

FLUTTER

Flutter'a Başlangıç

1–4. haftalar

BT2025MOB1106 modülünün kazanımlarından hareketle; Android/Java ayrıntılarını kopyalamadan, öğrenci dostu öğrenme sırasıyla hazırlanmıştır.

9 birim • 36 hafta • 72 çalışan Flutter uygulaması • üç seviyeli uygulama görevleri

Öğrenme Haritası

Hafta	Ders	Uygulamalar
1	Flutter ve Dart ile İlk Uygulama	Merhaba Okul • Ders Kartım
2	Widget Ağacı ve MaterialApp	Tanıtım Ekranı • Günlük Hedef
3	Metin, Simge ve Görsel	Profil Kartı • Duyuru Panosu
4	Satır, Sütun ve Boşluk	Ürün Kartı • Ders Programı Kartı

1. Birim — Flutter'a Başlangıç

Bu birimde dört haftalık öğrenme döngüsü kullanılır: gör, birlikte çöz, değiştir, yeni probleme aktar.

1. Hafta — Flutter ve Dart ile İlk Uygulama

Geliştirme ortamı, proje yapısı, hot reload ve ilk widget

Bu derste konu, önce çalışan bir mobil arayüz üzerinde görülür; sonra widget ağacı küçük parçalara ayrılır. Kaynak kitaptaki kazanım korunur, ancak XML/Activity merkezli sıra yerine Flutter'ın widget, durum ve geri bildirim yaklaşımı kullanılır.

1	Geliştirme ortamı, proje yapısı, hot reload ve ilk widget
2	Girdi, işlem, çıktı ve durum değişimini ayırt etme
3	Okunabilir Dart kodu ve yeniden kullanılabilir widget geliştirme
4	Boş, sınır ve hatalı verilerle test etme

Çözümlü kitap uyarlaması: Merhaba Okul

1. MaterialApp ve Scaffold ile ekran iskeletini kurun. 2. Veriyi TextField ile alın. 3. Değişken değeri State içinde tutun. 4. Olaydan sonra setState ile ekranı yenileyin. 5. Kullanıcıya açık sonuç ve hata mesajı gösterin.

```
FilledButton(onPressed: () => setState(() => value++), child: const Text('Uygula'));
```

Çalışan ekran — Merhaba Okul

Merhaba Okul

HAFTA 1

Merhaba Okul

Geliştirme ortamı, proje yapısı, hot reload ve ilk widget

Deneme verisi

► Uygula

Sıfırla

1

Canlı sonuç

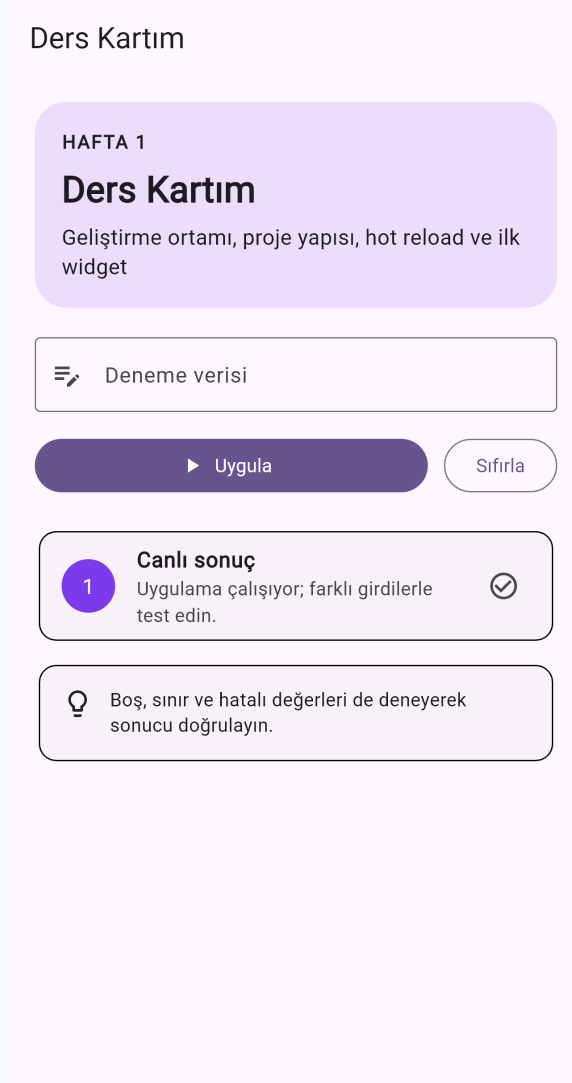
Uygulama çalışıyor; farklı girdilerle test edin.



Boş, sınır ve hatalı değerleri de deneyerek sonucu doğrulayın.

Bu ekran Flutter'ın gerçek widget testinden alınmıştır; kaynak proje çevrimdışı pakette çalışır durumdadır.

Özgün günlük yaşam uygulaması — Ders Kartım



Aynı teknik kazanım yeni bir problem bağlamında uygulanır. Öğrenci, örneği kopyalamak yerine veriyi ve ekran geri bildirimini kendi senaryosuna göre değiştirir.

Uygulama zamanı

Kolay	Başlık, renk, simge ve açıklamayı kendi senaryonuza uyarlayın.
Orta	Boş ve geçersiz girdiyi engelleyip Türkçe doğrulama mesajı gösterin.
İleri	Sonuçları bir modele dönüştürüp ListView içinde geçmiş olarak saklayın; silme ekleyin.

Geri bildirimden sonra kontrol: controller kaynakları dispose edildi mi, dar ekranda taşma var mı, boş/hata/yükleniyor durumları gösteriliyor mu ve en az üç test verisi denendi mi?

2. Hafta — Widget Ağacı ve MaterialApp

MaterialApp, Scaffold, AppBar ve widget ağacını okuma

Bu derste konu, önce çalışan bir mobil arayüz üzerinde görülür; sonra widget ağacı küçük parçalara ayrılır. Kaynak kitaptaki kazanım korunur, ancak XML/Activity merkezli sıra yerine Flutter'ın widget, durum ve geri bildirim yaklaşımı kullanılır.

1	MaterialApp, Scaffold, AppBar ve widget ağacını okuma
2	Girdi, işlem, çıktı ve durum değişimini ayırt etme
3	Okunabilir Dart kodu ve yeniden kullanılabilir widget geliştirme
4	Boş, sınır ve hatalı verilerle test etme

Çözümlü kitap uyarlaması: Tanıtım Ekranı

1. MaterialApp ve Scaffold ile ekran iskeletini kurun. 2. Veriyi TextField ile alın. 3. Değişen değeri State içinde tutun. 4. Olaydan sonra setState ile ekranı yenileyin. 5. Kullanıcıya açık sonuç ve hata mesajı gösterin.

```
FilledButton(onPressed: () => setState(() => value++), child: const Text('Uygula'));
```

Çalışan ekran — Tanıtım Ekranı

Tanıtım Ekranı

HAFTA 2

Tanıtım Ekranı

MaterialApp, Scaffold, AppBar ve widget ağacını okuma

Deneme verisi

► Uygula

Sıfırla

2

Canlı sonuç

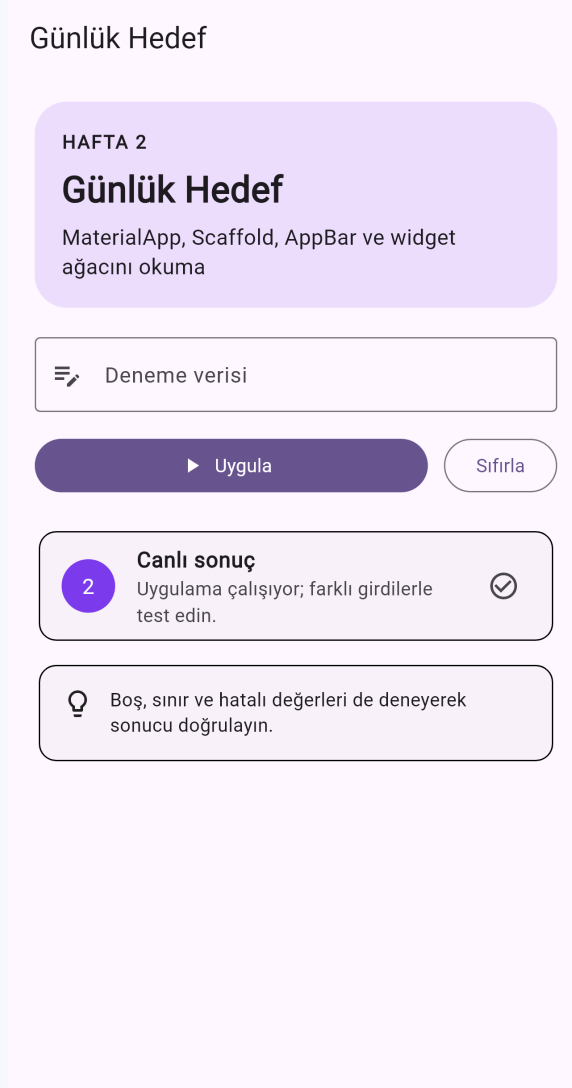
Uygulama çalışıyor; farklı girdilerle test edin.



Boş, sınır ve hatalı değerleri de deneyerek sonucu doğrulayın.

Bu ekran Flutter'ın gerçek widget testinden alınmıştır; kaynak proje çevrimdışı pakette çalışır durumdadır.

Özgün günlük yaşam uygulaması — Günlük Hedef



Aynı teknik kazanım yeni bir problem bağlamında uygulanır. Öğrenci, örneği kopyalamak yerine veriyi ve ekran geri bildirimini kendi senaryosuna göre değiştirir.

Uygulama zamanı

Kolay	Başlık, renk, simge ve açıklamayı kendi senaryonuza uyarlayın.
Orta	Boş ve geçersiz girdiyi engelleyip Türkçe doğrulama mesajı gösterin.
İleri	Sonuçları bir modele dönüştürüp ListView içinde geçmiş olarak saklayın; silme ekleyin.

Geri bildirimden sonra kontrol: controller kaynakları dispose edildi mi, dar ekranda taşma var mı, boş/hata/yükleniyor durumları gösteriliyor mu ve en az üç test verisi denendi mi?

3. Hafta — Metin, Simge ve Görsel

Text, Icon, Image ve temel görünüm özellikleri

Bu derste konu, önce çalışan bir mobil arayüz üzerinde görülür; sonra widget ağacı küçük parçalara ayrılır. Kaynak kitaptaki kazanım korunur, ancak XML/Activity merkezli sıra yerine Flutter'ın widget, durum ve geri bildirim yaklaşımı kullanılır.

1	Text, Icon, Image ve temel görünüm özellikleri
2	Girdi, işlem, çıktı ve durum değişimini ayırt etme
3	Okunabilir Dart kodu ve yeniden kullanılabilir widget geliştirme
4	Boş, sınır ve hatalı verilerle test etme

Çözümlü kitap uyarlaması: Profil Kartı

1. MaterialApp ve Scaffold ile ekran iskeletini kurun. 2. Veriyi TextField ile alın. 3. Değişen değeri State içinde tutun. 4. Olaydan sonra setState ile ekranı yenileyin. 5. Kullanıcıya açık sonuç ve hata mesajı gösterin.

```
FilledButton(onPressed: () => setState(() => value++), child: const Text('Uygula'));
```

Çalışan ekran — Profil Kartı

Profil Kartı

HAFTA 3

Profil Kartı

Text, Icon, Image ve temel görünüm özellikleri



Deneme verisi

► Uygula

Sıfırla

3

Canlı sonuç

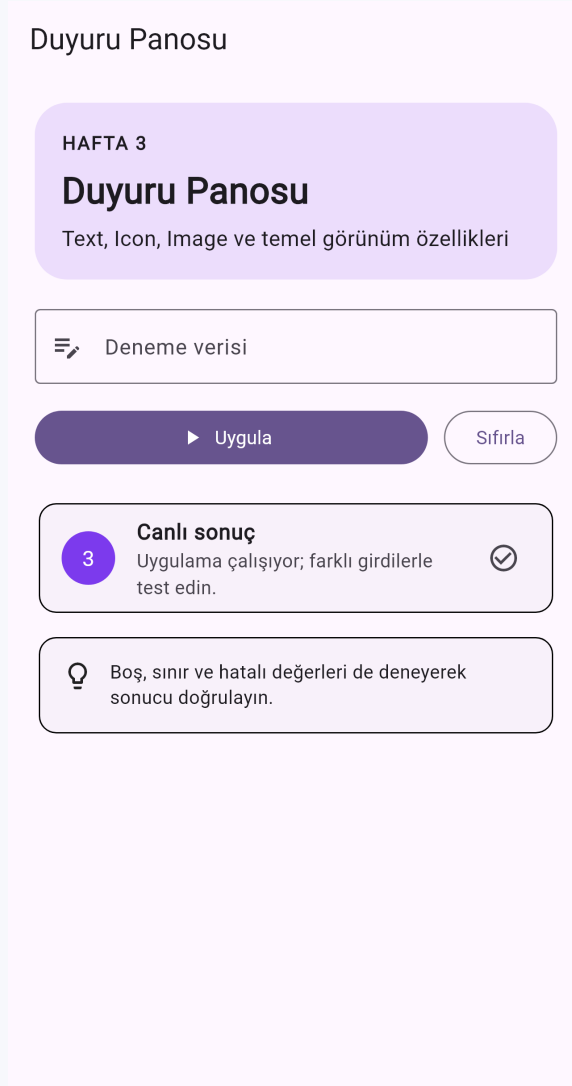
Uygulama çalışıyor; farklı girdilerle test edin.



Boş, sınır ve hatalı değerleri de deneyerek sonucu doğrulayın.

Bu ekran Flutter'ın gerçek widget testinden alınmıştır; kaynak proje çevrimdışı pakette çalışır durumdadır.

Özgün günlük yaşam uygulaması — Duyuru Panosu



Aynı teknik kazanım yeni bir problem bağlamında uygulanır. Öğrenci, örneği kopyalamak yerine veriyi ve ekran geri bildirimini kendi senaryosuna göre değiştirir.

Uygulama zamanı

Kolay	Başlık, renk, simge ve açıklamayı kendi senaryonuza uyarlayın.
Orta	Boş ve geçersiz girdiyi engelleyip Türkçe doğrulama mesajı gösterin.
İleri	Sonuçları bir modele dönüştürüp ListView içinde geçmiş olarak saklayın; silme ekleyin.

Geri bildirimden sonra kontrol: controller kaynakları dispose edildi mi, dar ekranda taşma var mı, boş/hata/yükleniyor durumları gösteriliyor mu ve en az üç test verisi denendi mi?

4. Hafta — Satır, Sütun ve Boşluk

Row, Column, Padding, SizedBox ve hizalama

Bu derste konu, önce çalışan bir mobil arayüz üzerinde görülür; sonra widget ağacı küçük parçalara ayrılır. Kaynak kitaptaki kazanım korunur, ancak XML/Activity merkezli sıra yerine Flutter'ın widget, durum ve geri bildirim yaklaşımı kullanılır.

1	Row, Column, Padding, SizedBox ve hizalama
2	Girdi, işlem, çıktı ve durum değişimini ayırt etme
3	Okunabilir Dart kodu ve yeniden kullanılabilir widget geliştirme
4	Boş, sınır ve hatalı verilerle test etme

Çözümlü kitap uyarlaması: Ürün Kartı

1. MaterialApp ve Scaffold ile ekran iskeletini kurun. 2. Veriyi TextField ile alın. 3. Değişen değeri State içinde tutun. 4. Olaydan sonra setState ile ekranı yenileyin. 5. Kullanıcıya açık sonuç ve hata mesajı gösterin.

```
FilledButton(onPressed: () => setState(() => value++), child: const Text('Uygula'));
```

Çalışan ekran — Ürün Kartı

Ürün Kartı

HAFTA 4

Ürün Kartı

Row, Column, Padding, SizedBox ve hizalama



Deneme verisi

► Uygula

Sıfırla

4

Canlı sonuç

Uygulama çalışıyor; farklı girdilerle test edin.



Boş, sınır ve hatalı değerleri de deneyerek sonucu doğrulayın.

Bu ekran Flutter'ın gerçek widget testinden alınmıştır; kaynak proje çevrimdışı pakette çalışır durumdadır.

Özgün günlük yaşam uygulaması — Ders Programı Kartı

Ders Programı Kartı

HAFTA 4

Ders Programı Kartı

Row, Column, Padding, SizedBox ve hizalama

☰

Deneme verisi

▶ Uygula

Sıfırla

4

Canlı sonuç

Uygulama çalışıyor; farklı girdilerle test edin.

✓

💡

Boş, sınır ve hatalı değerleri de deneyerek sonucu doğrulayın.

Aynı teknik kazanım yeni bir problem bağlamında uygulanır. Öğrenci, örneği kopyalamak yerine veriyi ve ekran geri bildirimini kendi senaryosuna göre değiştirir.

Uygulama zamanı

Kolay	Başlık, renk, simge ve açıklamayı kendi senaryonuza uyarlayın.
Orta	Boş ve geçersiz girdiyi engelleyip Türkçe doğrulama mesajı gösterin.
İleri	Sonuçları bir modele dönüştürüp ListView içinde geçmiş olarak saklayın; silme ekleyin.

Geri bildirimden sonra kontrol: controller kaynakları dispose edildi mi, dar ekranda taşma var mı, boş/hata/yükleniyor durumları gösteriliyor mu ve en az üç test verisi denendi mi?