MaterialPageRoute
60       );
61     }
62   }
63 }
64
```



Pişirme Süresi Sayfası:

Kullanıcının görmek süre aralığındaki yemekleri görmelerini sağlayan bir sayfa oluşturdum. Girilen yemek türü eşleştirmesini Firebase oluşturduğum “meals” tablosundaki pişirme süresi aralığındaki yemeklere göre filtreleme yapıyor.

```
main.dart x Login_Page.dart x Page1.dart x Page2.dart x Page3.dart x Page4.dart x
1 import 'package:flutter/material.dart';
2 import 'Page1.dart';
3 import 'package:cloud_firestore/cloud_firestore.dart';
4
5 //Pişme Süresi sayfası
6 class FourthPage extends StatefulWidget {
7   @override
8   _FourthPageState createState() => _FourthPageState();
9 }
10
11 class _FourthPageState extends State<FourthPage> {
12   int selectedTime = 15;
13
14   @override
15   Widget build(BuildContext context) {
16     return Scaffold(
17       backgroundColor: Color(0xFF607D88),
18       body: Column(
19         children: [
20           //Dropdownlist oluşturma işlemi
21           SizedBox(height: 16),
22           DropdownButton<int>(
23             value: selectedTime,
24             items: [
25               DropdownMenuItem(
26                 value: 15,
27                 child: Text('15-30 dakika'),
28               ), // DropdownMenuItem
29               DropdownMenuItem(
30                 value: 30,
31                 child: Text('30-45 dakika'),
32               ), // DropdownMenuItem
33               DropdownMenuItem(
34                 value: 45,
35                 child: Text('45-60 dakika'),
36               ), // DropdownMenuItem
37               DropdownMenuItem(
38                 value: 60,
39                 child: Text('60+ dakika'),
40               ), // DropdownMenuItem
41             ],
42             onChanged: (value) {
43               setState(() {
44                 selectedTime = value!;
45               });
46             },
47           ), // DropdownButton
48           //Arama butonu özellikleri
49           ElevatedButton(
50             style: ElevatedButton.styleFrom(
51               primary: Color(0xFF535252),
52               shape: RoundedRectangleBorder(),
53               padding: EdgeInsets.all(18.0),
54             ),
55             onPressed: () => _onSearchButtonPressed(context),
56             child: Text('Ara'),
57           ), // ElevatedButton
58         ],
59       ), // Column
60     ); // Scaffold
61   }
62
63   //Firebase meals tablosunda arama işlemi
64   void _onSearchButtonPressed(BuildContext context) async {
65     print("Selected Time: $selectedTime");
66     var snapshot = await FirebaseFirestore.instance
67       .collection('meals')
68       .where('ptime_s', isGreaterThanEqualTo: 30, isLessThanOrEqualTo: 45)
69       .get();
70
71     List<DocumentSnapshot> meals = snapshot.docs;
72     print("Number of meals matching the criteria: ${meals.length}");
73
74     if (meals.isEmpty) {
75       // Eğer hiç yemek bulunamadıysa kullanıcıya bilgi ver
76       showDialog(
77         context: context,
78         builder: (context) => AlertDialog(
79           title: Text('Uyarı'),
80           content: Text('Belirtilen sürede yemek bulunamadı '),
81         ),
82       );
83     }
84   }
85 }
```

```

80         content: Text('Belirtilen sürede yemek bulunamadı.'),
81         actions: [
82             TextButton(
83                 onPressed: () => Navigator.pop(context),
84                 child: Text('Tamam'),
85             ), // TextButton
86         ],
87     ), // AlertDialog
88 );
89 } else {
90     // Eğer yemek bulunduysa MeallistPage'e yönlendir
91     Navigator.push(
92         context,
93         MaterialPageRoute(
94             builder: (context) => MeallistPage(meals),
95         ), // MaterialPageRoute
96     );
97 }
98 }
99 }
100

```



Yemek Sonuç Sayfası:

TabBar’da bulunan Malzeme Girme, Yemek Türü seçme, Kalori seçme ve Pişirme Süresi girme sayfalarından herhangi birinden yemek filtrelemesi yaptıktan sonra o özellikleri içeren yemeklerin görüntülenmesini sağlayan bir sayfa oluşturdum.

Firebase “Meals” tablosunda bulunan yemek adı, yemek resmi, yemek türü, kalori, pişirme süresi, malzemeleri, yemek tarifi verilerini çekme işlemini yaptım.

```
class MeallistPage extends StatelessWidget {
  final List<DocumentSnapshot> meals;

  MeallistPage(this.meals);

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: Color(0xFFECC80),
      body: Column(
        children: [
          Expanded(
            child: ListView.builder(
              itemCount: meals.length,
              itemBuilder: (context, index) {
                var mealData = meals[index].data() as Map<String, dynamic>;

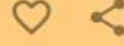
                // Favori ve Paylaş ikonları ekleniyor
                Widget trailingWidget = Row(
                  mainAxisAlignment: MainAxisAlignment.min,
                  children: [
                    IconButton(
                      icon: Icon(Icons.favorite_border),
                      onPressed: () {
                        Navigator.push(
                          context,
                          MaterialPageRoute(
                            builder: (context) =>
                              FavoritePage(mealData['isim']),
                          ), // MaterialPageRoute
                        );
                      },
                    ), // IconButton
                    IconButton(
                      icon: Icon(Icons.share),
                      onPressed: () {
                        Navigator.push(
                          context,
                          MaterialPageRoute(
                            builder: (context) => SharePage(mealData['isim']),
                          ), // MaterialPageRoute
                        );
                      },
                    ), // IconButton
                  ],
                );
                // Row
                //Yemeklerin adları, resimler, türleri, malzemeleri, pişme süreleri ve tarifi gösteriliyor
                return ListTile(
                  title: Text(mealData['isim'] ?? ''),
                  subtitle: Column(
                    crossAxisAlignment: CrossAxisAlignment.start,
                    children: [
                      SizedBox(height: 8),
                      Container(
                        height:
                          200, // Yüksekliği ayarlanabilir (istediğiniz değeri girin)
                        child: mealData['resim_url'] != null
                          ? Image.network(
                              mealData['resim_url'],
                              fit: BoxFit
                                .cover, // Resmi konteynıra tamamen sığdır
                              // Image.network
                            ) : SizedBox.shrink(),
                      ), // Container
                      SizedBox(height: 8),
                      Text('Kalori: ${mealData['kalori'] ?? ''}'),
                      Text('Malzemeler: ${mealData['malzemeler'] ?? ''}'),
                      Text('Tarif: ${mealData['tarif'] ?? ''}'),
                      Text('Yemek Türü: ${mealData['yemek_turu'] ?? ''}'),
                      Text(
```

```

        Text(
          'Pişirme Süresi: ${mealData['pisime_s'] ?? ''} dakika',
        ), // Text
      ],
    ), // Column
    trailing: trailingWidget,
  ); // ListTile
],
), // ListView.builder
), // Expanded
Container(
  padding: EdgeInsets.symmetric(vertical: 16),
  color: Color(0xFFFFCC80),
  child: Row(
    mainAxisAlignment: MainAxisAlignment
      .spaceEvenly, // Center için spaceEvenly kullanılabilir.
    children: [
      IconButton(
        icon: Icon(Icons.home),
        onPressed: () {
          Navigator.pop(context); // Anasayfaya dön
        },
      ), // IconButton
      IconButton(
        icon: Icon(Icons.search),
        onPressed: () {
          // Search ikonuna tıklandığında yapılacak işlemler buraya eklenebilir
        },
      ), // IconButton
      IconButton(
        icon: Icon(Icons.settings),
        onPressed: () {
          Navigator.push(
            context,
            MaterialPageRoute(
              builder: (context) =>
                FavoritePage(''), // Favored
            ), // MaterialPageRoute
          );
        },
      ), // IconButton
    ],
  ), // Row
), // Container
],
), // Column
); // Scaffold
}
}

```

Mantar orbası



Kalori: 150

Malzemeler: [Soğan, Mantar, Sıvıyağ, Un, Süt]

Tarif: Mantarları temizleyin ve dilimleyin. Geniş bir tencerede tereyağını veya zeytinyağını eritin. Doğranmış soğanları ve sarımsakları ekleyin, yumuşayana kadar kavurun. Dilimlenmiş mantarları ekleyin ve mantarlar suyunu salıp çekene kadar orta ateşte kavurun. Unu ekleyip karıştırın ve birkaç dakika daha kavurun. Unun kokusu çıkana kadar kısık ateşte karıştırarak pişirin. Tavuk veya sebze suyunu yavaşça ekleyin, sürekli karıştırarak topaklanmamasını sağlayın. orbayı kaynamaya bırakın ve ardından sütü ekleyin. Tuz



Ayarlar Sayfası:

Kullanıcıya ait bilgilerin olduğu, favorilere eklenen yemeklerin görüldüğü, uygulamadan çıkış işlemi iconunun olduğu ve diğer sayfalara geçiş yapmayı sağlayan özelliklerin olduğu bir Ayarlar sayfası oluşturdum.

```
main.dart x Login_Page.dart x Page1.dart M x Page2.dart x Page3.dart x Page4.dart M x Setting.dart x
1 import 'package:flutter/material.dart';
2
3 //Ayarlar sayfası
4 class FavoritePage extends StatelessWidget {
5   final String mealName;
6
7   FavoritePage(this.mealName);
8
9   // Yemeği silen metod
10  void deleteMeal(BuildContext context) {
11    ScaffoldMessenger.of(context).showSnackBar(
12      SnackBar(
13        content: Text('$mealName adlı yemek silindi!'),
14        duration: Duration(seconds: 2),
15      ), // SnackBar
16    );
17  }
18
19  @override
20  Widget build(BuildContext context) {
21    return Scaffold(
22      backgroundColor: Color(0xFF90CAF9),
23      //Kullanıcı bilgilerini içeren Container
24      body: Padding(
25        padding: const EdgeInsets.all(16.0),
26        child: Column(
27          crossAxisAlignment: CrossAxisAlignment.start,
28          children: [
29            Container(
30              //color: Color(0xffBBDEFB),
31              height: 100,
32              child: ListTile(
33                leading: CircleAvatar(
34                  child: Icon(Icons.person),
35                  radius: 25,
36                ), // CircleAvatar
37                title: Container(
38                  margin: EdgeInsets.only(left: 20),
39                  child: Column(
40                    crossAxisAlignment: CrossAxisAlignment.start,
41                    children: [
42                      Text(
43                        "Halime GILDAN",
44                        style: TextStyle(
45                          fontSize: 20,
46                          fontWeight: FontWeight.bold,
47                        ), // TextStyle
48                      ), // Text
49                      Text(
50                        "gldnhlm@gmail.com",
51                        style: TextStyle(fontSize: 16),
52                      ), // Text
53                    ],
54                  ), // Column
55                ), // Container
56              ), // ListTile
57            ), // Container
58            //Favorilere eklenen yemeğin gösterilme işlemi
59            SizedBox(height: 16),
60            Row(
61              mainAxisAlignment: MainAxisAlignment.spaceBetween,
62              children: [
63                Text(
64                  'Favori Yemeğiniz:',
65                  style: TextStyle(fontSize: 24, fontWeight: FontWeight.bold),
66                ), // Text
67                GestureDetector(
68                  onTap: () {
69                    // Silme işlevselliğini çağır
70                    deleteMeal(context);
71                  },
72                ),
73              ],
74            ),
75          ],
76        ),
77      ),
78    );
79  }
80 }
```

```

71     },
72     child: Row(
73       children: [
74         Icon(Icons.delete, color: Colors.blue),
75         SizedBox(width: 8),
76         Text(
77           'Sil',
78           style: TextStyle(fontSize: 18, color: Colors.blue),
79         ), // Text
80       ],
81     ), // Row
82   ), // GestureDetector
83 ],
84 ), // Row
85 SizedBox(height: 16),
86 Text(
87   mealName,
88   style: TextStyle(fontSize: 18),
89 ), // Text
90 ],
91 ), // Column
92 ), // Padding
93 bottomNavigationBar: Container(
94   //Çıkış işlemi
95   padding: EdgeInsets.only(left: 16, top: 16, right: 16),
96   child: GestureDetector(
97     onTap: () {},
98     child: Column(
99       mainAxisAlignment: MainAxisAlignment.min,
100       children: [
101         Icon(
102           Icons.exit_to_app,
103           size: 30,
104         ), // Icon
105         SizedBox(height: 10),
106         //Sayfanın altında bulunan Person, Search, Home ikonları
107         Row(
108           mainAxisAlignment: MainAxisAlignment.spaceEvenly,
109           children: [
110             IconButton(
111               icon: Icon(Icons.home),
112               onPressed: () {
113                 Navigator.pop(context); // Anasayfaya dön
114               },
115             ), // IconButton
116             IconButton(
117               icon: Icon(Icons.search),
118               onPressed: () {
119                 // Search ikonuna tıklandığında yapılacak işlemler buraya eklenebilir
120               },
121             ), // IconButton
122             IconButton(
123               icon: Icon(Icons.settings),
124               onPressed: () {
125                 Navigator.push(
126                   context,
127                   MaterialPageRoute(
128                     builder: (context) =>
129                       FavoritePage(''), // Favori sayfasına git
130                   ), // MaterialPageRoute
131                 );
132               },
133             ), // IconButton
134           ],
135         ), // Row
136       ],
137     ), // Column
138   ), // GestureDetector
139 ), // Container
140 ); // Scaffold

```



Halime GİLDAN
gldnhlm@gmail.com

Favori Yemeğiniz:



Sil

Mantar Çorbası



Paylaşma Sayfası:

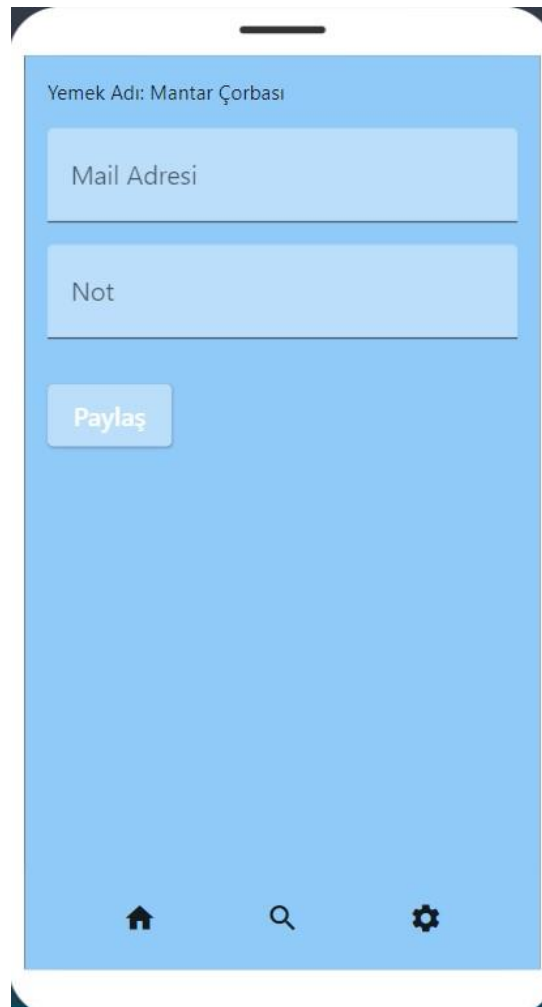
Yemeklerin gösterildiği sonuç sayfasında yemekleri paylaşmak için eklenen paylaş iconu bu sayfaya yönlendirme işlemi yapar. Bu sayfada Paylaşmak istenilen kişinin mail adresi, paylaşılacak not girme işlemleri yapılır.

```
main.dart x Login_Page.dart x Page1.dart M x Page2.dart x Page3.dart x Page4.dart M x Setting.dart x Share_Page.dart x
1 import 'package:flutter/material.dart';
2 import 'Setting.dart';
3
4 //Paylaşma sayfası
5 class SharePage extends StatelessWidget {
6   final String mealName;
7
8   SharePage(this.mealName);
9
10  @override
11  Widget build(BuildContext context) {
12    return Scaffold(
13      backgroundColor: Color(0xFF90CAF9),
14      body: Padding(
15        padding: const EdgeInsets.all(16.0),
16        child: Column(
17          crossAxisAlignment: CrossAxisAlignment.start,
18          children: [
19            Text('Yemek Adı: $mealName'),
20            SizedBox(height: 16),
21            //Paylaşma işlemi için mail ve not girme işlemi
22            TextField(
23              decoration: InputDecoration(
24                labelText: 'Mail Adresi',
25                filled: true,
26                fillColor: Color(0xFFBBDEFB),
27                contentPadding: EdgeInsets.all(16.0),
28              ), // InputDecoration
29              style: TextStyle(fontSize: 18.0),
30              textAlign: TextAlign.center,
31            ), // TextField
32            SizedBox(height: 16),
33            TextField(
34              decoration: InputDecoration(
35                labelText: 'Not',
36                filled: true,
37                fillColor: Color(0xFFBBDEFB),
38                contentPadding: EdgeInsets.all(16.0),
39              ), // InputDecoration
40              style: TextStyle(fontSize: 18.0),
41              textAlign: TextAlign.center,
42            ), // TextField
43            SizedBox(height: 32),
44            ElevatedButton(
45              onPressed: () {
46                // Paylasma işlemleri burada gerçekleştirilebilir.
47              },
48              style: ElevatedButton.styleFrom(
49                primary: Color(0xFFBBDEFB),
50                padding: EdgeInsets.all(18.0),
51              ),
52              child: Text(
53                'Paylaş',
54                style: TextStyle(
55                  color: Colors.white,
56                  fontSize: 18.0,
57                ), // TextStyle
58              ), // Text
59            ), // ElevatedButton
60          ],
61        ), // Column
62      ), // Padding
63      //Sayfanın altında bulunan Setting, Home, Search iconları
64      bottomNavigationBar: Container(
65        padding: EdgeInsets.symmetric(vertical: 16),
66        color: Color(0xFF90CAF9),
67        child: Row(
68          mainAxisAlignment: MainAxisAlignment.spaceEvenly,
69          children: [
70            IconButton(
```

```

71     icon: Icon(Icons.home),
72     onPressed: () {
73       Navigator.pop(context); // Anasayfaya dön
74     },
75   ), // IconButton
76   IconButton(
77     icon: Icon(Icons.search),
78     onPressed: () {
79       // Search ikonuna tıklandığında yapılacak işlemler buraya eklenebilir
80     },
81   ), // IconButton
82   IconButton(
83     icon: Icon(Icons.settings),
84     onPressed: () {
85       Navigator.push(
86         context,
87         MaterialPageRoute(
88           builder: (context) =>
89             FavoritePage(''), // Favori sayfasına git
90         ), // MaterialPageRoute
91       );
92     },
93   ), // IconButton
94 ],
95 ), // Row
96 ), // Container
97 ); // Scaffold
98 }
99 }

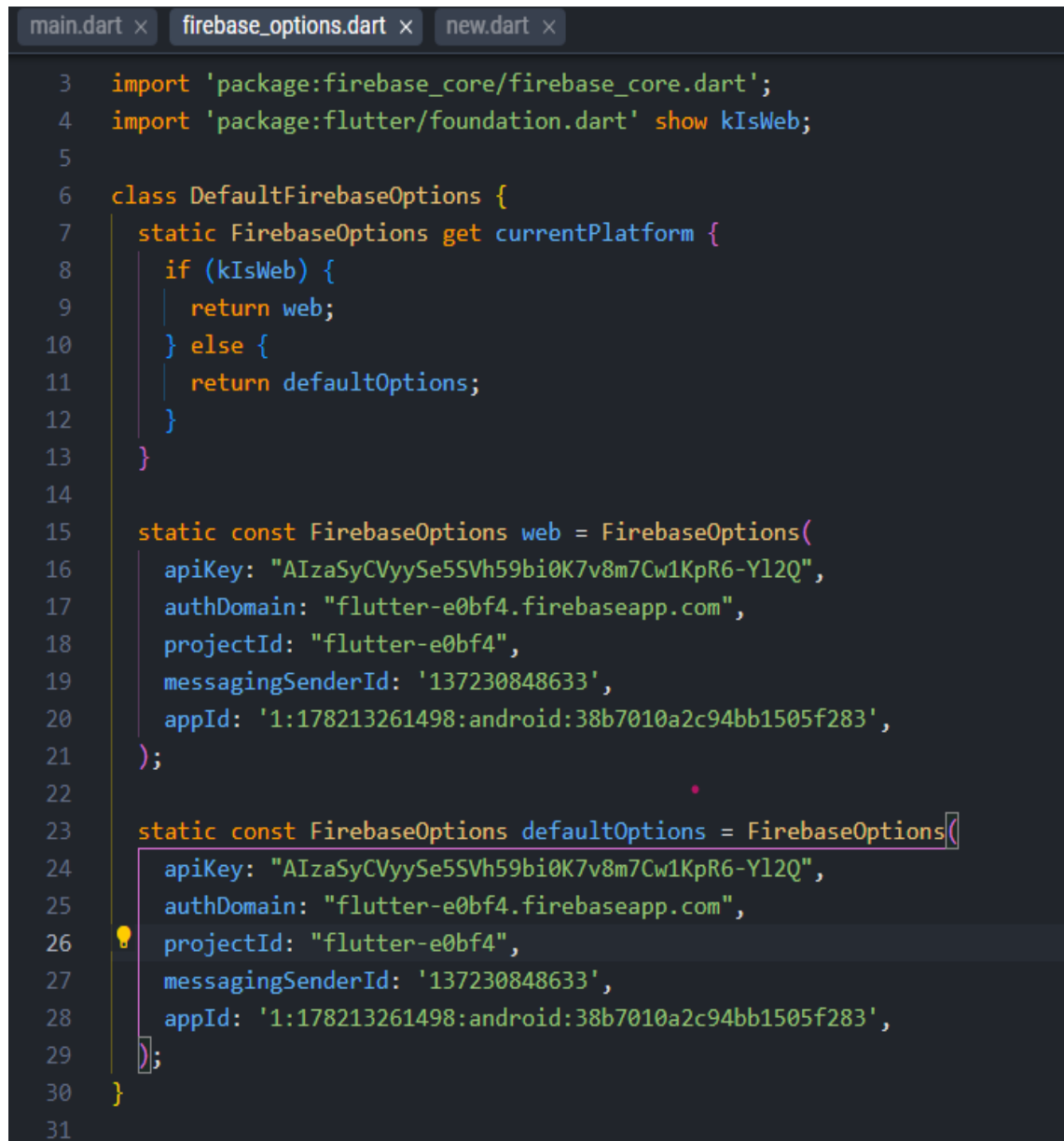
```



BACKEND KODLARI:

Firebase ve uygulama bağlantısını oluşturmak için Firebase Veri tabanımın ayarlar kısmında bulunan google_services.json dosyasını indirip uygulamanın içerisindeki App klasörü içerisine ekledim. Daha sonra Firebase_options.dart kod dosyası içerisine Firebase Veri tabanındaki bağlantı bilgilerini ve apiyi ekledim ve uygulama ile Firebase arasındaki bağlantıyı oluşturdum.

Firebase_options.dart



```
main.dart x  firebase_options.dart x  new.dart x

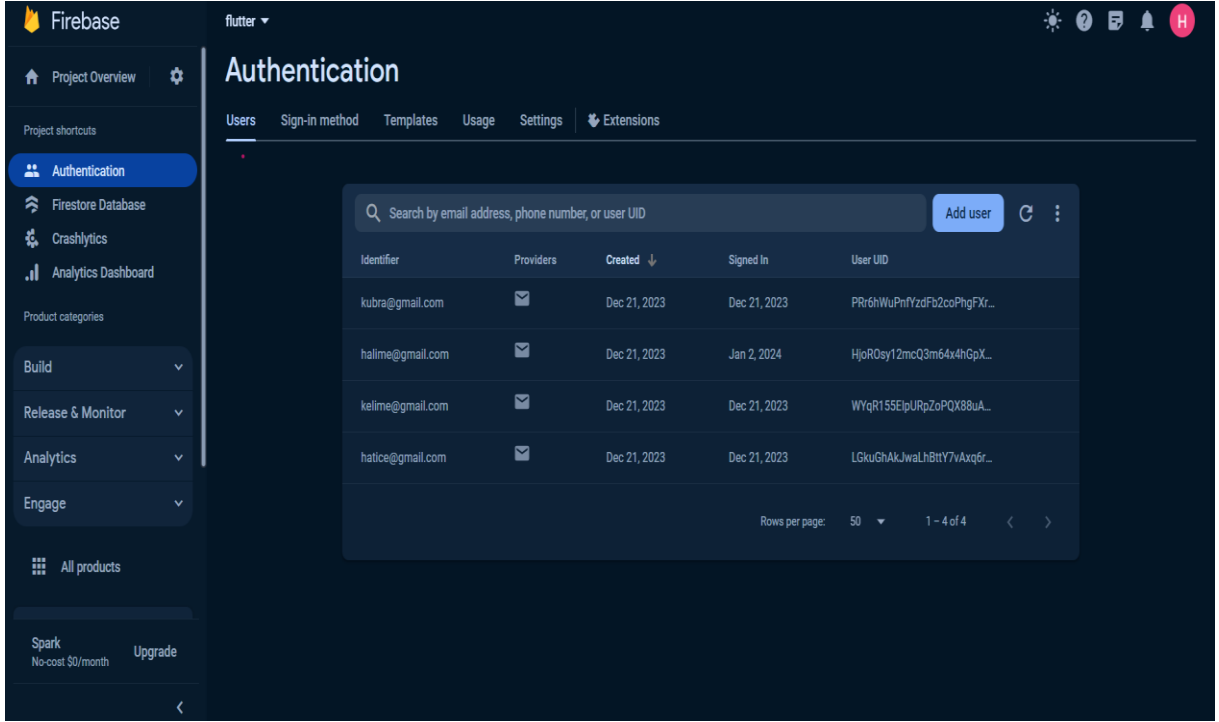
3  import 'package:firebase_core/firebase_core.dart';
4  import 'package:flutter/foundation.dart' show kIsWeb;
5
6  class DefaultFirebaseOptions {
7      static FirebaseOptions get currentPlatform {
8          if (kIsWeb) {
9              return web;
10         } else {
11             return defaultOptions;
12         }
13     }
14
15     static const FirebaseOptions web = FirebaseOptions(
16         apiKey: "AIzaSyCVyySe5SVh59bi0K7v8m7Cw1KpR6-Yl2Q",
17         authDomain: "flutter-e0bf4.firebaseio.com",
18         projectId: "flutter-e0bf4",
19         messagingSenderId: '137230848633',
20         appId: '1:178213261498:android:38b7010a2c94bb1505f283',
21     );
22
23     static const FirebaseOptions defaultOptions = FirebaseOptions(
24         apiKey: "AIzaSyCVyySe5SVh59bi0K7v8m7Cw1KpR6-Yl2Q",
25         authDomain: "flutter-e0bf4.firebaseio.com",
26         projectId: "flutter-e0bf4",
27         messagingSenderId: '137230848633',
28         appId: '1:178213261498:android:38b7010a2c94bb1505f283',
29     );
30 }
31
```

Kullanıcıların giriş sayfasında kayıt sayfasından olduğu gibi Firebase Veri tabanı bağlantısında birden fazla bağlantı oluşmaması için giriş ve kayıt işlemleri için ayrı fonksiyonlar oluşturdum. Ve bu fonksiyonları Auth_Service.dart kod dosyasında birleştirdim. Bu fonksiyonu giriş sayfasında import ettim.

Auth_Service.dart

```
main.dart x  firebase_options.dart x  new.dart x  Auth_Service.dart x
1  import 'package:firebase_auth/firebase_auth.dart';
2
3  class AuthService {
4    final FirebaseAuth _auth = FirebaseAuth.instance;
5
6    // Kullanıcı giriş işlemi
7    Future<User?> signIn(String email, String password) async {
8      try {
9        UserCredential userCredential = await _auth.signInWithEmailAndPassword(
10          email: email,
11          password: password,
12        );
13        return userCredential.user;
14      } on FirebaseAuthException catch (e) {
15        // Giriş hatası durumunda hata mesajını yazdır
16        print("Giriş hatası: ${e.message}");
17        return null;
18      }
19    }
20
21    // Kullanıcı kaydı işlemi
22    Future<User?> register(String email, String password) async {
23      try {
24        UserCredential userCredential =
25          await _auth.createUserWithEmailAndPassword(
26            email: email,
27            password: password,
28          );
29        return userCredential.user;
30      } on FirebaseAuthException catch (e) {
31        // Kullanıcı kaydı hatası durumunda hata mesajını yazdır
32        print("Kullanıcı kaydı hatası: ${e.message}");
33        return null;
34      }
35    }
36
37    // Kullanıcı çıkış işlemi
38    Future<void> signOut() async {
39      await _auth.signOut();
40    }
41  }
```

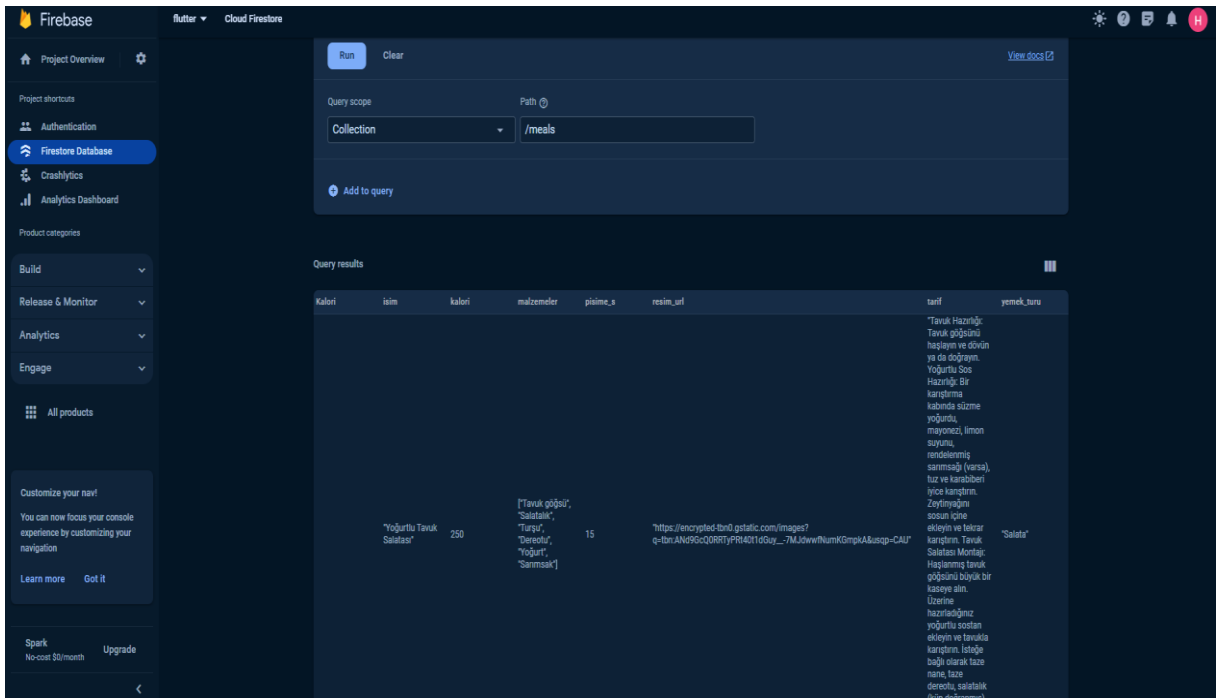
Users(Kullanıcılar) Tablosu:



The screenshot shows the Firebase Authentication console. The left sidebar contains navigation links for Project Overview, Authentication, Firestore Database, Crashlytics, and Analytics Dashboard. The main area displays the 'Users' tab with a search bar and a table of users.

| Identifier | Providers | Created | Signed In | User UID |
|------------------|-----------|--------------|--------------|-----------------------------|
| kubra@gmail.com | | Dec 21, 2023 | Dec 21, 2023 | PRr6hWuPnfyZdFbZcoPhgFXr... |
| halime@gmail.com | | Dec 21, 2023 | Jan 2, 2024 | HjoR0sy12mcQ3m64x4hGpX... |
| kelime@gmail.com | | Dec 21, 2023 | Dec 21, 2023 | WYqR15SElpURpZoPQX88uA... |
| hatice@gmail.com | | Dec 21, 2023 | Dec 21, 2023 | LGkuGhAkJwaLh8ttY7vAxdqr... |

Meals(Yemekler) Tablosu:



The screenshot shows the Firebase Firestore console. The left sidebar contains navigation links for Project Overview, Authentication, Firestore Database, Crashlytics, and Analytics Dashboard. The main area displays the 'Firestore Database' tab with a query editor and a table of query results.

| Kalori | isim | kalori | malzemeler | pisime_s | resim_url | tarif | yemek_turu |
|--------|---------------------------|--------|---|----------|--|---|------------|
| | "Yoğurtlu Tavuk Salatası" | 250 | ["Tavuk göğsü", "Salatalık", "Taze", "Dereotu", "Sarmasak"] | 15 | "https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9cCQ08RTYpRI401dGy...7MJoWwNnmK6mpkAUsqg-CAU/" | "Tavuk Hazırlık: Tavuk göğsünü haşlayın ve dilimleyin ya da doğrayın. Yoğurtlu Sos Hazırlık: Bir karışım kabında süzme yoğurt, mayonez, limon suyu, rendelenmiş sarımsak (varsa), tuz ve karabiberi iyice karıştırın. Zeytinyağını sosun içine ekleyin ve tatlar karıştıkça Tavuk Salatası Montajı: Haşlanmış tavuk göğsünü büyük bir kaseye alın. Üzerine hazırladığınız yoğurtlu sosu ekleyin ve tavukla karıştırın. İsteğe bağlı olarak taze nane, taze dereotu, salatalık (küp doğranmış) | "Salata" |

| | | | | | | |
|-----------------|-----|---|----|---|---|---------|
| "Magnolya" | 250 | ["Şeker", "Un", "Krema", "Vanilya", "Findik", "Çilek"] | 60 | "https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcS7A48apBtlLoVczv-2qEzjNCT_5rFJ-sR42A8usqp=CAU" | <p>tepaşye dokun ve önceden ısıtılmış 180 derece fırında pişirin. Kek pişerken, karamel sos için şeker ile suyu bir tencerede koyu kıvama gelene kadar karıştırarak kaynatın. Kaynamış karamel sosu, fırından çıkan sıcak kekin üzerine dökün. Kek soğuyana kadar bekleyin. Kremayı hazırlamak için süt, toz şeker, un, nişasta ve yumurta sarılarını bir tencerede karıştırarak pişirin. Pişen kremayı ocaktan alıp içine vanilini ekleyin ve karıştırın. Soğuyan kekin üzerine hazırlanan kremayı sürün. Tatlıyı buzdolabında bir süre dinlendirip servis yapın."</p> <p>"Kabakları yıkayıp kabuklarını soyun ve küp şeklinde doğrayın. Soğanı doğrayın veya rendeleyin. Bir tencerede tereyağını veya sıvı yağı eritin. Soğanı ekleyip pembeleşene kadar kavurun. Kabakları ekleyin ve birkaç dakika karıştırarak kavurun. Üzerine unu ekleyip kokusu çıkana kadar kısık ateşte karıştırarak kavurun. Tıvuk veya sebze suyunu yavaşça ekleyerek karıştırın. Kaynamaya başladığında kısık</p> | "Tatlı" |
| "Kabak Çorbası" | 400 | ["Kabak", "Un", "Tereyağı", "Sıvıyağ", "Süt", "Derestur", "Nane"] | 25 | "https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcTBNrFqVEnZC0aUp47PpKtmp3Ty2M9UbUXA9A8usqp=CAU" | <p>"Çorba"</p> | |

| | | | | | |
|-------------------------|-----|---|----|---|---|
| | | | | | isteğe bağlı olarak üzerine ekstra tuz ve baharat ekleyebilirsiniz. " |
| "Tavuklu Sultan Kebabı" | 300 | ["Yufka", "Tavuk göğsü", "Tavuk pirzola", "Soğan", "Bezelye", "Patlıcan", "Biber", "Un", "Sut"] | 50 | "https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcQKfpsDUuSJpB78akEHLv6jBr6VVR2K_L6w&usqp=CAU" | "Tavuklu harcı için tavukları, rendelenmiş soğanı, sarımsağı, sıvı yağ, tuz, karabiber, pul biber ve kimyonu bir kaba alın. Karşımı iyice yoğurun ve buzdolabında en az 1 saat dinlendirin. Bulgur harcı için ince bulguru bir kaba alın. Üzerine sıvı yağ, sıcak su ve tuz ekleyip karıştırın. Karşımı bir streç film ile üzerini kapatıp dinlendirin. Tavuk harcından ceviz büyüklüğünde parçalar alıp elinizle yuvarlayın. Ortasına bulgur harcından bir miktar koyup eti bulgurla sararak köfte şekli verin. Tüm köfteleri hazırlayın ve yağlı kağıt serili fırın tepsiğine düzin. Üzerlerine sıvı yağ, eritilmiş tereyağı ve sıcak su karşımını sürün. Önceden ısıtılmış 200 derece fırında, üzeri iyice kızarana kadar pişirin (yaklaşık 30-40 dakika). Sıcak servis yapın. " |
| | | | | | "Sıcacetti |

| | | | | | |
|--------------|-----|---|----|---|---|
| "Spagetti" | 500 | ["Spagetti makarna", "Zeytinyağı", "Domates", "Fesleğen", "Sarımsak"] | 30 | "https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcR9q_f1eltiqHVoH3rIdkaFfJZcM0xgx_qEiw&usqp=CAU" | aceteleyin. Rendelenmiş domatesi tavaya ekleyin, karıştırın. Tuz, karabiber ve isteğe bağlı olarak pul biber ekleyin. Domates sosu kaynamaya başladıktan sonra ateşi kısın ve sosu 10-15 dakika kadar pişirin. Haşlanmış spagetti makarnayı sosun içine ekleyip karıştırın, birkaç dakika daha pişirin. Servis tabaklarına almadan önce üzerine taze veya kuruy fesleğen serpiştirin. Rendelenmiş parmesan peyniri ile servis edin." |
| "Profiterol" | 600 | ["Yumurta", "Süt", "Un", "Tereyağı", "Şeker", "Vanilya"] | 40 | "https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcRJ4F4gVhGKZtqc9YkfkkbKQ7EpL9nAG00g&usqp=CAU" | "Profiterol Keki: Tereyağı veya margarin su ile bir tencerede ertin. Kaynamaya başlayan karşıma unu ekleyin ve karıştırarak pişirin. Karşım hamur halini alana kadar karıştırın ve ocaktan alın. Hamur biraz soğuduktan sonra içine tek tek yumurtaları ekleyip karıştırın. Hazırlanan hamuru sıkma torbasına alın ve yağlı kağıt serili tepsiye küçük parçalar halinde sıkın. Önceden ısıtılmış 200 derece fırında üzeri iyice kızarana kadar pişirin. Krema: Süt, un, niğasta, yumurta sarıları ve toz şekeri bir tencerede karıştırarak oluşturun. |

