

Flutter ile Kamera Entegrasyonu



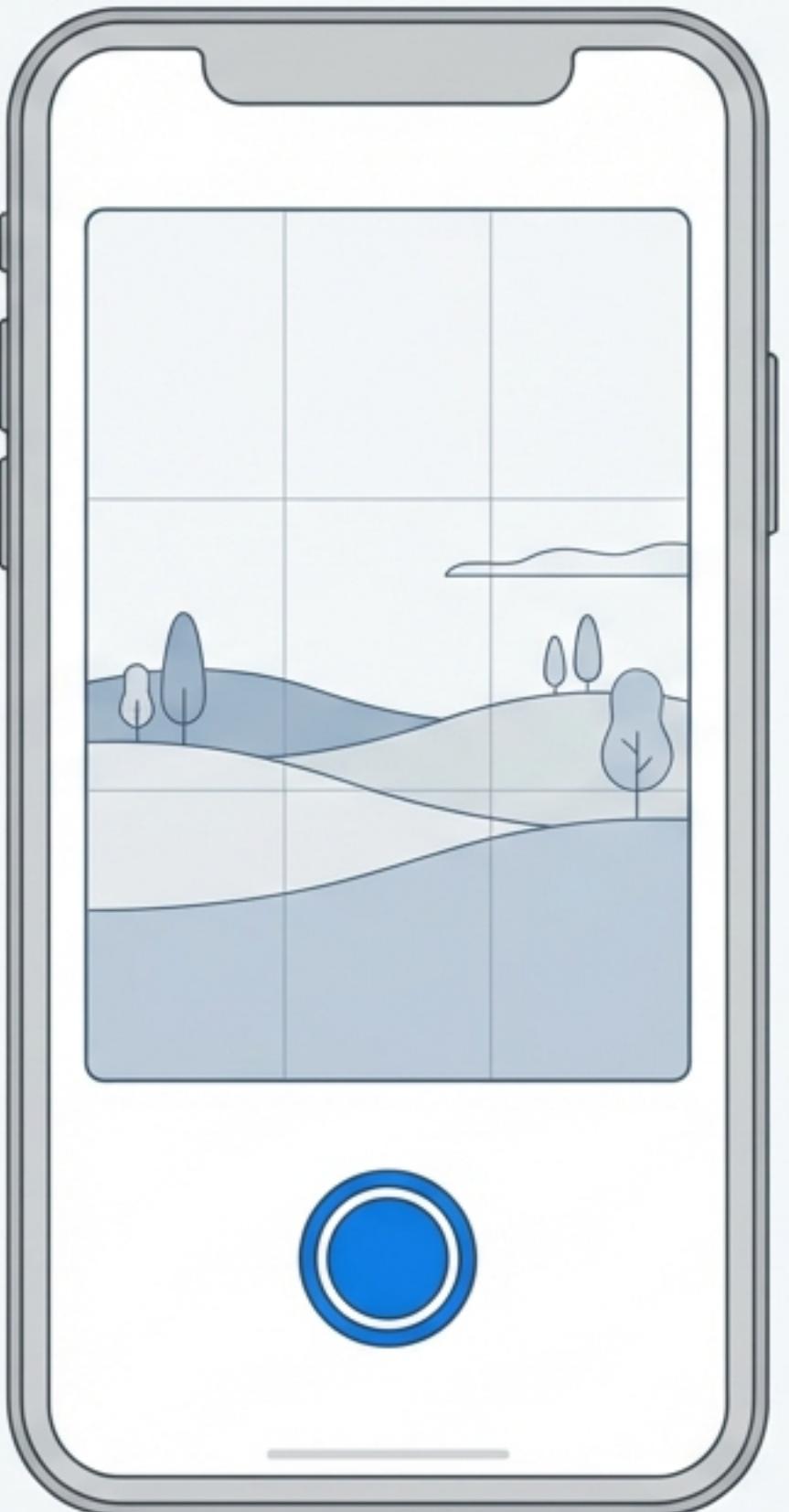
Cihaz Kamerasıyla Fotoğraf Çekme ve Kaydetme Rehberi

Hedef: Canlı Önizleme ve GörSEL Kaydı

Bu rehberde, cihazın kamerasını kullanan, canlı bir önizleme (live preview) sunan ve çekilen fotoğrafı cihazın geçici dizinine kaydeden bir uygulama inşa edeceğiz.

Insight

Resmi 'camera' paketini kullanarak herhangi bir cihaz kamerasından görüntü akışı almayı ve fotoğraf işlemeyi öğreneceğiz.



Gerekli Bağımlılıklar (Dependencies)

camera - Cihazın donanım kameralarıyla çalışmak için resmi eklenti.

path_provider - İşletim sisteminin dosya dizinlerine (klasör yollarına) erişmek için.

path - Farklı platformlarda dosya yollarını hatasız birleştirmek için.

```
dependencies:  
  flutter:  
    sdk: flutter  
  camera: ^0.10.0  
  path_provider: ^2.0.0  
  path: ^1.8.0
```

Kritik Yapılandırma Ayarları



Android Konfigürasyonu



camera eklentisi için minimum SDK sürümü 21 olmalıdır.

```
defaultConfig {  
    // ...  
    minSdkVersion 21  
    // ...  
}
```



iOS İzinleri

ios/Runner/Info.plist dosyasına kullanıcı izin açıklamaları eklenmelidir.

- Privacy - Camera Usage Description
- Privacy - Microphone Usage Description

Donanımı ve Uygulamayı Başlatma

```
void main() async {
  WidgetsFlutterBinding.ensureInitialized();

  final cameras = await availableCameras();
  final firstCamera = cameras.first;

  runApp(TakePictureApp(camera: firstCamera));
}
```

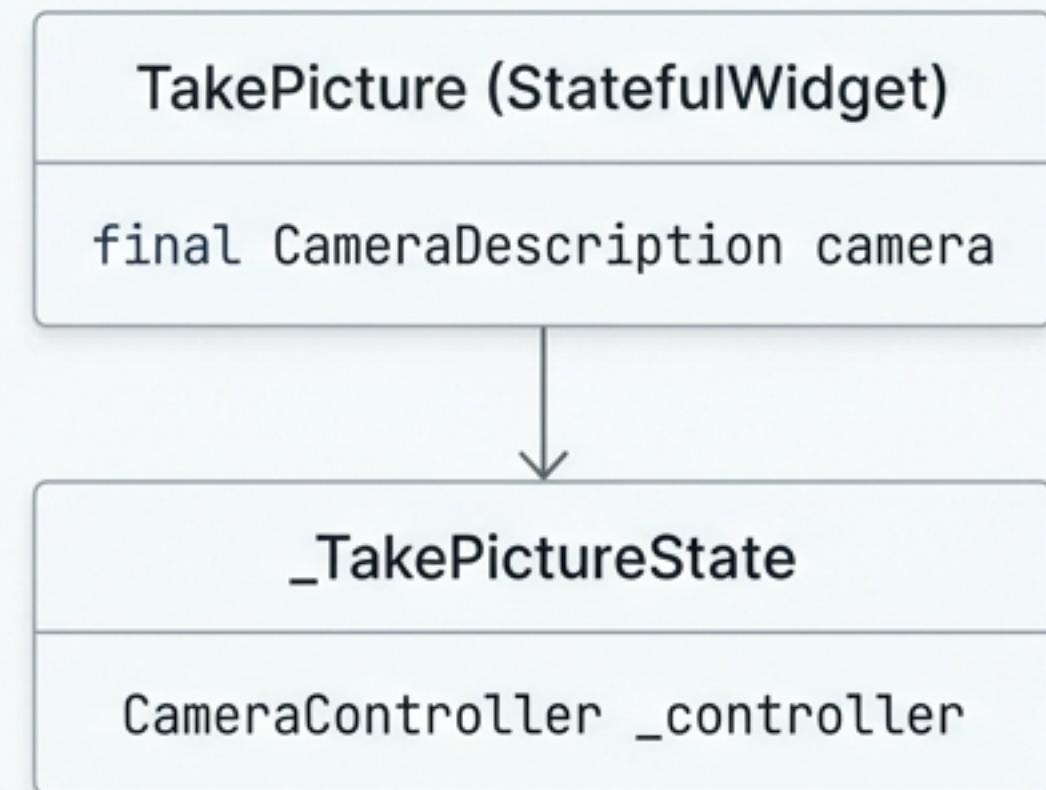
Eklentiler yüklenmeden önce Flutter motorunu hazırlar.

Cihazdaki tüm kameraları (ön/arka) listeler.

Genellikle arka kameryayı seçer.

Kontrolcü (Controller) Mimarisi

Cihaz kamerasıyla bağlantı kurmak için bir "CameraController" gereklidir. Bu kontrolcü, sistem kaynaklarını yönetmek adına başlatılmalı (initialize) ve işi bittiğinde kapatılmalıdır (dispose). Bu nedenle bir StatefulWidget kullanıyoruz.



```
class TakePicture extends StatefulWidget {  
    final CameraDescription camera;  
    const TakePicture({super.key, required this.camera});  
    // ...  
}
```

Yaşam Döngüsü ve Çözünürlük

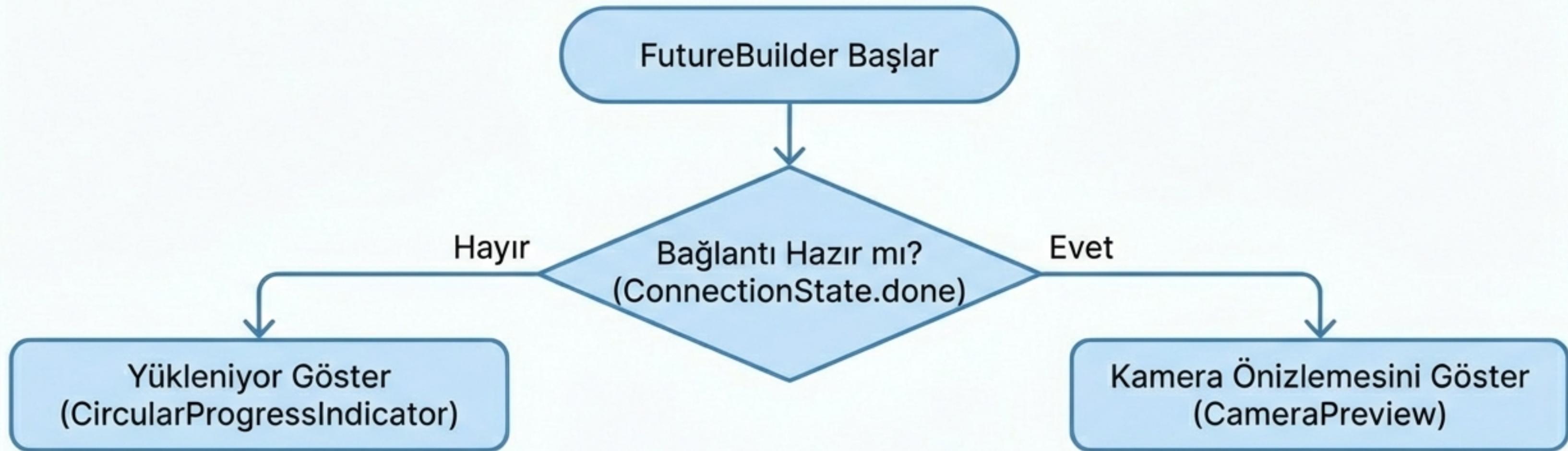
```
// Başlangıç (Init)
@Override
void initState() {
    super.initState();
    _controller = CameraController(
        widget.camera,
        ResolutionPreset.high,
    );
    _initController = _controller.initialize();
}
```

```
// Temizlik (Dispose)
@Override
void dispose() {
    _controller.dispose();
    super.dispose();
}
```

ResolutionPreset



Asenkron Arayüz Yönetimi



Kamera donanımı hemen hazır olmaz. '`_initController`' tamamlanana kadar kullanıcıya bir yükleme ekranı, tamamlandığında ise kamera görüntüsü sunmalıyız.

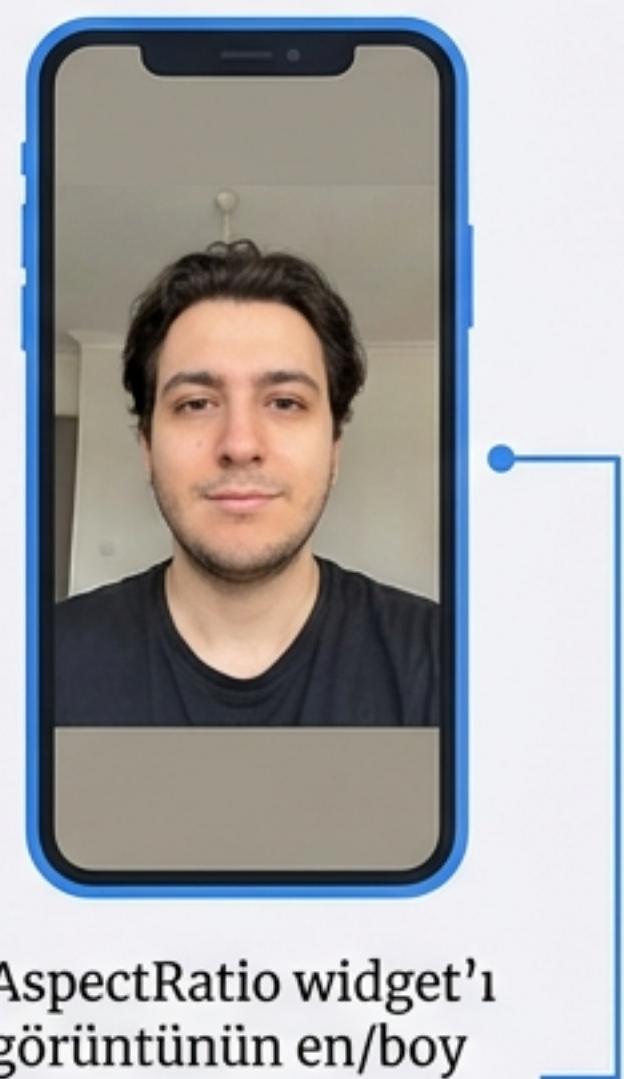
Kamera Önizlemesini Çizmek

```
if(snapshot.connectionState == ConnectionState.done) {  
    return Expanded(  
        child: Center(  
            child: AspectRatio(  
                aspectRatio: _controller.value.aspectRatio,  
                child: CameraPreview(_controller),  
            ),  
        ),  
    );  
}
```

AspectRatio Yok

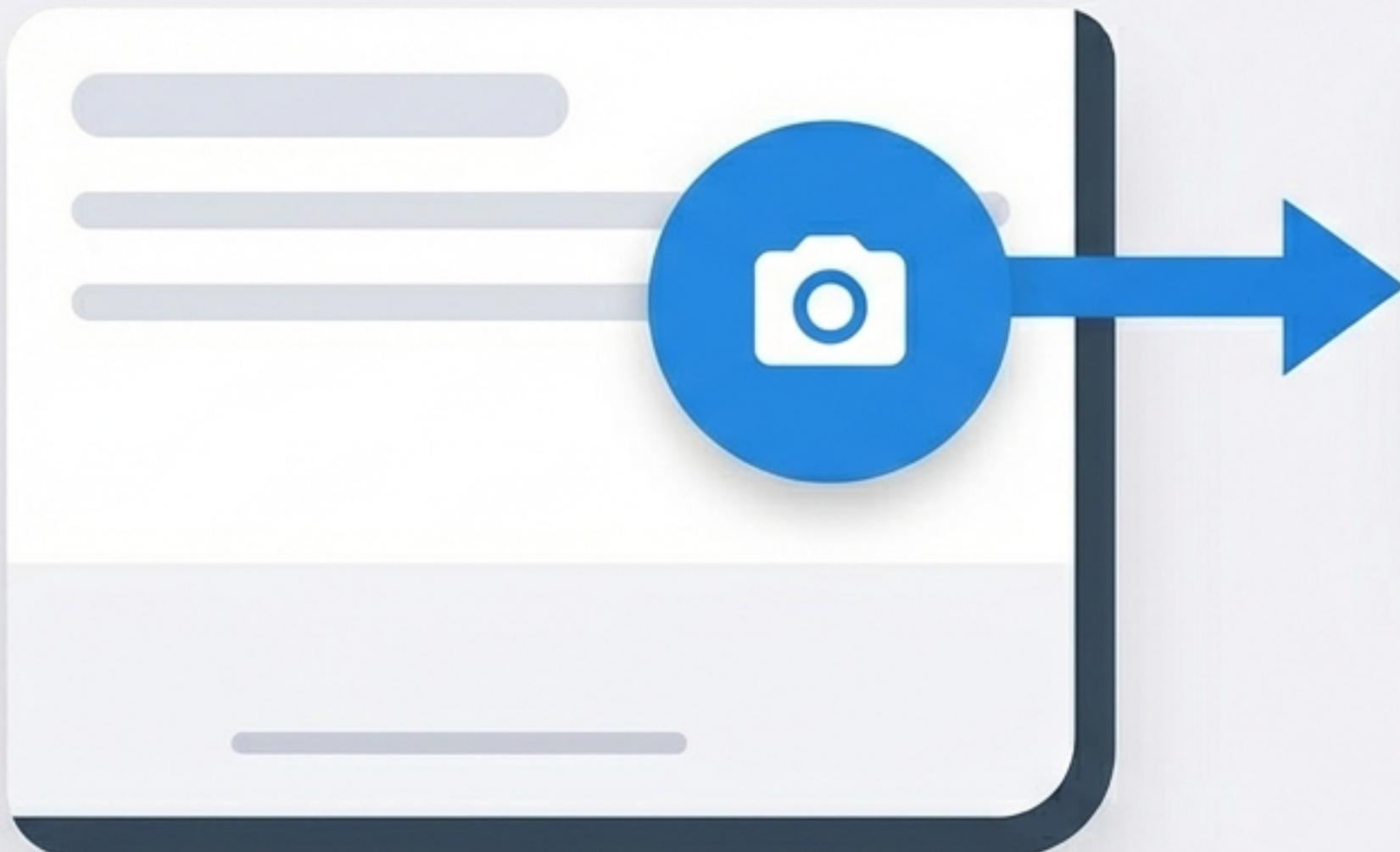


AspectRatio Var



Kullanıcı Etkileşimi

Arayüzü iş mantığından ayıriyoruz. Fotoğraf çekmek için basit bir buton ekliyoruz.



```
IconButton(  
  icon: const Icon(Icons.photo_camera),  
  onPressed: () => _takePhoto(context),  
)
```

`_takePhoto` fonksiyonu ile mantığı izole ediyoruz.

Dosya Yolu ve İsimlendirme



Dizin Bulma

```
getTemporaryDirectory()
```



İsim Üretme

```
mypic_{DateTime.now()}.png
```



Birleştirme

```
path.join(dir, name)
```

iOS ve Android dosya sistemleri farklıdır. 'path' paketi, işletim sisteme uygun güvenli bir dosya yolu oluşturur.

Fotoğrafı Yakalama Mantığı

```
void _takePhoto(BuildContext context) async {  
  // 1. Kontrolcü Hazır mı?  
  await _initController;  
  
  // 2. Dosya Yolu  
  final dir = await getTemporaryDirectory();  
  final name = 'img_${DateTime.now()}.png';  
  final fullPath = path.join(dir.path, name);  
  
  // 3. Çekim ve Kayıt  
  await _controller.takePicture(fullPath);  
  
  // 4. Geri Bildirim  
  ScaffoldMessenger.of(context).showSnackBar(  
    SnackBar(content: Text('Fotoğraf kaydedildi!'))  
  );  
}
```



Sonuç ve Özeti



Donanım ile kullanıcı arayüzü arasındaki köprüyü başarıyla kurduk. Artık bu temel üzerine filtreler, galeri veya video kaydı gibi özellikler ekleyebilirsiniz.