



T.C.
BİLECİK ŞEYH EDEBALI ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

ONLİNE BAKKAL
MAHAMAT KABIR SOULEYMAN ISSEIN
BİTİRME ÇALIŞMASI

DANISMAN: Dr. Öğr. Üyesi Salim CEYHAN

BİLECİK
14 Ocak 2024



T.C.
BİLECİK ŞEYH EDEBALI ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

ONLİNE BAKKAL
MAHAMAT KABIR SOULEYMAN ISSEIN
BİTİRME ÇALIŞMASI

DANIŞMANI :Dr. Öğr. Üyesi Salim CEYHAN

BİLECİK
14 Ocak 2024

BİLDİRİM

Bu çalışmada bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edildiğini ve yazım kurallarına uygun olarak hazırlanan bu çalışmada bana ait olmayan her türlü ifade ve bilginin kaynağına eksiksiz atıf yapıldığını bildiririm.

DECLARATION

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all materials and results that are not original to this work.

İmza

AD-SOYAD

Tarih: 14 Ocak 2024

ÖZET

BİTİRME ÇALIŞMASI

BİTİRME TEZİNİN ADI

AD-SOYAD

Bilecik Şeyh Edebali Üniversitesi
Mühendislik Fakültesi
Bilgisayar Mühendisliği Bölümü

Danışman: Ünvan AD-SOYAD

2024, 36 Sayfa

Jüri Üyeleri

İmza

.....
.....
.....

.....
.....
.....

Türkçe özetin yazılacağı yer.

ABSTRACT

THESIS

TEZİN İNGİLİZCE ADI

AD-SOYAD

**Bilecik Şeyh Edebali University
Engineering Faculty
Department of Computer Engineering**

Advisor : İngilizce ünvan AD-SOYAD

2024, 36 Pages

Jury

Sign

.....
.....
.....

.....
.....
.....

Where The Thesis Is Written In English.

ÖNSÖZ

Bitirme çalışmamın başından sonuna kadar emeği geçen ve beni bu konuya yönlendiren saygı değer hocam ve danışmanım Sayın Dr. Öğr. Üyesi Salim CEYHAN 'a tüm katkılarından ve hiç eksiltmediği desteğinden dolayı teşekkür ederim.

MAHAMAT KABIR SOULEYMAN ISSEIN

14 Ocak 2024

İÇİNDEKİLER

| | |
|---|-----------|
| ÖNSÖZ | v |
| ŞEKİLLER TABLOSU | viii |
| 1 GİRİŞ | 1 |
| 1.1 Projenin İçeriği | 2 |
| 1.2 Öğrenciler İçin Projenin Önemi | 2 |
| 1.3 Benzer Projeler | 3 |
| 1.3.1 Instacart | 3 |
| 1.3.2 Amazon Fresh | 3 |
| 1.3.3 Walmart Grocery | 4 |
| 1.3.4 Projenin github linki | 4 |
| 2 KULLANILAN YAZILIMLAR | 5 |
| 2.1 Flutter | 5 |
| 2.2 Firebase | 5 |
| 2.2.1 Mobil uygulama için Firebase entegrasyonu | 6 |
| 2.2.2 Firebase Yönetici Paneli Entegrasyonu | 9 |
| 3 Web Uygulama (Admin Paneli) | 11 |
| 3.1 Firebase'e Ürün Ekleme, Güncelleme ve Silme | 11 |
| 4 Mobil Uygulama Geliştirme (Flutter) | 13 |
| 4.1 Firebase Cloud Functions ve Firestore Entegrasyonu: | 13 |
| 4.2 Kullanıcı İnteraktifliği | 17 |
| 4.3 Kullanılan Paketler | 19 |
| 4.3.1 shared Preferences | 20 |
| 4.3.2 Provider | 20 |
| 4.3.3 google sign in | 20 |
| 4.3.4 Firebase | 20 |
| 4.3.5 Firebase- Auth | 21 |
| 4.3.6 Firebase-core | 21 |

| | | |
|----------|--------------------------------|-----------|
| 4.3.7 | Cloud -Firestore | 21 |
| 4.3.8 | Firebase-Storage | 21 |
| 4.3.9 | Flutter-Spinki | 22 |
| 4.3.10 | Fluttertoast | 22 |
| 4.3.11 | Uuid | 22 |
| 4.3.12 | Flutter-Stripe | 22 |
| 4.3.13 | Http | 23 |
| 4.3.14 | image picker for web | 23 |
| 5 | SONUÇLAR VE ÖNERİLER | 24 |
| 6 | EKLER | 25 |
| | KAYNAKLAR | 26 |
| | ÖZGEÇMİŞ | 27 |

ŞEKİLLER TABLOSU

| | | |
|-----------|--|----|
| Şekil 2.1 | Firebase İntegration | 6 |
| Şekil 2.2 | Firebase Oluşturma | 7 |
| Şekil 2.3 | google-service | 8 |
| Şekil 2.4 | Firebase SDK | 8 |
| Şekil 2.5 | Firebase Web Panel Oluşturma | 9 |
| Şekil 3.1 | Firebase Ekleme | 12 |
| Şekil 3.2 | Firebase Ekleme | 13 |
| Şekil 4.1 | Login in Sign Up | 14 |
| Şekil 4.2 | Forget-Password | 15 |
| Şekil 4.3 | AnaSayfa | 16 |
| Şekil 4.4 | Categories | 17 |
| Şekil 4.5 | Orders-Screen | 18 |
| Şekil 4.6 | Kullanıcı-Ekran | 19 |

1 GİRİŞ

Bakkal alışverişini dijitalleştirmek için Flutter ile geliştirilen yenilikçi bir uygulamadır. Teknolojinin sunduğu imkanlarla, alışveriş deneyimi de dönüşüme uğruyor. Geleneksel bakkal alışverişini dijitalleştiren ve kullanıcıların elini rahatlatan yeni bir uygulama geliştirme süreci, Flutter çatısı altında gerçekleştirildi. Bu uygulama, kullanıcıların taleplerine odaklanarak alışverişin sadece bir tık uzaklıkta olmasını sağlıyor.

Flutter'ın esnekliği ve kullanım kolaylığı, bakkal alışverişini çağın gereksinimlerine uygun hale getirmek için ideal bir araç haline gelmiştir. Bu projenin amacı, kullanıcıların alışveriş yaparken daha kolay, daha hızlı ve daha kişiselleştirilmiş bir deneyim yaşamalarını sağlamaktır.

Bu uygulamanın temel özelliklerinden biri, kullanıcıların istedikleri miktarlarda ürünleri seçebilme özgürlüğüdür. *Kg* cinsinden sipariş verme seçeneği, kullanıcıların taleplerini doğrudan belirleyebilmelerini sağlayarak alışveriş deneyimini kişiselleştiriyor.

Kullanıcılar, aradıkları ürünleri hızlıca bulabilmek için geliştirilen arama ve filtreleme özellikleri sayesinde istedikleri ürünleri kolayca seçebiliyorlar. Ayrıca, favori veya sıkça alınan ürünleri hatırlatma veya kayıt altına alma imkanı da uygulamanın sunduğu pratik özelliklerden biri olarak öne çıkıyor. Bu uygulama aynı zamanda kullanıcıların siparişlerini yönetmelerini, farklı ödeme seçeneklerini kullanmalarını ve teslimat sürecini izlemelerini sağlayarak alışveriş deneyimini daha da kolaylaştırıyor.

Teknolojinin sunduğu bu olanaklarla geleneksel alışveriş deneyimini dijital platformlara taşıyan bu uygulama, kullanıcıların hayatını kolaylaştırmayı ve alışveriş alışkanlıklarını modernize etmeyi hedefliyor. Geliştirilen bu bakkal alışveriş uygulaması, kullanıcıların günlük hayatlarını kolaylaştırmayı ve alışveriş deneyimini daha keyifli hale getirmeyi amaçlıyor. Bu bağlamda, uygulamanın sunduğu olanakların detaylı bir şekilde incelenmesi ve kullanıcı geri bildirimleriyle şekillenmesi, gelecekte daha da kullanıcı dostu ve işlevsel bir platform haline gelmesine katkı sağlayacaktır.

1.1 Projenin İeriđi

Bu uygulama, kullanıcıların alışveriş yaparken daha fazla esneklik, kolaylık ve kişiselleştirme imkanı bulmalarını hedefliyor.

Temel özellikler şu şekilde ayrılabilir:

Ürün Seçimi ve Miktarı: Kullanıcıların istedikleri ürünleri seçme özgürlüğü bulunmakla birlikte, bu ürünleri istedikleri miktarlarda alma şansına sahipler. Örneğin, kg cinsinden sipariş verebilme seçeneđi, kullanıcıların taleplerine uygun alışveriş yapmalarını sağlıyor.

Arama ve Filtreleme: Kullanıcılar, aradıkları ürünleri hızlıca bulabilmek için geliştirilen arama ve filtreleme özelliklerini kullanarak istedikleri ürünlere kolayca ulaşabiliyorlar. Bu özellik, alışveriş sürecini daha verimli hale getiriyor.

Kişiselleştirme ve Favoriler: Kullanıcılar, favori ürünleri belirleyebilir ve sıkça alınan ürünleri hatırlatabilir veya kayıt altına alabilirler. Bu da alışveriş deneyimini daha özelleştirilmiş bir hale getiriyor.

Sipariş Yönetimi ve Ödeme Seçenekleri: Uygulama, kullanıcıların siparişlerini kolayca yönetmelerine ve farklı ödeme seçeneklerini kullanmalarına olanak tanıyor. Ayrıca, teslimat sürecini izleyebilme imkanı da sunuluyor.

Geliştirme ve Geri Bildirimler: Kullanıcı geri bildirimleri, uygulamanın sürekli olarak geliştirilmesinde önemli bir rol oynuyor. Kullanıcıların ihtiyaçlarına ve geri bildirimlerine göre uygulama sürekli olarak iyileştirilmeye açık tutuluyor.

1.2 Öğrenciler İçin Projenin Önemi

öğrenciler için bu tür bir proje birçok açıdan önemli olabilir:

Teknolojiyi Kullanarak İnovasyon: Öğrenciler, Flutter gibi yeni teknolojileri kullanarak yenilikçi bir proje geliştirme fırsatı bulabilirler. Bu, teknolojiyi gerçek hayatta kullanarak uygulamalı deneyim kazanmalarını sağlar.

Problem Çözme Yeteneği Geliştirme: Bu tür bir projenin geliştirilmesi, gerçek bir ihtiyaca yönelik çözüm üretmeyi gerektirir. Öğrenciler, kullanıcıların ihtiyaçlarını anlama ve bunlara uygun çözümler üretme konusunda deneyim kazanabilirler.

Ekip Çalışması ve İletişim: Proje genellikle bir ekip tarafından yürütülür. Öğrenciler, birlikte çalışarak, fikirleri paylaşarak ve görevleri dağıtarak ekip çalışmasının önemini öğrenirler.

Kullanıcı Deneyimi ve Geri Bildirimler: Kullanıcı odaklı bir proje geliştirmek, kullanıcıların ihtiyaçlarını anlamak ve onlardan geri bildirim almak demektir. Öğrenciler, bu süreçte kullanıcı deneyimi ve geri bildirimlerin önemini kavrayabilirler.

Yaratıcılık ve Kişisel Gelişim: Projeyi geliştirirken, öğrenciler yaratıcı düşünme becerilerini kullanır, sorunları farklı açılardan ele almayı öğrenir ve kişisel gelişimlerine katkı sağlarlar.

Kariyer Fırsatları ve Portföy Oluşturma: Bu tür bir proje, öğrencilerin gelecekteki kariyerleri için önemli bir referans olabilir. Ayrıca, geliştirilen uygulama veya proje, öğrencilerin portföylerini zenginleştirebilir.

1.3 Benzer Projeler

bakkal alışverişi veya market alışverişi odaklı mobil uygulamalar birkaç farklı platformda benzer şekilde geliştirilmiştir. İşte benzer projelere birkaç örnek:

1.3.1 Instacart

Bu uygulama, kullanıcıların yerel marketlerden alışveriş yapmalarını sağlar. Kullanıcılar, uygulama üzerinden ürünleri seçebilir ve siparişlerini oluşturabilirler. Bu siparişler daha sonra belirlenen zaman diliminde kullanıcılara teslim edilir. [1].

1.3.2 Amazon Fresh

Amazon'un sunduğu bu hizmet, kullanıcıların market alışverişini çevrimiçi platformda gerçekleştirmesine olanak tanır. Kullanıcılar, geniş bir ürün yelpazesi arasından seçim ya-

pabilir ve Amazon'un teslimat hizmetiyle istedikleri ürünleri kapılarına teslim alabilirler.

1.3.3 Walmart Grocery

Walmart, kullanıcıların market alışverişini çevrimiçi olarak yapabilmelerini sağlayan bir platform sunar. Kullanıcılar, uygulama veya web sitesi üzerinden ürünleri seçebilir ve belirli bir saat aralığında teslimat veya mağazadan almayı tercih edebilirler.

1.3.4 Projenin github linki

Proje kaynak kod olarak aşağıdaki github linkinde sunulmuştur.

2 KULLANILAN YAZILIMLAR

2.1 Flutter

Bu projede Google tarafından geliştirilen Flutter kullanıldı. Flutter, Google'ın mobil, masaüstü ve web uygulamaları geliştirmek için sunmuş olduğu açık kaynak kodlu bir yazılımdır. geliştirme aracıdır. Flutter uygulamaları yine Google tarafından geliştirilen ve nesne tabanlı bir dil olan Dart programlama dili ile yazılmaktadır. Aşağıdaki avantajları sebebiyle Flutter tercih edildi. [1]

- Android ve iOS uygulamaları için aynı kod kullanıldığından geliştirme hızı yüksektir. Flutter tekil kod tabanına sahiptir. Bu durum uygulama geliştirme maliyetini olabildiğince düşürür. Ayrıca uygulamanın pazara sunulma süresini hızlandırır.
- Flutter gerçek zamanlı uygulamalara eşit performans sunar. Diğer yazılım geliştirme yapılarına göre uygulamaların yerel bileşenlerle etkileşim kurması zorunlu değildir. Söz konusu etkileşim köprüler aracılığıyla gerçekleştirilir ve bunlar performans sorunlarına yol açabilirler.
- Flutter kullanan geliştiriciler yapılan tüm değişiklikleri gerçek zamanlı olarak görme fırsatına sahiptirler. Uygulama çalışırken değiştirilen kod hemen yansır ve yeniden başlatma gerekli değildir. Hata düzeltme ve özellik eklemek oldukça basittir.

2.2 Firebase

Firebase, Google tarafından geliştirilen ve uygulama geliştirmeyi kolaylaştıran çeşitli hizmetler sunan bir platformdur. Mobil ve web uygulamalarının oluşturulması, yönetilmesi ve geliştirilmesi için bir dizi araç sunar.

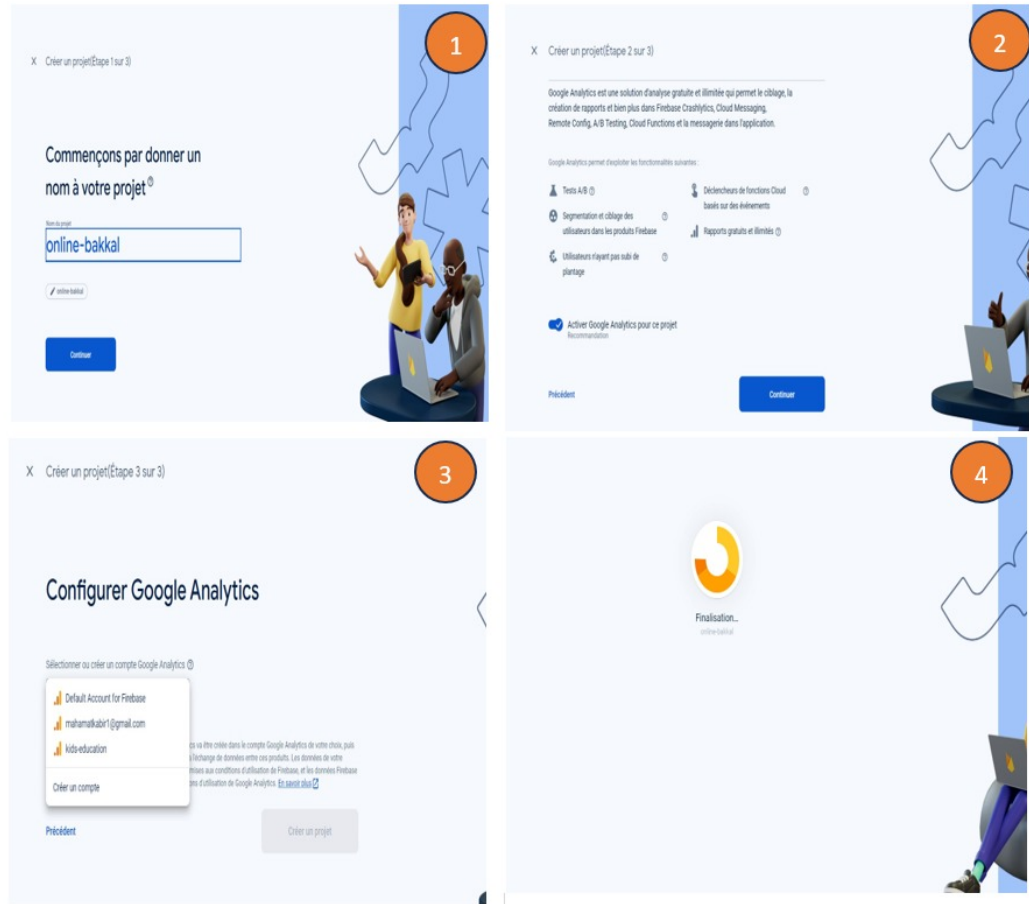


Şekil 2.1: Firebase İntegration

2.2.1 Mobil uygulama için Firebase entegrasyonu

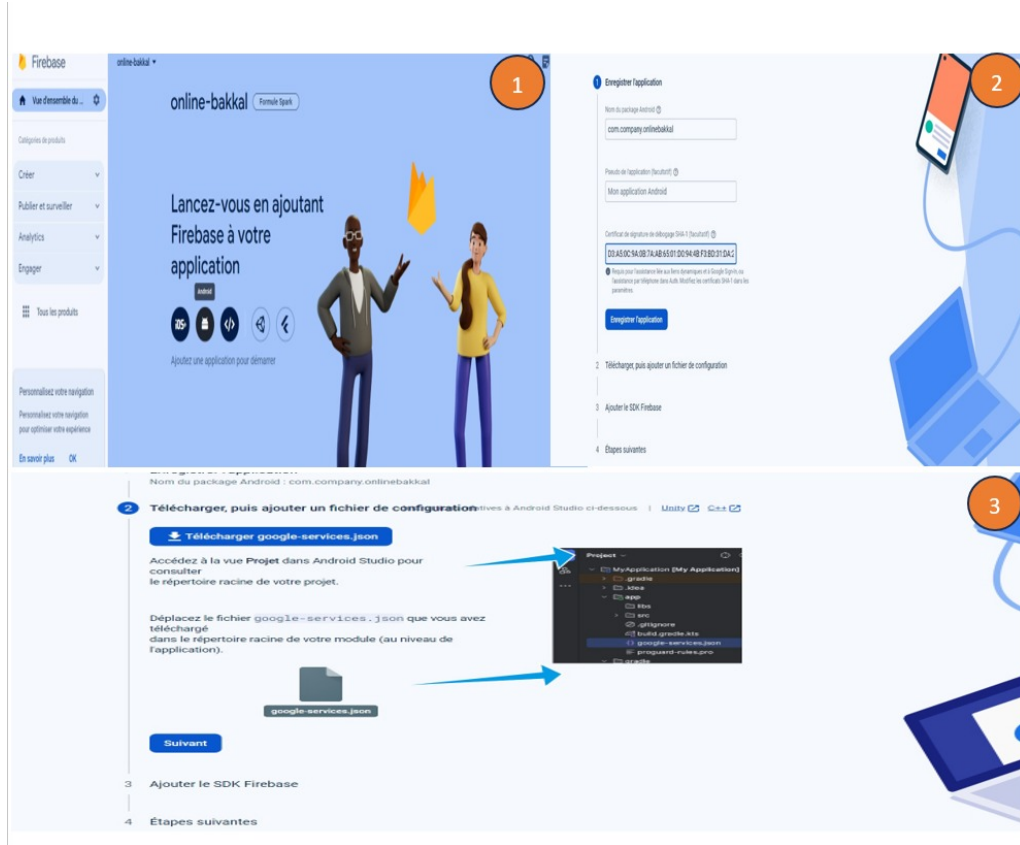
Uygulamamız Firebase'e bağlamak için genel adımlar kullanılmıştır:

- Firebase Projesi Oluşturmak : Firebase konsola giderek projemiz oluşturabiliriz.Aşağıdaki örnek takip ederek



Şekil 2.2: Firebase Oluşturma

- Uygulamamız Firebase'e Ekledik : Firebase'e platformunda Android seçerek uygulamamızı ekledik. Platforma özgü ayrıntıları ekleyin (örneğin, Android için google-services.json, dosyalarını eklemiştik).



Şekil 2.3: google-service

- Firebase SDK'sını Uygulamamıza integre ettik: Her platform için (Android, iOS, Web), Firebase SDK'sını uygulamanıza entegre ettik. Bu, bağımlılıkların eklenmesi, yapılandırma dosyalarının eklenmesi ve SDK'nın başlatılmasını içermektedir.



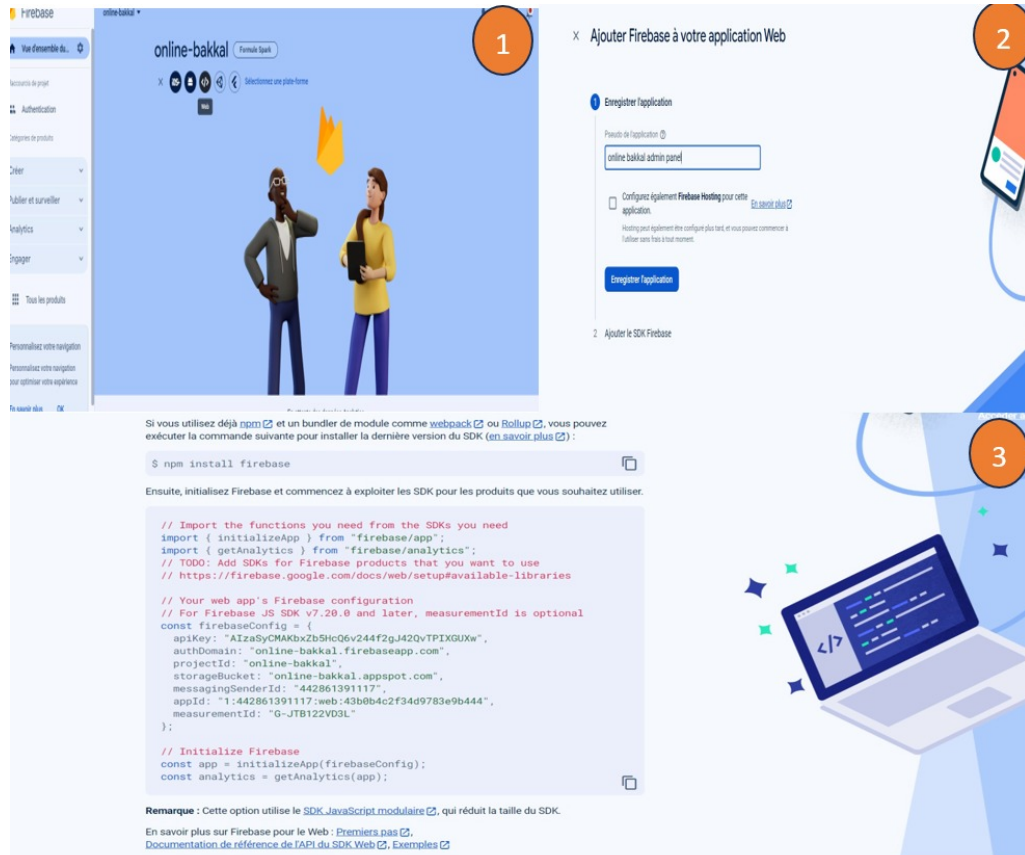
Şekil 2.4: Firebase SDK

- **Firestore Initialization:** firestore projemizde kullanmak için firebase-core paketi kullanılarak main klasorumuzda aşağıdaki kod ekleyerek çalıştırdık .

```
1 import 'package:firebase_core/firebase_core.dart';
2 void main() async {
3   WidgetsFlutterBinding.ensureInitialized();
4   await Firebase.initializeApp();
5   runApp(const MyApp());
6 }
```

2.2.2 Firebase Yönetici Paneli Entegrasyonu

- **Firestore Projesi Oluşturma:** Firestore Konsolu'ndan yeni bir Firestore projesi oluşturun veya var olan bir proje kullanın (<https://console.firebase.google.com/>). Firestore projesinizin, gereksinimlerinize göre Kimlik Doğrulama, Gerçek Zamanlı Veritabanı, Firestore veya Depolama gibi gerekli hizmetlerle yapılandırıldığından emin olun.



Şekil 2.5: Firebase Web Panel Oluşturma

- Web Uygulama Yapılandırma: Firebase'i web uygulamanıza ekleyerek başlayın. Genellikle bunu, HTML dosyanıza bir script etiketi ekleyerek veya npm/yarn kullanarak paket yönetimi ile yapabilirsiniz.

```
1 <script defer src="https://www.gstatic.com/firebasejs/8.10.1/  
  firebase-app.js"></script>  
2 <script defer src="https://www.gstatic.com/firebasejs/8.10.1/  
  firebase-auth.js"></script>  
3 <script defer src="https://www.gstatic.com/firebasejs/8.10.1/  
  firebase-firestore.js"></script>  
4 <script defer src="https://www.gstatic.com/firebasejs/8.10.1/  
  firebase-storage.js"></script>  
5 <script defer src="https://www.gstatic.com/firebasejs/8.10.1/  
  firebase-analytics.js"></script>
```

- Firebase'i Yönetici Panelinde Başlatma: Web yönetici panelinizin JavaScript kodunda, Firebase'i Firebase projesinin yapılandırmasıyla başlatın.

```
1  
2 options: const FirebaseOptions (  
3     apiKey: "AIzaSyDxUEEYqnwni5LV-CoC11sUM97s2FFUHM0",  
4     authDomain: "gracery-app.firebaseio.com",  
5     projectId: "gracery-app",  
6     storageBucket: "gracery-app.appspot.com",  
7     messagingSenderId: "515831734983",  
8     appId: "1:515831734983:web:dd3fafad03818328f554e2",  
9     measurementId: "G-1WR8XNYQMC" ) ;
```

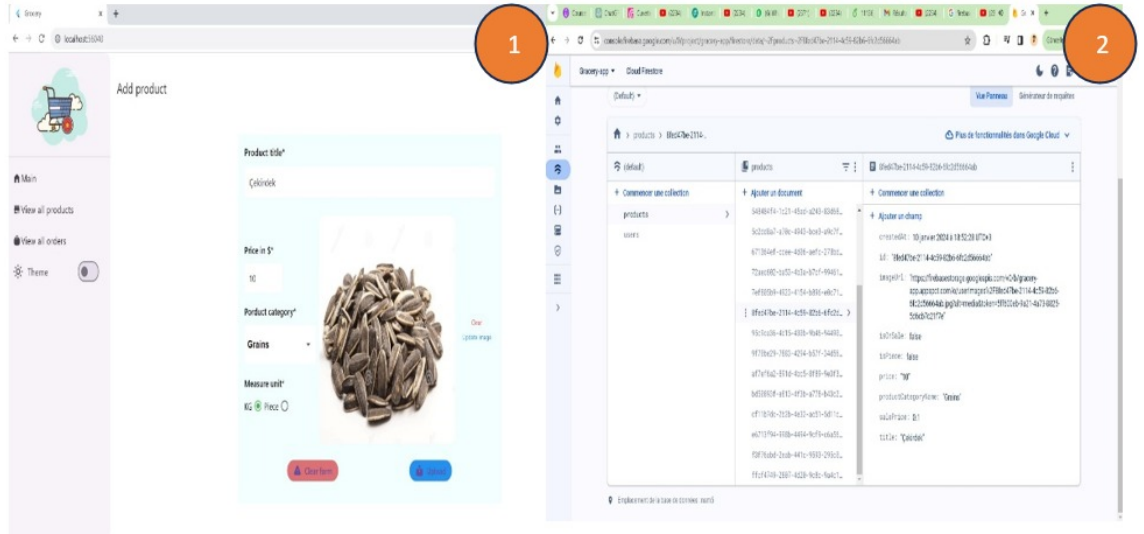
3 Web Uygulama (Admin Paneli)

3.1 Firebase’e Ürün Ekleme, Güncelleme ve Silme

Projemiz tek bir kullanıcı tarafından web uygulamamız kullanılmıştır bunun için bu süreçte authentication kullanılmamıştır.

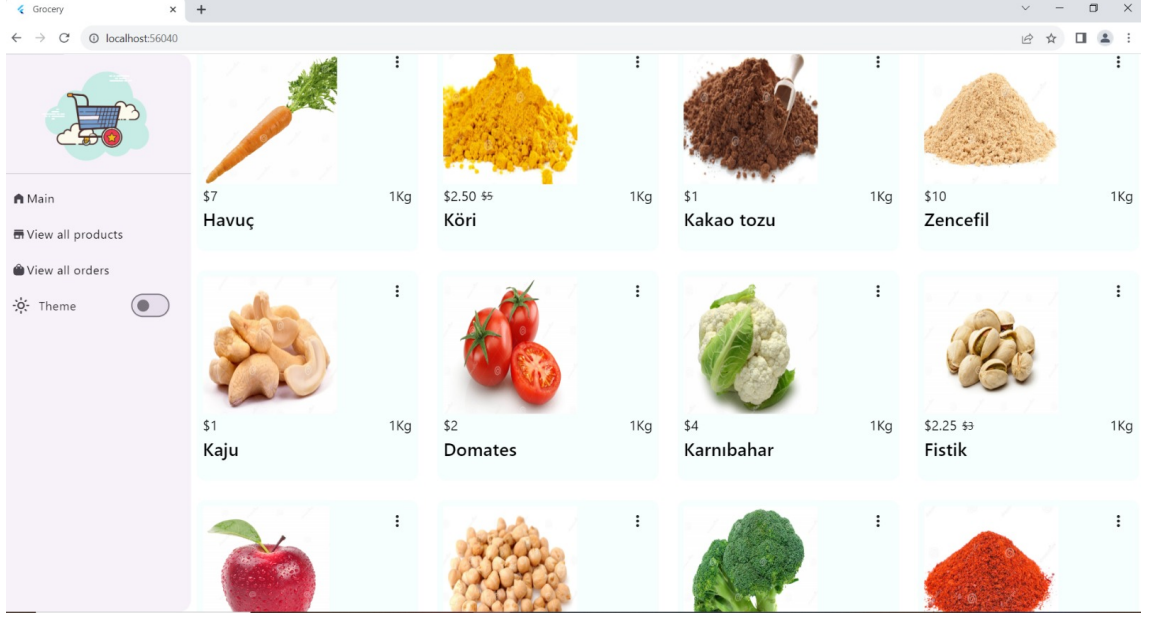
- Ürün Ekleme: sayfamızda main kısmında sağ tarafı bulunan Add product tıklayarak açılacak sayfadan ekleme yapabiliriz. Uygulama, ürün bilgilerini bir form aracılığıyla kullanıcıdan alır ve bu bilgileri bir Firebase Firestore veritabanına yükler. Form, ürün başlığı, fiyatı, kategorisi ve ölçü birimi gibi detayları içerir. Kullanıcı, bir resim seçme işlevselliği aracılığıyla ürüne bir görsel ekleyebilir. Kod, resim seçme işlemlerini hem mobil cihazlar hem de web platformları için uygun şekilde yönetir. Firebase hizmetleri kullanılarak, yüklenen resim Firebase Storage’a depolanır ve ürün bilgileri Firestore veritabanına kaydedilir. Ayrıca, kullanıcıya bilgilendirici toast mesajları ve hata iletileri göstermek için Fluttertoast ve özel bir hata iletişim kutusu kullanılır.

```
1 await FirebaseFirestore.instance.collection('products').doc(  
    _uuid).set({  
2         'id': _uuid,  
3         'title': _titleController.text,  
4         'price': _priceController.text,  
5         'salePrice': 0.1,  
6         'imageUrl': imageUrl,  
7         'productCategoryName': _catValue,  
8         'isOnSale': false,  
9         'isPiece': isPiece,  
10        'createdAt': Timestamp.now(),  
11    });  
12    _clearForm();  
13    Fluttertoast.showToast(  
14        msg: "Product uploaded succefully",  
15        toastLength: Toast.LENGTH_LONG,  
16        gravity: ToastGravity.CENTER,  
17        timeInSecForIosWeb: 1, );}
```



Şekil 3.1: Firebase Ekleme

- **Ürün Güncelleme ve silme:** ürünleri düzenleme işlevselliği sağlayan bir ekranı içerir. Kullanıcı, ürün başlığı, fiyatı, kategori, birim ölçüsü, indirim durumu ve resim gibi özellikleri düzenleyebilir. Ayrıca ürün resmini güncelleyebilir veya ürünü silebilir. Firebase Firestore kullanılarak ürün bilgileri depolanır ve güncellenir. Kullanıcı etkileşimleri için çeşitli Flutter widget'ları ve Firebase Storage kullanılarak resim işlemleri gerçekleştirilir. Kod, asenkron işlemleri ve hata yönetimini içerir ve kullanıcıya bilgilendirme mesajları göstermek için Fluttertoast kullanır.
- **Ürün listeleme :** Ürünler ekledikten sonra ürünlerin detaylarını görüntülemek için . getProductsData fonksiyonu, belirtilen id'ye göre ürün verilerini getirir ve widget'ın UI'sını güncellenecek. UI, bir resim, fiyat bilgisi ve başlık içerir. Ayrıca, düzenleme ve silme gibi seçenekler sunan bir PopupMenuButton öğesi bulunmaktadır.



Şekil 3.2: Firebase Ekleme

4 Mobil Uygulama Geliştirme (Flutter)

4.1 Firebase Cloud Functions ve Firestore Entegrasyonu:

- Firebase Authentication login ve signIn: Bu kısım kullanıcı oturum açma işlemleri için tasarlanmıştır. Firebase yetkilendirme kullanılarak, e-posta ve şifre ile giriş yapma özelliği sağlanır. Kullanıcı dostu bir arayüz içerir; arka planında görsel efektler kullanılarak bir slayt gösterisi ve yarı şeffaf siyah renk bulunur. Kullanıcı girişi sırasında, geçerli bir formun doldurulup doldurulmadığı kontrol edilir ve Firebase üzerinden oturum açma işlemi gerçekleştirilir. Hata durumlarında bilgilendirme iletişim kutuları kullanıcıya gösterilir. Ek olarak, şifreyi unuttum bağlantısı, Google ile giriş butonu ve misafir olarak devam et bağlantısı gibi ek özellikler sunar. Kod, form yönetimi, Firebase entegrasyonu ve kullanıcı etkileşimi açısından kapsamlı bir örnek sunar. kod olarak aşağıda gösterilmiştir.

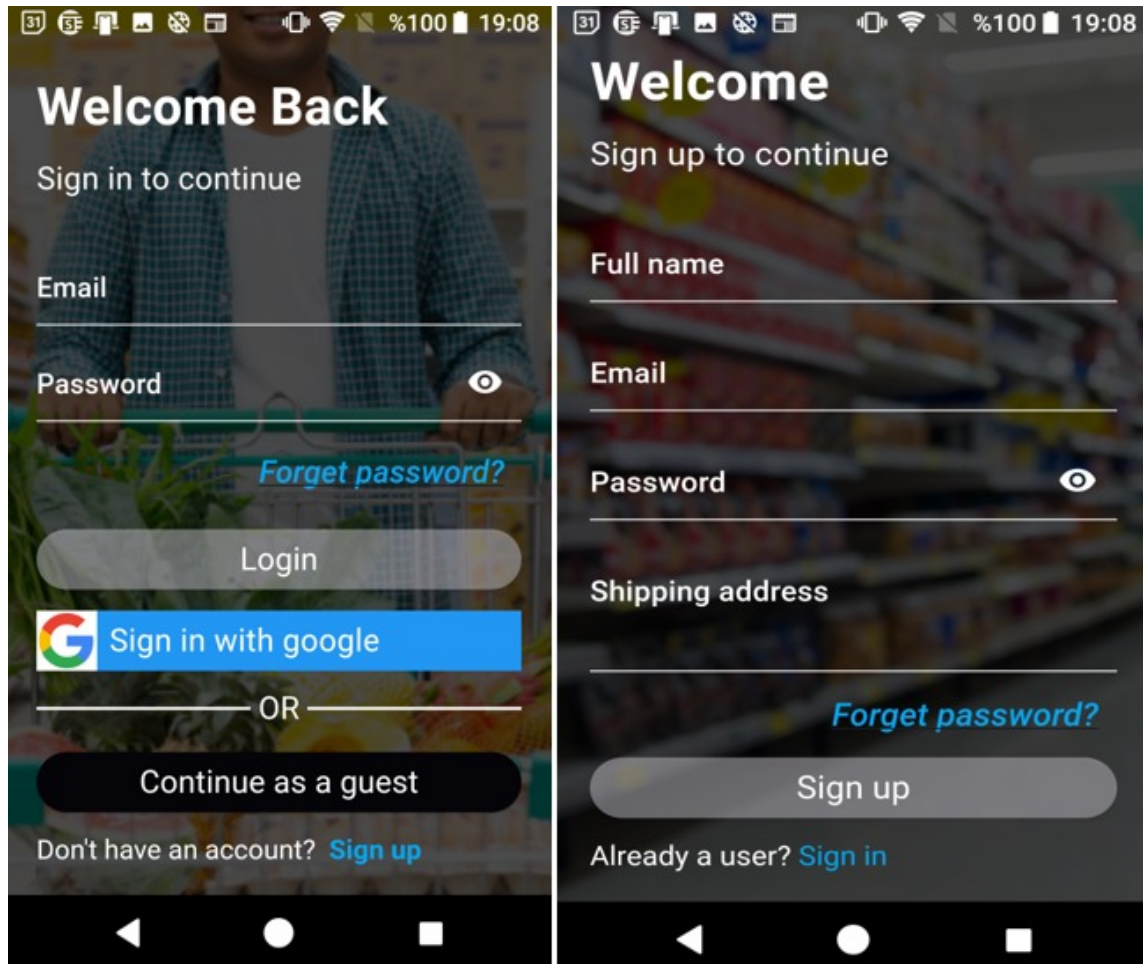
```
1 try {  
2     await authInstance.signInWithEmailAndPassword(  
3         email: _emailTextController.text.toLowerCase().trim()  
4         (),  
5         password: _passTextController.text.trim());  
6 }
```



```

5      Navigator.of(context).pushReplacement(MaterialPageRoute
      (
6          builder: (context) => const FetchScreen(),
7      ));
8      print('Succesfully logged in');
9  } on FirebaseException catch (error) {
10     GlobalMethods.errorDialog(
11         subtitle: '${error.message}', context: context);
12     setState(() {
13         _isLoading = false;
14     });

```

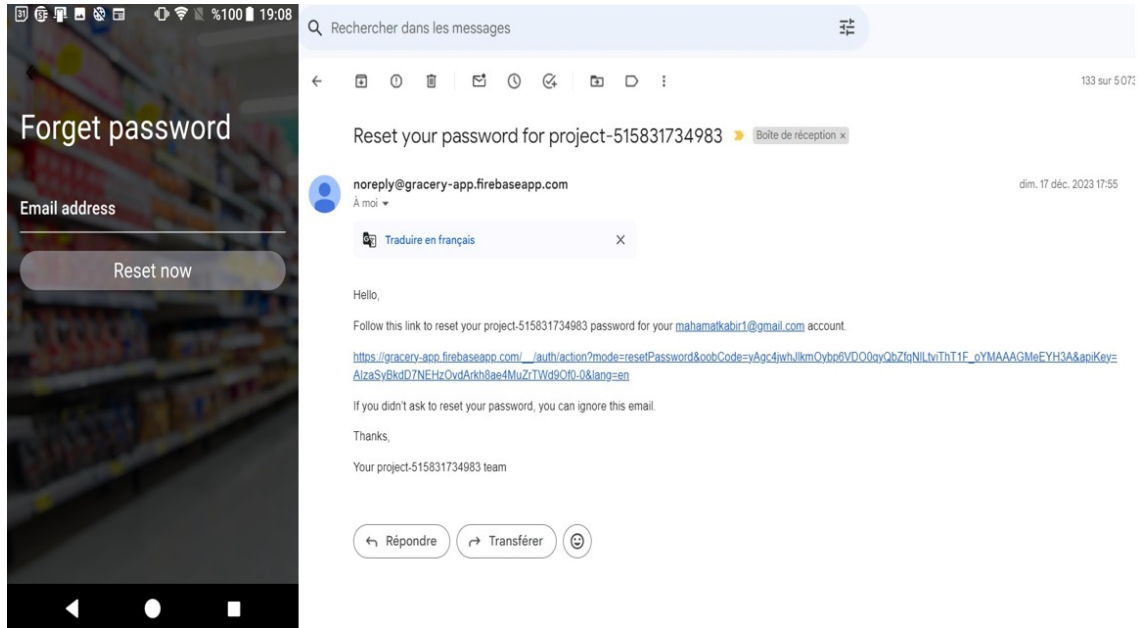


Şekil 4.1: Login in Sign Up

- Şifre Unutma: kullanıcıların unuttukları şifreleri sıfırlamak için tasarlanmış bir ekranı içermektedir. Kullanıcı, geçerli bir e-posta adresi girer ve ardından "Şimdi Sı-

fırla" düğmesine tıklar. Ardından, Firebase Authentication kullanılarak şifre sıfırlama işlemi gerçekleşir. Eğer işlem başarılı olursa, kullanıcıya başarılı bir bildirim gösterilir; aksi takdirde, hata mesajı görüntülenir.

```
1      try {
2          await authInstance.sendPasswordResetEmail (
3              email: _emailTextController.text.toLowerCase() );
4          Fluttertoast.showToast (
5              msg: "An Email has been sent to your email address
6                  ",
7              toastLength: Toast.LENGTH_LONG,
8              gravity: ToastGravity.TOP,
9              timeInSecForIosWeb: 1,
10             backgroundColor: Colors.green.shade800,
11             textColor: Colors.white,
12             fontSize: 16.0);
13     } on FirebaseException catch (error) {
```



Şekil 4.2: Forget-Password

- Mobil uygulama, Firestore üzerindeki ürün bilgilerini alır ve kullanıcılara görüntüler. Aşağıdaki görsel Uygulamada kullanılabilecek bir ürün kartını temsil eder. Kart, ürünün resmi, başlığı, fiyatı, sepete ekleme düğmesi ve isteğe bağlı olarak kalp simgesi içerir. Kullanıcı, kart üzerine tıkladığında ürün detaylarına gitmek için bir yön-

lendirme sağlanmıştır.ONSale Kısmi indirimli ürünleri temsil eder admin tarafından indirim modu aktifleştirdikten sonra mobile uygulamamız görünecek . normale belirtilen fiyatın yanında indirim yüzdeliği görünecek tabi üründen ürüne yüzdeliği değişiyor

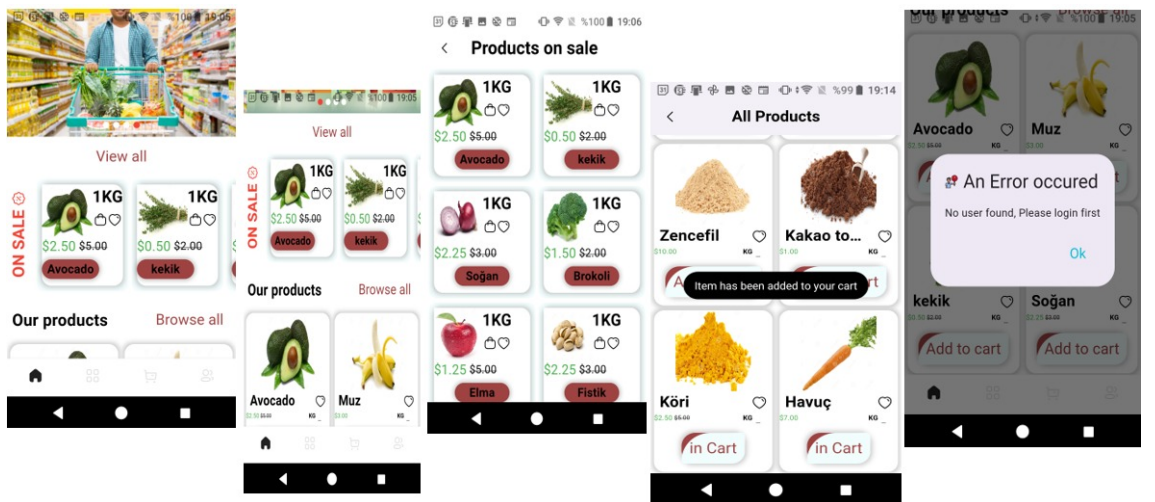
Widget, başlangıçta quantityTextController adlı bir TextEditingController kullanarak ürün miktarını takip eder. Widget başlatıldığında miktar varsayılan olarak '1' olarak ayarlanır ve dispose metoduyla temizlenir.

Kartın içeriği, FancyShimmerImage widget'i ile gösterilen ürün resmi, başlık, fiyat ve sepete ekleme düğmesi içerir. Ayrıca, isteğe bağlı olarak kalp simgesi de bulunur. Ürün miktarı kullanıcı tarafından bir TextFormField ile belirlenir.

Sepete ekle düğmesi, ürünün zaten sepette olup olmadığına bağlı olarak etkin veya etkisiz olabilir. Düğmeye tıklandığında, kullanıcının oturum açmış olması beklenir. Oturum açılmışsa, ürün sepete eklenir ve daha sonra sepeti güncellenir. Oturum açılmamışsa, kullanıcıya hata iletişimi gösterilir.

Görsel öğeler, ürün resimlerini şık bir şekilde göstermek için FancyShimmerImage widget'i kullanılarak düzenlenmiştir. Widget, görsel öğelerin yüklenme sırasında şimşek efektiyle görüntülenmesini sağlar.

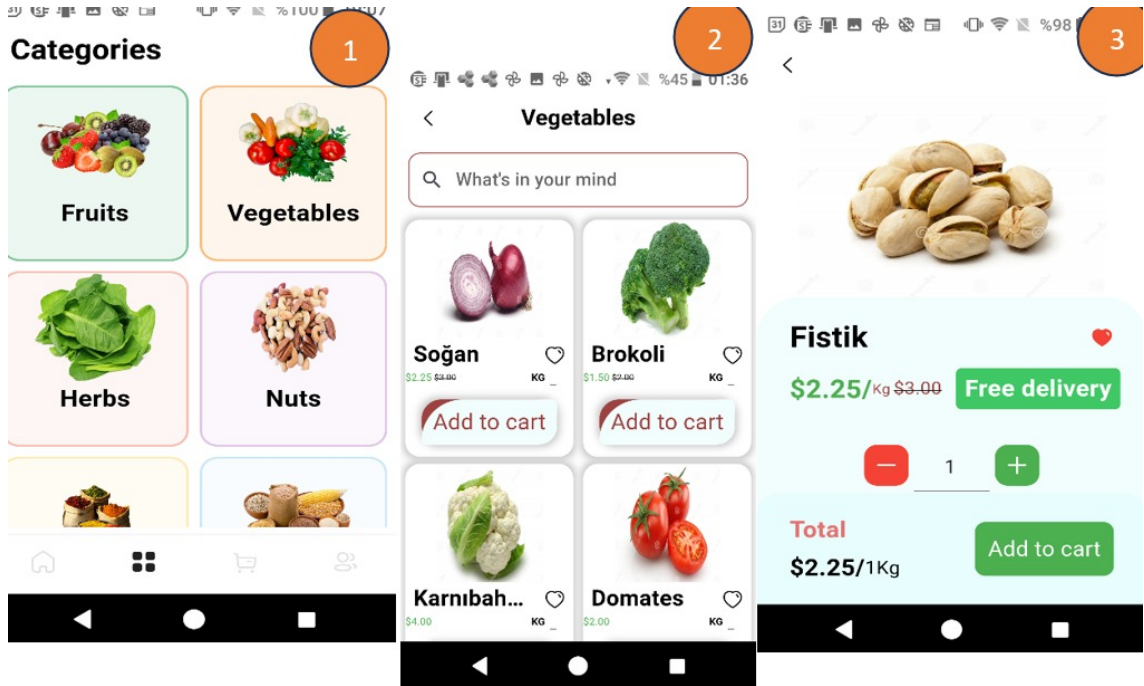
Bu widget, alışveriş uygulamalarında ürünleri listeleyp kullanıcının etkileşime geçmesini kolaylaştırmak için kullanılabilir.



Şekil 4.3: AnaSayfa

4.2 Kullanıcı İnteraktifliği

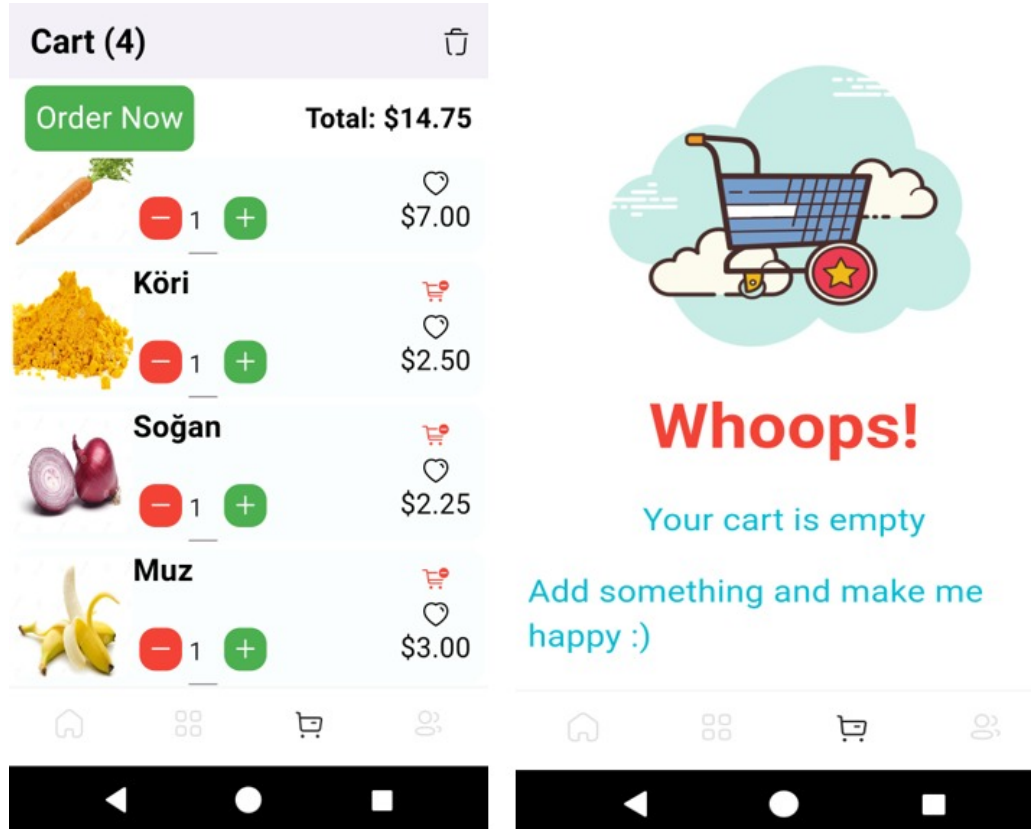
- **Kategori:** Aşağıdaki resim kategori temelli ürünleri gösteren bir ekranını oluşturur. CategoryScreen sınıfı, ekranın durumunu yönetir ve arama metni ile ilgili kontrolleri içerir. Ekranın içeriği, seçilen kategoride ürünlerin olup olmamasına bağlı olarak değişir. Eğer ürün yoksa, boş bir ürün bileşeni görüntülenir; aksi takdirde, bir arama çubuğu ve arama sonuçlarına göre düzenlenmiş ürün ızgarası veya bir mesaj görüntülenir. Arama çubuğu, kullanıcıların ürünleri aramasını sağlar ve girişe göre arama sonuçlarını dinamik olarak günceller. Grid View bölümü, ürünleri gösteren bir ızgara içerir ve arama sorgusu varsa veya yoksa, sağlayıcıyı değiştirerek dinamik bir şekilde güncellenir. Bu kod, Flutter'ın Provider paketini kullanarak durum yönetimini sağlar ve genel olarak ürünleri kategorilere göre düzenleyen ve aramayı destekleyen bir ekran tasarımını temsil eder.



Şekil 4.4: Categories

- **Alışveriş Sepeti:** yönetimi ve ödeme entegrasyonu sağlayan bir yapı içerir. Ana ekran, sepet içindeki öğeleri listeleyerek kullanıcıya bilgi sunar ve sepetin temizlenmesini sağlayan bir düğme içerir. Sepet boşsa, özel olarak tasarlanmış bir ekran görüntülenir, durum yönetimini sağlamak için Provider paketini kullanmaktadır. Bu

yaklaşım, alışveriş sepeti, ürün bilgileri ve favori durumu gibi verilere erişim sağlar. ürünün resmi, başlığı, miktarı ve fiyatını içermektedir. Kullanıcı, bu widget'a tıkladığında ProductDetails sayfasına yönlendirilmektedir. Kartın içeriği, resim, başlık, miktar değiştirme düğmeleri ve fiyat bilgilerini içermekte olup, miktarı artırmak ve azaltmak için düğmeler bulunmaktadır. Ayrıca, sepetten ürün çıkarmak için kaldırma düğmesi ve favorilere ekleme/den çıkarma için bir kalp ikonu da mevcuttur. Ödeme işlevselliği için Stripe entegrasyonu bulunmaktadır. Kullanıcılar, "Şimdi Sipariş Ver" düğmesine tıklayarak ödeme sürecini başlatır ve bu süreç, sunucu üzerinde Stripe ile iletişim kurarak gerçekleştirilir. Firebase kullanılarak kimlik doğrulama ve Firestore ile sipariş depolama işlemleri gerçekleştirilir. Yardımcı fonksiyonlar, kullanıcıya bilgi ileten iletişim kutularını görüntülemek için kullanılır. Kod, kullanıcıların alışveriş sepetini etkileşimli bir şekilde yönetmelerini ve güvenli bir ödeme süreci üzerinden sipariş vermelerini sağlayan bir mobil uygulama temelini oluşturur.

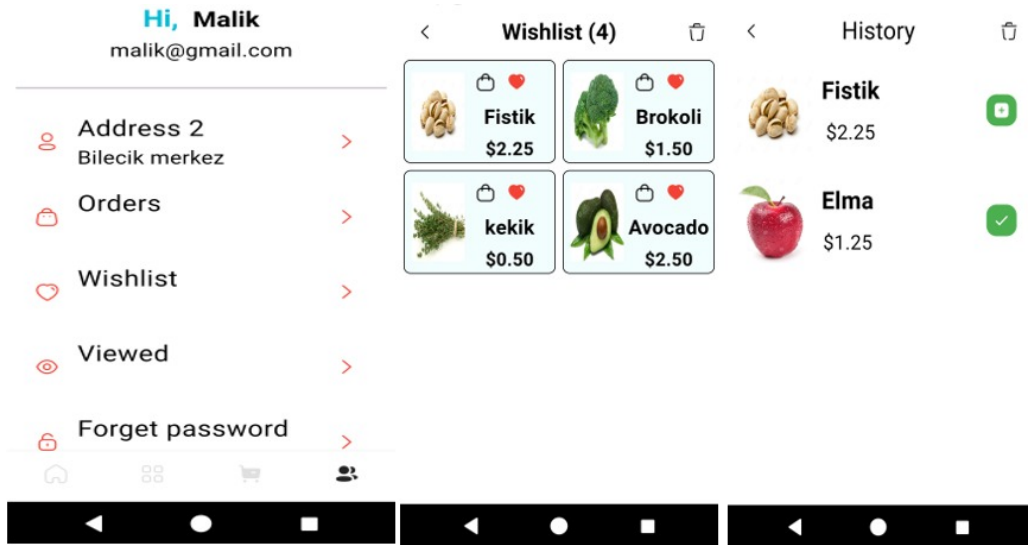


Şekil 4.5: Orders-Screen

- Aşağıdaki resim kullanıcı profili ekranını oluşturan bir UserScreen widget'ını içerir. Firebase Authentication ve Cloud Firestore kullanılarak kullanıcı bilgileri alınır ve bu bilgiler ekran üzerinde görüntülenir. Kullanıcının adı, e-posta adresi ve adres bilgisi gibi temel bilgiler, kullanıcı arayüzünde yer alır. Profil sayfasında ayrıca kullanıcının sipariş geçmişi, favori ürünleri, ve son görüntülenen ürünleri gibi farklı ekranlara yönlendiren sekmeler de bulunmaktadır.

Kod, kullanıcının adres bilgisini güncellemesine izin veren bir diyalog penceresi de içerir. Ayrıca, uygulamanın temasını değiştirme özelliği sayesinde kullanıcılar dark mode ve light mode arasında seçim yapabilirler. Oturum açmamış kullanıcılar için "Login" butonu, oturum açılmış kullanıcılar için ise "Logout" butonu bulunmaktadır.

Kodun düzenli bir yapıya sahip olması ve kullanıcı etkileşimleri için kullanılan farklı widget'lar, uygulamanın kullanıcı dostu bir arayüze sahip olmasını sağlar. Ayrıca, hata durumları için kullanıcıya bilgi veren dialog pencereleri de bulunmaktadır.



Şekil 4.6: Kullanıcı-Ekran

4.3 Kullanılan Paketler

Açık kaynak paketleri, özellikle Flutter uygulamaları oluşturmaya gelince, yazılım geliştirmede büyük rol oynar. Pub.dev, Flutter uygulama işlevsellik eklemesi ve topluluk tarafından oluşturulan çözümlerden yararlanması yardımcı olmak için 24.000'den fazla

paketten oluşan büyüyen bir ekosisteme sahiptir. Bu projede, kullanmış olduğum Flutter ve Dart paketlerinin derlenmiş bir listesini içermektedir.

4.3.1 shared Preferences

Flutter'da küçük boyuttaki verileri yerel olarak saklamak için kullanılan bir pakettir. Anahtar-değer çiftleri şeklinde veri saklar ve temel veri tiplerini (string, int, double, bool, string listesi gibi) destekler. Özellikle kullanıcı tercihleri, oturum durumu gibi küçük verilerin yerel depolanması için idealdir. Asenkron işlemleri destekler ve cihazda uygulama başlatıldığında verilere erişilebilir olmasını sağlar.

4.3.2 Provider

Flutter'da durum yönetimi için kullanılan bir pakettir. Widget ağacında veri akışını kolaylaştırır ve değişen verilerin widgetlere otomatik olarak iletilmesini sağlar. "ChangeNotifierProvider" gibi sınıflarla durumları yönetir ve "Consumer" veya "Provider.of" gibi widgetlerle bu durumları widgetlere aktarır. Basit kullanımı ve widget ağacında veri aktarımını sağlamasıyla bilinir.

4.3.3 google sign in

Flutter'da Google hesapları üzerinden kullanıcı kimlik doğrulamasını yönetmek için kullanılır. Bu paket, kullanıcıların Google hesaplarıyla oturum açmalarına olanak tanır, profil bilgilerine erişim sağlar ve genellikle Firebase Authentication ile entegre edilerek kullanılır. Uygulamaların Google hesapları aracılığıyla kimlik doğrulamasını basitleştirir ve yönetir.

4.3.4 Firebase

Firebase, uygulamalarınıza kimlik doğrulama, veritabanı, depolama, analitik, barındırma gibi bir dizi hizmet sunar. Bu hizmetlerin kullanımıyla, kullanıcı deneyimini geliştirebilir, uygulama performansını artırabilir ve kullanıcılarınızın verilerini güvenli bir şekilde saklayabilirsiniz. Firebase, uygulama geliştiricilerine sunduğu bu kolaylık ve çeşitli hizmetlerle genellikle tercih edilen bir platformdur.

4.3.5 Firebase- Auth

Flutter'da Firebase Authentication'ı entegre ederek uygulamalara kimlik doğrulama ve kullanıcı yönetimi yetenekleri kazandıran bir pakettir. Kullanıcıların farklı yöntemlerle (e-posta, telefon numarası, sosyal medya hesapları) oturum açmalarını ve uygulama içinde kimlik doğrulama işlemlerini kolaylaştırır. Ayrıca, Firebase'in güvenlik özelliklerini kullanarak veri güvenliğini artırır ve genellikle diğer Firebase servisleriyle entegre olarak kullanılır.

4.3.6 Firebase-core

Flutter uygulamalarına Firebase'in temel yapılandırılmasını sağlayarak Firebase platformunu entegre etmeyi amaçlar. Bu paket, Firebase'in diğer hizmetlerini kullanmadan önce uygulamayı Firebase platformuyla bağlar. Firebase projesinin tanımlanması ve temel konfigürasyonunun yapılması için kullanılır ve diğer Firebase hizmetlerinin kullanılabilmesi için gereklidir.

4.3.7 Cloud -Firestore

Flutter'da Firebase Cloud Firestore ile etkileşimde bulunmayı sağlayan bir pakettir. Firestore, NoSQL tabanlı bir bulut veritabanıdır ve bu paket, Flutter uygulamalarının Firestore veritabanıyla etkileşim kurmasına olanak tanır. Bu paket sayesinde Firestore'da veri okuma, yazma, güncelleme ve silme gibi işlemler gerçekleştirilebilir. Ayrıca, Firestore'un sunduğu gerçek zamanlı veri senkronizasyonu özelliğinden faydalanarak anlık veri güncellemeleri takip edilebilir. Bu paket, veritabanı işlemlerini kolaylaştırır ve Firebase'in sunduğu bulut veritabanı olanaklarından faydalanmayı sağlar.

4.3.8 Firebase-Storage

, Flutter'da Firebase Cloud Storage ile etkileşimde bulunmayı sağlar. Bu paket, kullanıcıların medya dosyalarını (resimler, videolar, dosyalar) Firebase'in bulut tabanlı depolama çözümüne yüklemelerine, indirmelerine ve yönetmelerine olanak tanır. Kullanıcılar, bu paket aracılığıyla dosyaları güvenli bir şekilde depolayabilir, paylaşabilir ve yönetebilir. Firebase Cloud Storage'in sunduğu ölçeklenebilir depolama imkanları bu paket üzerinden

kullanılabilir.

4.3.9 Flutter-Spink

Flutter’da kullanılabilecek çeşitli animasyonlu yüklenme göstergeleri sunan bir pakettir. Bu paket, çeşitli tarzlarda hazır animasyonlar içerir ve genellikle uygulamalarda işlem gerçekleştirilirken veya veri yüklenirken kullanıcıya görsel geri bildirim sağlamak için tercih edilir. Bu animasyonlu göstergeler, uygulamanın kullanıcı dostu ve etkileşimli olduğunu gösterir.

4.3.10 Fluttoast

Flutter’da mobil uygulamalarda basit metin tabanlı bildirimlerin (toast) gösterilmesini sağlayan bir pakettir. Bu paket, kullanıcıya geçici bir şekilde bildirim göstermek için kullanılır. Genellikle kullanıcıya geri bildirim vermek veya işlem tamamlandığında bilgilendirmek için tercih edilir. Özelleştirilebilir özellikleri sayesinde mesajın görüntülenme süresi ve stili gibi parametreler ayarlanabilir.

4.3.11 Uuid

Flutter’da benzersiz tanımlayıcıları oluşturmak için kullanılan bir pakettir. Universal Unique Identifier (Evrensel Benzersiz Tanımlayıcı) anlamına gelir. Bu paket, rastgele ve benzersiz ID’ler oluşturmak için kullanılır ve genellikle bir nesnenin veya öğenin benzersiz bir kimliğe ihtiyaç duyulduğu durumlarda kullanılır. Örneğin, veritabanı girişleri, oturum kimlikleri, öğeler arasında benzersiz kimlikler oluşturmak için kullanılabilir.

4.3.12 Flutter-Stripe

Flutter uygulamalarında Stripe ödeme işlemleri için kullanılan bir pakettir. Bu paket, Stripe API’leriyle etkileşime geçerek ödeme işlemlerini yönetmeyi sağlar. Kredi kartıyla ödeme, ödeme formu oluşturma, ödeme geçmişi takibi gibi Stripe özelliklerini uygulamalarınıza entegre etmenize yardımcı olur. Bu sayede kullanıcılarınızın kredi kartı bilgilerini güvenli bir şekilde işleyebilir ve ödemelerini yönetebilirsiniz.

4.3.13 Http

paketi, Flutter'da HTTP protokolünü kullanarak sunucularla iletişim kurmayı sağlayan bir pakettir. Bu paket, web API'leriyle etkileşimde bulunmak, sunuculardan veri almak ve sunuculara veri göndermek için kullanılır. GET, POST, PUT, DELETE gibi temel HTTP metodlarını kullanarak istekler yapılmasını sağlar. Gelen yanıtları işlemek ve veri türlerini kodlama/çözümlemek için yöntemler sunar. Kısacası, bu paket, Flutter uygulamalarının sunucularla veri alışverişi yapmasını kolaylaştırır.

4.3.14 image picker for web

Flutter web projelerinde resim seçme işlevselliğini sağlamak için kullanılan bir pakettir. Bu paket, tarayıcıların input elementinin type="file" özelliğini kullanarak resim seçme işlemi gerçekleştirir. Kullanıcıların galeriden veya kameradan resim seçmelerine olanak tanır ve seçilen resimleri uygulamaya entegre etmelerini sağlar. Bu sayede Flutter web uygulamalarında resim seçme ve kullanma işlemleri kolaylaşır ve kullanıcı deneyimi geliştirilir.

5 SONUÇLAR VE ÖNERİLER

6 EKLER

KAYNAKLAR

- [1] https://store.steampowered.com/app/375820/Human_Resource_Machine/ [Ziyaret Tarihi: 30 Mayıs 2022]
- [2] <https://tr.wikipedia.org/wiki/GitHub> [Ziyaret Tarihi: 6 Nisan 2022]
- [3] <https://sungraphica.itch.io/sci-fi-game-ui-collection-free-version> [Ziyaret Tarihi: 6 Nisan 2022] [Ziyaret Tarihi: 6 Nisan 2022]

ÖZGEÇMİŞ

KİŞİSEL BELGELER

Adı Soyadı :
Uyruğu : T.C.
Doğum Yeri ve Tarihi:
Adres :

Telefon :
E-mail :

EĞİTİM DURUMU

Lisans Öğrenimi : BŞEÜ Bilgisayar Mühendisliği Bölümü
Bitirme Yılı :
Lise :

İŞ DENEYİMLERİ

Yıl :
Kurum :
Stajlar :

İLGİ ALANLARI:

YABANCI DİLLER:

BELİRTMEK İSTEDİĞİNİZ DİĞER ÖZELLİKLER: