

Mobil Programlama

Sena Gürkan - 20120205021

HAFTA 1 - ÖDEV

1- Aşağıdaki soruları cevaplayınız

- **1- Cross platform nedir?**

Cross-platform, bir yazılımın tek bir kod tabanıyla birden fazla işletim sisteminde çalıştırılabilmesini sağlar. Bu yöntem, geliştiricilerin uygulamayı sadece bir kez yazıp, farklı platformlarda kullanabilir hale getirmesine olanak tanır. Uygulama, işletim sistemi ile bir köprü veya SDK (Software Development Kit) aracılığıyla derlenir ve çalıştırılır. Bu teknolojiye örnek olarak React Native, Flutter, ve Xamarin gibi framework'ler gösterilebilir.

- **2- Native platform nedir?**

Native platform, uygulamaların işletim sistemi tarafından doğrudan derlenebildiği geliştirme yöntemidir. Her platformun kendine özgü programlama dilleri kullanılarak geliştirilir. Örneğin, Android için Android Studio, Eclipse, veya IntelliJ IDEA IDE'leri kullanılarak Java veya Kotlin dilleriyle uygulama geliştirilir. iOS için ise Xcode kullanılarak Objective-C veya Swift dilleriyle mobil uygulamalar geliştirilebilir. Native uygulamalar, platforma özgü en iyi performansı ve kullanıcı deneyimini sunar.

- **3- Üstekine alternatifler Ionic, Cordoba... nedir ve neden ihtiyaç duyulur? alternatif midir?**

Ionic ve Apache Cordova, hibrit mobil uygulama geliştirme çerçeveleridir. Bu çerçeveler, HTML, CSS, ve JavaScript gibi web teknolojilerini kullanarak mobil uygulama geliştirilmesine olanak tanır. Geliştirilen bu uygulamalar, bir native kabuğa sarılarak Android ve iOS gibi platformlarda çalıştırılır. Yani web teknolojileriyle yazılan uygulamalar, mobil uygulamalara dönüştürülmüş olur. Bu sayede, geliştiriciler tek bir kod tabanıyla birden fazla platforma uygulama yayinskyabilirler.

Apache Cordova, web teknolojileriyle yazılan uygulamaların native bileşenlerle entegre edilmesi için kullanılır. Cordova, cihazın kamera, GPS gibi donanımlarına erişimi sağlayarak, geliştiricilerin web teknolojileriyle oluşturduğu uygulamaların mobil cihazlarla uyumlu şekilde çalışmasını sağlar. Cordova, web uygulamalarını mobil platformlara uyarlamak için bir köprü görevi görür.

Ionic, Cordova'nın üzerine inşa edilmiş olup, geliştiricilere daha modern bir arayüz ve kullanıcı deneyimi sağlar. Ionic, çeşitli kullanıcı arayüzü bileşenleri ve temalar sunarak, mobil uygulama geliştirme sürecini hızlandırır. Ionic ile yapılan uygulamalar, hem estetik açıdan daha güçlü bir arayüze sahip olur hem de kullanıcı dostu bir deneyim sunar.

Bu çerçevelere ihtiyaç duyulmasının temel nedeni, bir kez yazılan kodun birden fazla platformda çalıştırılabilmesidir. Geliştiriciler, native uygulama geliştirme sürecinde her platform için ayrı ayrı kod yazmak yerine, hibrit çözümlerle iş yükünü azaltabilirler. Bu açıdan Ionic ve Cordova, React Native, Flutter gibi diğer cross-platform çözümlerine birer alternatif olarak öne çıkar.

- **4- Arayüz geliştirme hakkında bilgi veriniz**

Arayüz geliştirme, bir mobil veya web uygulamasının kullanıcı ile etkileşim kurduğu kısmını oluşturma sürecidir. Kullanıcı arayüzü (UI), görsel tasarım unsurlarını içerir ve kullanıcının uygulamaya nasıl etkileşimde bulunduğuunu belirler. İyi bir arayüz, kullanıcı deneyimini (UX) olumlu yönde etkileyen bir yapıya sahip olmalıdır.

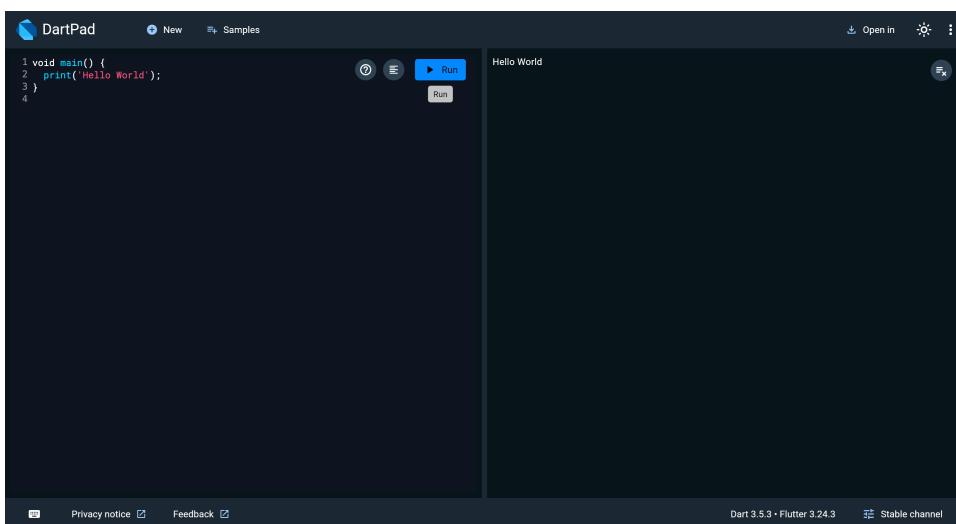
Mobil uygulama arayüzü geliştirme sürecinde:

- **Wireframe:** Uygulamanın temel yapısını ve tasarımını oluşturur.
- **Mockup:** Wireframe üzerinde daha detaylı, renkler ve görseller içeren arayüz tasarımcıdır.
- **Prototip:** Uygulamanın interaktif versiyonu olup, arayüz bileşenlerinin nasıl çalışacağını gösterir.

Geliştirme sırasında arayüz elementleri tasarlanır ve kullanıcının deneyimsel bir şekilde uygulamayı kullanmasını sağlar. React Native ve Flutter gibi framework'ler ile arayüz geliştirme yapılabilir.

2- Aşağıdaki soruları cevaplayınız

- Hello World

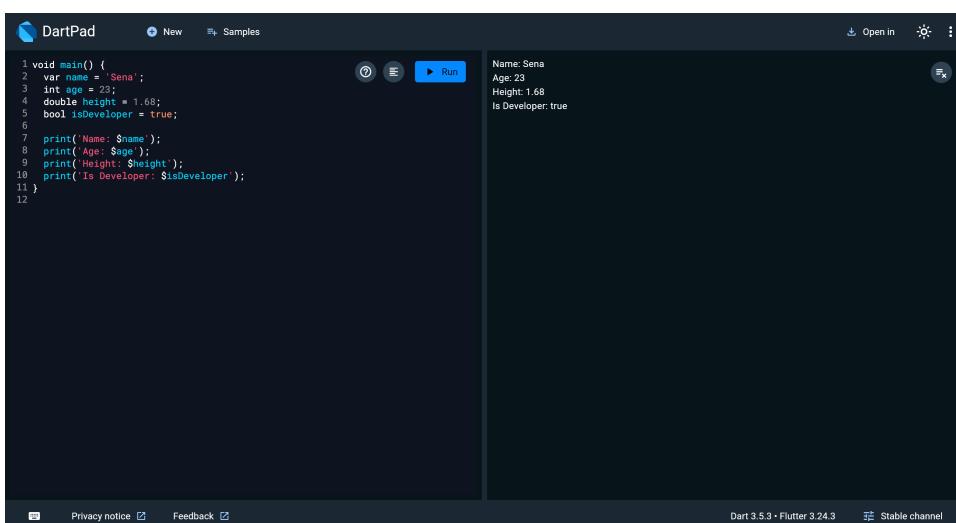


The screenshot shows the DartPad interface. On the left, the code editor contains the following Dart code:

```
1 void main() {  
2   print('Hello World');  
3 }  
4
```

On the right, the preview window displays the output of the code: "Hello World". Below the preview, the status bar shows "Dart 3.5.3 • Flutter 3.24.3" and "Stable channel".

- Variables



The screenshot shows the DartPad interface. On the left, the code editor contains the following Dart code:

```
1 void main() {  
2   var name = 'Sena';  
3   int age = 23;  
4   double height = 1.68;  
5   bool isDeveloper = true;  
6  
7   print('Name: $name');  
8   print('Age: $age');  
9   print('Height: $height');  
10  print('Is Developer: $isDeveloper');  
11 }  
12
```

On the right, the preview window displays the output of the code:
Name: Sena
Age: 23
Height: 1.68
Is Developer: true

Below the preview, the status bar shows "Dart 3.5.3 • Flutter 3.24.3" and "Stable channel".

- Control flow statements

DartPad interface showing a Dart program. The code uses if-else statements and a for loop to print letter grades based on a score. The output shows the result of running the code.

```

1 void main() {
2   var not = 92;
3
4   if (not >= 90) {
5     print('Harf notu AA.');
6   } else if (not >= 80) {
7     print('Harf notu BA.');
8   } else if (not >= 70) {
9     print('Harf notu BB.');
10 } else if (not >= 60) {
11   print('Harf notu CB.');
12 } else if (not >= 50) {
13   print('Harf notu CC.');
14 } else {
15   print('Harf notu FF.');
16 }
17
18 for (var i = 1; i <= 5; i++) {
19   print('*' * i);
20 }
21
22

```

Output: Harf notu AA.

*
**

Dart 3.5.3 • Flutter 3.24.3 Stable channel

- Functions

DartPad interface showing a Dart program. It defines a factorial function and calls it with the argument 5. The output shows the result of running the code.

```

1 void main() {
2   int sayı = 5;
3   int sonuc = faktöriyel(sayı);
4   print('Faktöriyel $sayı = $sonuc');
5 }
6
7 int faktöriyel(int n) {
8   int sonuc = 1;
9
10  for (int i = 1; i <= n; i++) {
11    sonuc *= i;
12  }
13
14  return sonuc;
15 }
16
17
18

```

Output: Faktöriyel 5 = 120

Dart 3.5.3 • Flutter 3.24.3 Stable channel

- Comments

DartPad interface showing a Dart program. It contains a multi-line comment block. The output shows the result of running the code.

```

1 void main() {
2
3 // Tek satırlık yorum
4
5   print ('Yorum satırları');
6
7 /*
8 * Çok
9 * Satırlı
10 * Yorum
11 */
12
13

```

Output: Yorum satırları

Dart 3.5.3 • Flutter 3.24.3 Stable channel

- Imports

A screenshot of the DartPad interface. The left panel shows the following Dart code:

```
1 import 'dart:math';
2 void main() {
3     var randomNumara = Random().nextInt(10);
4     print('Random numara: $randomNumara');
5 }
6
7
```

The right panel displays the output "Random numara: 5". The bottom status bar indicates "Dart 3.5.3 • Flutter 3.24.3" and "Stable channel".

- Classes

A screenshot of the DartPad interface. The left panel shows the following Dart code:

```
1 void main() {
2     print('Hello World');
3 }
4
```

The right panel displays the output "Hello World". The bottom status bar indicates "Dart 3.5.3 • Flutter 3.24.3" and "Stable channel".

- Enums

A screenshot of the DartPad interface. The left panel shows the following Dart code:

```
1 enum KullaniciTuru {
2     Yonetici,
3     Egitmen,
4     Ogrenci
5 }
6
7 void main() {
8     var Kullanici = KullaniciTuru.Ogrenci;
9
10    switch (Kullanici) {
11        case KullaniciTuru.Yonetici:
12            print('Kullanici türü: Yonetici');
13            break;
14        case KullaniciTuru.Egitmen:
15            print('Kullanici türü: Egitmen');
16            break;
17        case KullaniciTuru.Ogrenci:
18            print('Kullanici türü: Ogrenci');
19            break;
20    }
21 }
```

The right panel displays the output "Kullanici türü: Ogrenci". The bottom status bar indicates "Dart 3.5.3 • Flutter 3.24.3" and "Stable channel".

- Inheritance

```

1 class Arac {
2   void hareketEt() {
3     print('Arac hareket ediyor.');
4   }
5 }
6
7 class Bisiklet extends Arac {
8   @override
9   void hareketEt() {
10    print('Sena bisiklette hareket ediyor.');
11  }
12 }
13
14 class Araba extends Arac {
15   @override
16   void hareketEt() {
17    print('Sena araba ile hareket ediyor.');
18  }
19 }
20
21 void main() {
22   var bisiklet = Bisiklet();
23   var araba = Araba();
24
25   bisiklet.hareketEt();
26   araba.hareketEt();
27 }
28

```

Sena bisiklette hareket ediyor.
Sena araba ile hareket ediyor.

Dart 3.5.3 • Flutter 3.24.3 Stable channel

- Mixins

```

1 mixin Kosmak {
2   void kos() {
3     print('Koşabilmek');
4   }
5 }
6
7 mixin Ziplamak {
8   void zipla() {
9     print('Ziplayabilmek');
10  }
11 }
12
13 class Sporcu with Kosmak, Ziplamak {}
14
15 void main() {
16   var sporcu = Sporcu();
17   sporcu.kos();
18   sporcu.zipla();
19 }
20

```

Koşabilmek
Ziplayabilmek

Dart 3.5.3 • Flutter 3.24.3 Stable channel

- Interfaces and abstract classes

```

1 abstract class Hayvan {
2   void sesCikar();
3 }
4
5 class Kedi extends Hayvan {
6   @override
7   void sesCikar() {
8     print('Kedi miyavlıyor.');
9   }
10 }
11
12 class Kopek extends Hayvan {
13   @override
14   void sesCikar() {
15     print('Köpek havlıyor.');
16   }
17 }
18
19 void main() {
20   var kedi = Kedi();
21   var kopek = Kopek();
22
23   kedi.sesCikar();
24   kopek.sesCikar();
25 }
26

```

Kedi miyavlıyor.
Köpek havlıyor.

Dart 3.5.3 • Flutter 3.24.3 Stable channel

- Async

```

1 Future<String> veriGetir() async {
2   return Future.delayed(Duration(seconds: 3), () => 'Veri başarıyla yüklandı');
3 }
4
5 void main() async {
6   print('Veri çekiliyor...');
7
8   var veri = await veriGetir();
9
10  print(veri);
11
12  print('Diğer işler devam ediyor...');
13 }
14

```

Dart 3.5.3 • Flutter 3.24.3 Stable channel

- Exceptions

```

1 void main() {
2   try {
3     int sonuc = 10 ~/ 0;
4     print('Sonuc: $sonuc');
5   } catch (e) {
6     print('Hata: $e');
7   } finally {
8     print('Program sonlandı.');
9   }
10 }
11

```

Dart 3.5.3 • Flutter 3.24.3 Stable channel

3- Kurulumlar

Geçen hafta 1. haftadaki ödevi React Native kurulumu olarkin mailden iletmisdim, ancak Flutter ile devam etmeye karar verdigim için kurulum kismini guncelleyerek tekrar iletiyorum. Xcode (simulatör için) ve VSCode bilgisayarında zaten kurulu. Öncelikle Flutter'ı bilgisayarıma kurarak başladım.

Get started

[Set up Flutter](#)

Learn Flutter [▼](#)

Stay up to date [▼](#)

App solutions [▼](#)

User interface

Introduction

Widget catalog

Layout [▼](#)

Adaptive & responsive design [▼](#)

Design & theming [▼](#)

Interactivity [▼](#)

Assets & media [▼](#)

Navigation & routing [▼](#)

Download then install Flutter

To install Flutter, download the Flutter SDK bundle from its archive, move the bundle to where you want it stored, then extract the SDK.

1. Download the following installation bundle to get the latest stable release of the Flutter SDK.

Intel Processor	Apple Silicon
flutter_macos_3.24.3-stable.zip	flutter_macos_arm64_3.24.3-stable.zip

For other release channels, and older builds, check out the [SDK archive](#).

The Flutter SDK should download to the macOS default download directory: `~/Downloads/`.

2. Create a folder where you can install Flutter.

Consider creating a directory at `~/development/`.

3. Extract the file into the directory you want to store the Flutter SDK.

```
$ unzip ~/Downloads/flutter_macos_arm64_3.24.3-stable.zip \
-d ~/development/
```

When finished, the Flutter SDK should be in the `~/development/flutter` directory.

Content

Verify sys
Hardw
Softwa

[Install the](#)

Configur
Install
Config
device
Install
Check yo
setup

Run Flk
Troubl
issues

Start dev
macOS a
Manage

```
senagurkan@senas-mac-3 ~ % mkdir development
senagurkan@senas-mac-3 ~ % senagurkan -- zsh -- 124x40
senagurkan@senas-mac-3 ~ % unzip ~/Downloads/flutter_macos_arm64_3.24.3-stable.zip \d ~/development/
```

```
senagurkan@senas-mac-3 ~ % cd development
senagurkan@senas-mac-3 ~ % open .zshrc
```

```
.zshrc — Edited
export PATH="/opt/homebrew/bin:$PATH"

# Herd injected NVM configuration
export NVM_DIR="/Users/senagurkan/Library/Application Support/Herd/config/nvm"
[ -s "$NVM_DIR/nvm.sh" ] && \. "$NVM_DIR/nvm.sh" # This loads nvm
[ -f "/Applications/Herd.app/Contents/Resources/config/shell/zshrc.zsh" ] && builtin source "/Applications/Herd.app/Contents/Resources/config/shell/zshrc.zsh"

# Herd injected PHP 8.3 configuration.
export HERO_PHP_83_INI_SCAN_DIR="/Users/senagurkan/Library/Application Support/Herd/config/php/83/"

# Herd injected PHP binary.
export PATH="/Users/senagurkan/Library/Application Support/Herd/bin/:$PATH"
[ -s "$NVM_DIR/bash_completion" ] && \. "$NVM_DIR/bash_completion" # This loads nvm
bash_completion
ulimit -n 10240
export PATH=$HOME/development/flutter/bin:$PATH

|
```

```
senagurkan@senas-mac-3 ~ % open .zshrc
senagurkan@senas-mac-3 ~ % flutter doctor
Doctor summary (to see all details, run flutter doctor -v):
[-] Flutter (Channel stable, 3.24.3, on macOS 15.0.1 24A348 darwin-arm64, locale en-TR)
  [-] Android toolchain - developed for Android devices (Android SDK version 35.0.0)
  [-] Xcode - developed for iOS and macOS (Xcode 16.0)
  [-] Java - developed for the web
  [-] Android Studio (version 2024.1)
  [-] VS Code (version 1.94.0)
  [-] Connected device (3 available)
  [-] Network resources

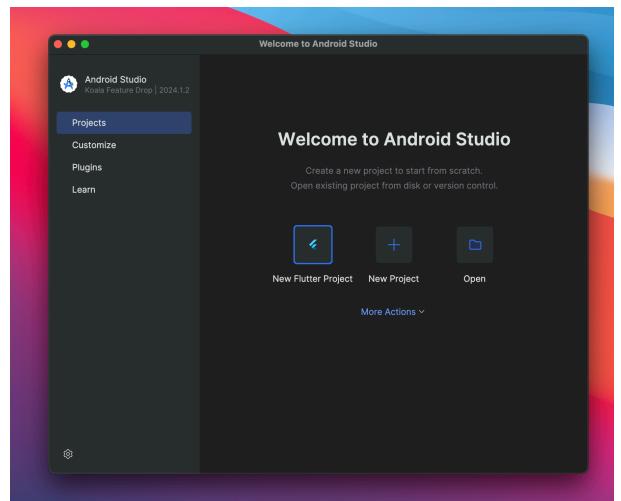
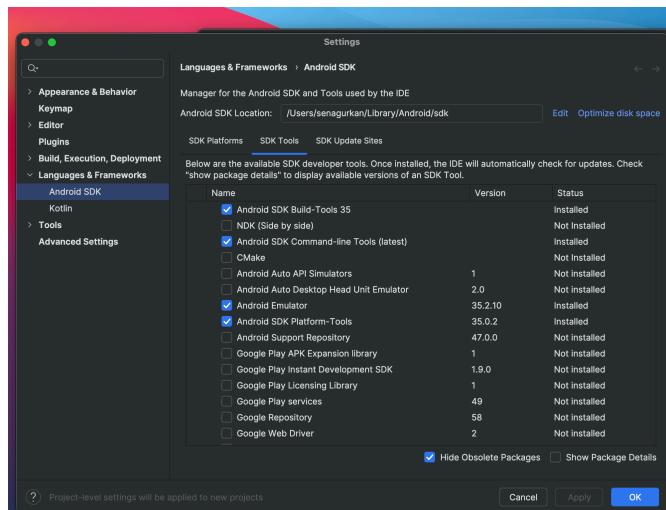
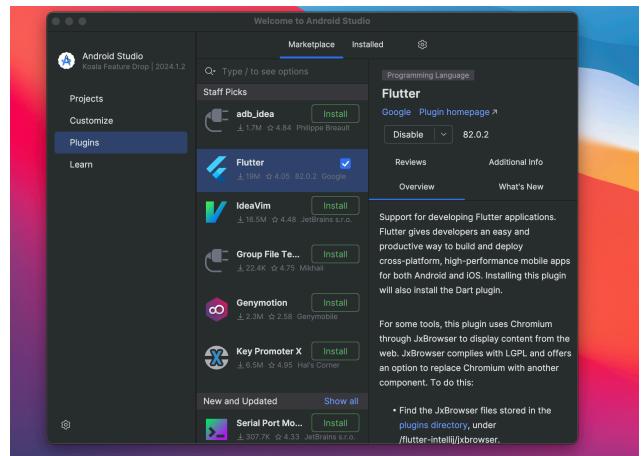
* No issues found!
senagurkan@senas-mac-3 ~ % open .zshrc
senagurkan@senas-mac-3 ~ % open .zshrc
senagurkan@senas-mac-3 ~ % flutter --version
Flutter 3.24.3 • channel stable • https://github.com/flutter/flutter.git
Framework • revision 26631baaa7 (4 weeks ago) • 2024-09-11 16:27:48 -0500
Engine • revision 3.5.3 • DevTools 2.37.3
senagurkan@senas-mac-3 ~ %
```

Flutter kurulumu bu şekildeydi, daha sonra Android Studio kurulumuna geçtim.

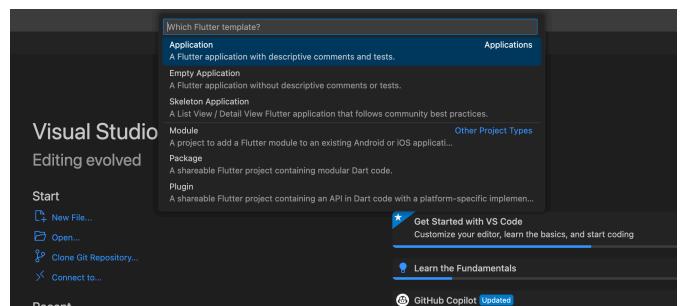
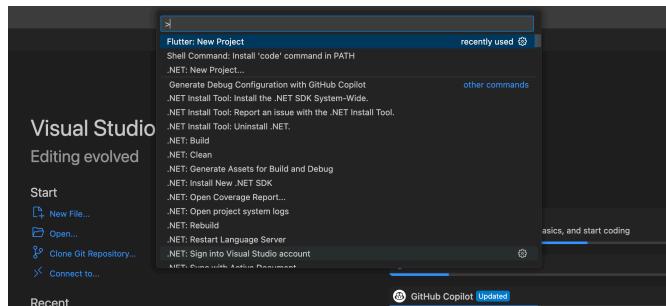
Android Studio Kurulum :

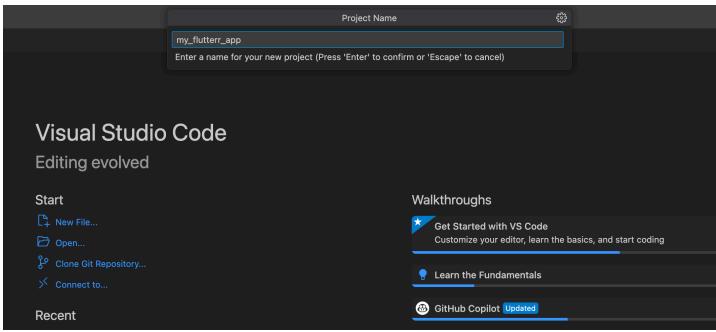
Platform	Android Studio paketi	Boyut	SHA-256 sağlaması
Windows (64 bit)	android-studio-2024.1.2.12-windows.exe Önerili	1.2 GB	81aaef2197a6ebbd7c60effe7ab74ddb6a18eeb1c94249c53a42b9efs59054
Windows (64 bit)	android-studio-2024.1.2.12-windows.zip exe yükleyici yok	1.2 GB	50180dc87d045f4c9a6ba6aa920f8a06780bf27325b2676a5e232871d772f7fe
Mac (64 bit)	android-studio-2024.1.2.12-mac.dmg	1.3 GB	1cc2919c25a544bef63eace74575df334d1dd96bc86f98bf2cbfd5d9e51c540
Mac (64 bit, ARM)	android-studio-2024.1.2.12-mac_arm.dmg	1.2 GB	e2519765464d4614cfbc477d0cc8ee6f92055da283a4ab7042783a8dc27d1
Linux (64 bit)	android-studio-2024.1.2.12-linux.tar.gz	1.2 GB	745168820e989a9085f842d47ce541407db09df7b8ab20770f6ea89e41a6e92
Chrome OS	android-studio-2024.1.2.12-cros.deb	992.2 MB	0a01c7dfb3cae55a8498779382981619f228a4c45fe87aca0d0e7d0cc776f3c6

Daha fazla indirme şurada mevcuttur: arşivleri indirebilirsiniz. Örneğin, Android Emulator'ın indirilmesi için [Emülatör indirme arşivleri](#).



Daha sonra projemi oluşturdum ve hem iOS hem android simülatörde çalıştım:





```

my_flutter_app
main.dart
lib/main.dart
  MyApp
    import 'package:flutter/material.dart';
    ...
    void main() {
      runApp(MyApp());
    }
  class MyApp extends StatelessWidget {
    const MyApp({Key key}) : super(key);
    ...
    @override
    Widget build(BuildContext context) {
      return MaterialApp(
        title: 'Flutter Demo',
        theme: ThemeData(
          ...
        ),
        ...
      );
    }
  }
}

void main() {
  runApp(MyApp());
}

class MyApp extends StatelessWidget {
  const MyApp({Key key}) : super(key);

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Flutter Demo',
      theme: ThemeData(
        ...
      ),
      ...
    );
  }
}

void main() {
  runApp(MyApp());
}

class MyApp extends StatelessWidget {
  const MyApp({Key key}) : super(key);

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Flutter Demo',
      theme: ThemeData(
        ...
      ),
      ...
    );
  }
}

```

The terminal output shows the command `flutter run` being executed, followed by the application's splash screen and the home page with a button counter.

Bu da模拟器lerde çalıştığım hali :

