

Flutter ile uygulama geliřtirme

HAKAN CELASUN

21360859215

BİLGİSAYAR MÜHENDİSLİĞİ 3.SINIF

25.04.2024



İÇİNDEKİLER

Neden Flutter?.....	1
Flutter ile uygulama geliştirme.....	2
Flutter kurulumu.....	3
İlk flutter projesinin oluşturulması.....	4
Projenin geliştirilmesi.....	5
Uygulamanın test edilmesi.....	6
Uygulamanın dağıtılması.....	7
Widgetlar ve UI tasarım.....	8
Paketlerin kullanımı.....	9
Firebase ile entegrasyon.....	10
Performans ve optimizasyon.....	11
Sürekli geliştirme ve güncelleme.....	12
Sonuç olarak.....	13
Sorularınız.....	14
Kaynaklar.....	15



Neden Flutter?

Flutter'ı kullanmanın birçok nedeni vardır ve işte bunlardan bazıları:

Tek Kod Tabanı: Flutter, hem iOS hem de Android için tek bir kod tabanı kullanarak uygulama geliştirmeyi sağlar. Bu, geliştirme sürecini hızlandırır ve aynı özellikleri farklı platformlara tekrar tekrar kodlamak zorunda kalmazsınız.

Performans: Flutter, hızlı ve akıcı kullanıcı arayüzleri oluşturmak için Google'ın gelişmiş teknolojilerini kullanır.

Güzel UI Tasarımı: Flutter, materyal tasarımı stilini içeren zengin bir widget kütüphanesine sahiptir. Bu, kullanıcı dostu ve görsel olarak çekici uygulamalar oluşturmanızı sağlar.

Hızlı Geliştirme Döngüsü: Flutter, sıcak yeniden yükleme (hot reload) özelliğini sunar. Bu özellik sayesinde, kodunuzu değiştirdikten hemen sonra sonuçları anında görebilirsiniz. Bu, geliştirme sürecini hızlandırır ve verimliliği artırır.

Açık Kaynak ve Topluluk Desteği: Flutter, açık kaynaklı bir projedir ve geniş bir geliştirici topluluğuna sahiptir. Bu topluluk, sorunları çözmek, bilgi paylaşmak ve yeni özellikler geliştirmek için birlikte çalışır.

Çeşitli Platform Desteği: Flutter, sadece iOS ve Android için değil, aynı kod tabanını kullanarak web, masaüstü ve gömülü sistemler gibi farklı platformlara da uygulama geliştirmenize olanak tanır. Bu, uygulamanızı birden fazla platformda yayınlamanızı ve kullanıcı tabanınızı genişletmenizi sağlar.



Flutter

Flutter ile Uygulama Geliştirme

- Flutter, Google tarafından geliştirilen açık kaynaklı bir UI yazılım geliştirme kitidir. Hem iOS hem de Android için tek bir kod tabanı kullanarak yüksek performanslı, güzel ve kullanıcı dostu uygulamalar oluşturmanıza olanak tanır. Bu sunumda, Flutter ile uygulama geliştirmenin adımlarını adım adım ele alacağız.

android studio



Adım 1: Flutter Kurulumu

Flutter'ı kullanmaya başlamak için bilgisayarınıza Flutter SDK'yı kurmanız gerekir. Flutter'ı resmi web sitesinden indirebilir ve kurulum adımlarını takip edebilirsiniz. Ayrıca, geliştirme için Android Studio, IntelliJ IDEA veya Visual Studio Code gibi bir IDE kurmanız önerilir.

Adım 2: İlk Flutter Projesinin Oluşturulması

- Flutter SDK yüklendikten sonra, bir Flutter projesi oluşturmak oldukça kolaydır. Bir terminal veya komut istemcisinde aşağıdaki komutu çalıştırarak yeni bir Flutter projesi oluşturabilirsiniz:
- `flutter create ilk_flutter_projesi`
- Bu komut, "ilk_flutter_projesi" adında yeni bir Flutter projesi oluşturacaktır.

Adım 3: Projenin Geliştirilmesi

- Flutter projesi oluşturulduktan sonra, uygulamanızı geliştirmeye başlayabilirsiniz. Projenizin ana klasörüne giderek lib klasöründeki main.dart dosyasını düzenleyin. Bu dosya, uygulamanızın ana giriş noktasıdır.
- Dart programlama dilini kullanarak UI bileşenleri oluşturabilirsiniz. Flutter, materyal tasarımı destekler, böylece uygulamanızın görünümünü kolayca özelleştirebilirsiniz.

Adım 4: Uygulamanın Test Edilmesi

TEST COVERAGE

- Uygulamanızı geliştirmeye başladığınızda, sık sık test etmek önemlidir. Flutter, bu süreci hızlandırmak için "sıcak yeniden yükleme" (hot reload) özelliğini sunar. Bu özellik sayesinde, kodunuzu değiştirdikten hemen sonra sonuçları anında görüntüleyebilirsiniz. İşte uygulamanızı test etmek için adımlar:
- Bir komut istemcisinde veya terminalde projenizin dizinine gidin.
- Aşağıdaki komutu kullanarak uygulamanızı çalıştırın:
- `flutter run`
- Bu komut, uygulamanızı bir simülatörde veya bağlı bir cihazda çalıştıracaktır. Uygulama yüklendikten sonra, kodunuzu değiştirdiğinizde veya güncellediğinizde, sıcak yeniden yükleme özelliği sayesinde değişiklikleri hemen görebilirsiniz.



Adım 5: Uygulamanın Dağıtılması

Uygulamanızı test ettiğinizde ve hazır olduğunda, dağıtım geçme zamanı gelmiştir. Flutter, uygulamanızı hem Android hem de iOS için dağıtmayı kolaylaştırır. İşte her iki platform için uygulamanızı dağıtmanın adımları:

Android için:

Bir komut istemcisinde veya terminalde, projenizin dizinine gidin.

Aşağıdaki komutu kullanarak APK dosyasını oluşturun:

```
flutter build apk
```

Bu komut, projenizin APK dosyasını build/app/outputs/flutter-apk dizinine oluşturacaktır. Bu APK dosyasını Google Play Store veya diğer dağıtım platformlarına yükleyebilirsiniz.

iOS için:

Öncelikle, iOS için bir geliştirici hesabına ve bir sertifikaya sahip olmanız gerekir.

Flutter projesinin ana dizininde bir komut istemcisinde veya terminalde aşağıdaki komutu çalıştırın

```
flutter build ios
```

Bu komut, projenizin iOS projesini oluşturacaktır.

Adım 6: Widget'lar ve UI Tasarımı

Types of widgets

- Flutter'da UI tasarımı, widget'lar kullanılarak yapılır. Widget'lar, uygulamanızın farklı parçalarını temsil eder ve bir araya getirilerek kompleks kullanıcı arayüzleri oluşturulur. Flutter, hem içsel (built-in) hem de özel (custom) widget'lar sunar. İşte bazı temel widget türleri:
- StatelessWidget:** Durumu (state) olmayan, sabit bir görünüm sunan widget.
- StatefulWidget:** Durumu olan, dinamik olarak değişen bir görünüm sunan widget.
- Container:** Diğer widget'ları düzenlemek ve stillemek için kullanılan bir widget.
- Row ve Column:** Sırasıyla yatay ve dikey düzenlemeler sağlayan widget'lar.
- Text:** Metin görüntülemek için kullanılan bir widget.
- Image:** Görüntüleri görüntülemek için kullanılan bir widget.

Adım 7: Paketlerin Kullanımı

Flutter, birçok farklı özelliği ve işlevselliği desteklemek için geniş bir paket ekosistemine sahiptir. Örneğin, HTTP istekleri yapmak, veritabanı işlemleri gerçekleştirmek, resimleri işlemek gibi birçok işlev için paketler bulunmaktadır. İhtiyacınız olan bir paketi projenize dahil etmek için `pubspec.yaml` dosyasını düzenlemeniz gerekir.

```
dependencies: http: ^0.13.3
```

Yukarıdaki örnek, HTTP istekleri için `http` paketini projeye dahil eder. Daha sonra, bu paketi kullanarak HTTP istekleri yapabilirsiniz.

Adım 6: Firebase ile Entegrasyon

- Firebase, Google'ın sunduğu bir platformdur ve uygulamalarınıza gerçek zamanlı veritabanı, kimlik doğrulama, analitikler, bulut depolama gibi birçok özellik eklemenizi sağlar. Flutter projenize Firebase'i eklemek oldukça kolaydır. Firebase'i projenize eklemek için aşağıdaki adımları takip edebilirsiniz:
- Firebase Console'da yeni bir proje oluşturun.
- Projenize Firebase'i ekleyin ve gerekli hizmetleri etkinleştirin.
- Flutter projesine Firebase SDK'nın eklenmesi.
- Firebase'in sunduğu özellikleri kullanmak için Firebase SDK'larını ve ilgili paketleri projenize ekleyebilirsiniz.

With Firebase

Adım 9: Performans ve Optimizasyon

- Uygulamanızın performansını ve kullanıcı deneyimini artırmak için bazı optimizasyon tekniklerini kullanabilirsiniz. Örneğin, gereksiz yeniden çizimleri önlemek için `const` anahtar kelimesini kullanabilir veya `ListView` gibi veri yoğun widget'ları kullandığınızda sayfa yüklenirken ekranı doldurmak için `ListView.builder` kullanabilirsiniz.
- Ayrıca, uygulamanızın performansını izlemek için Flutter'ın sunduğu araçları kullanabilir ve gerektiğinde iyileştirmeler yapabilirsiniz.



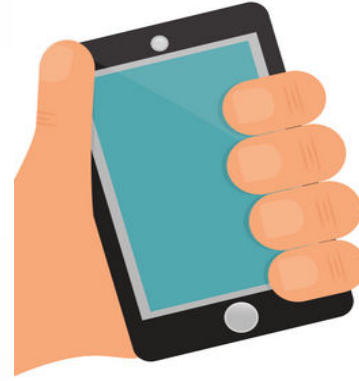
Adım 10: Sürekli Geliştirme ve Güncelleme

Flutter

- Bir uygulama geliştirdikten sonra, sürekli olarak geri bildirim almak, hataları düzeltmek ve yeni özellikler eklemek önemlidir. Kullanıcıların geri bildirimlerini dikkate alarak uygulamanızı sürekli olarak geliştirmeniz gerekir. Ayrıca, platformlar (iOS ve Android) için düzenli olarak güncellemeler yayımlamak da önemlidir.

SONUÇ OLARAK

Flutter, Google tarafından geliştirilen açık kaynaklı bir UI yazılım geliştirme kitidir. Hem iOS hem de Android için tek bir kod tabanı kullanarak yüksek performanslı, güzel ve kullanıcı dostu uygulamalar oluşturmanıza olanak tanır. Hızlı geliştirme döngüsü, güçlü widget kütüphanesi, açık kaynak ve geniş topluluk desteği gibi avantajları bulunmaktadır. Ancak, uygulama boyutu, native performans ve üçüncü taraf entegrasyonu gibi bazı dezavantajlar da vardır. Sonuç olarak, Flutter, modern ve hızlı mobil uygulama geliştirmek isteyenler için güçlü bir seçenektir.



Flutter

SORULARINIZ?

- Flutter'ın sıcak yeniden yükleme (hot reload) özelliği nasıl çalışır ve geliştirme sürecine nasıl katkı sağlar?
- **Cevap:** Flutter'ın sıcak yeniden yükleme özelliği, uygulama çalışırken yapılan kod değişikliklerini anında yansıtır. Geliştirici, kodu değiştirdiğinde uygulama anında yeniden yüklenir ve değişiklikler hemen görüntülenir. Bu, geliştirme sürecini hızlandırır ve anlık geri bildirim almayı sağlar.
- Flutter ile aynı kod tabanını kullanarak hangi platformlara uygulama geliştirilebilir?
- **Cevap:** Flutter ile aynı kod tabanını kullanarak iOS, Android, web, masaüstü ve gömülü sistemler gibi farklı platformlara uygulama geliştirilebilir.
- Flutter'da Firebase'i entegre etmek kolay mıdır avantajları nelerdir?
- **Cevap:** Flutter'da Firebase'i entegre etmek kolaydır çünkü Firebase, Flutter SDK'sıyla doğrudan entegre edilmiştir. Bu, gerçek zamanlı veritabanı, kimlik doğrulama, analizler, bulut depolama gibi birçok hizmetin kolayca kullanılmasını sağlar.
- Flutter'ın Dart programlama dili ile hangi avantajları sunar ve Dart'ın diğer dillerden farkları nelerdir?
- **Cevap:** Flutter'ın Dart programlama dili ile sunduğu avantajlar şunlardır: hızlı derleme süresi, kolay okunabilirlik, güçlü asenkron programlama desteği ve geniş kapsamlı standart kütüphaneler. Dart, özellikle Flutter ile uyumlu olarak geliştirilmiş ve performans odaklı bir dil olarak tasarlanmıştır.

Kaynaklar

- <https://flutter.dev/docs>
- <https://flutter.dev/docs/code labs>
- <https://www.youtube.com/channel/UCwXdFgeE9KYz1DdR7TG9cMw>
- <https://www.reddit.com/r/FlutterDev/>
- <https://flutter.dev/community>

A background of deep red, vertically pleated stage curtains. The curtains are slightly parted in the center, revealing a dark stage floor at the bottom. The lighting is soft, creating subtle shadows and highlights on the fabric folds.

DİNLEDİĞİNİZ İÇİN
TEŞEKKÜRLER!