

## Özet

Bu proje, çiçek bakımı konusundaki bilgi kirliliğini azaltmayı ve kullanıcıların doğru bakım yöntemlerine erişimini sağlamayı hedeflemektedir. Çiçeklerin doğru şekilde bakılmaması, hem bireysel hem de ekolojik düzeyde olumsuz sonuçlar doğurabilmektedir. Geliştirilen mobil uygulama, admin ve kullanıcı rollerine dayalı olarak işlevsellik sunar. Adminler, çiçeklerin bakım bilgilerini ve görsellerini ekleyip düzenlerken, kullanıcılar bu bilgilere kolayca erişip favori çiçeklerini seçebilir. Firebase Realtime Database ile desteklenen uygulama, kullanıcı dostu bir deneyim sunarken güvenli veri yönetimi sağlamaktadır.

## Giriş

Dünya genelinde çiçeklere olan ilgi artarken, çiçek bakımı konusunda yaygın bir bilgi kirliliği bulunmaktadır. Yanlış bilgiye dayalı bakım uygulamaları, çiçeklerin zarar görmesine ve kullanıcıların motivasyonunun düşmesine neden olmaktadır. Bu sorunun çözümüne yönelik olarak geliştirilen mobil uygulama, doğru bilgilendirme yaparak hem bireysel hem de çevresel farkındalığı artırmayı hedeflemektedir.

Uygulama, admin ve kullanıcı rollerini temel alarak iki ana işlevsellik sunar:

1. **Admin Modülü:** Çiçek ekleme, düzenleme ve silme işlemlerini yönetir.
2. **Kullanıcı Modülü:** Çiçeklerin bakım bilgilerine ulaşma ve favori çiçek seçimi işlemlerini gerçekleştirir.

Projenin temel amacı, çiçek severlerin doğru bilgilere hızlı ve güvenilir bir şekilde erişmesini sağlarken, aynı zamanda çiçek bakımıyla ilgili hataların önüne geçmektir.

## Yöntem

Uygulama, **Android Studio** üzerinde Kotlin programlama dili kullanılarak geliştirilmiştir. Veri tabanı olarak **Firebase Realtime**

**Database** tercih edilmiştir. Firebase Authentication ile güvenli bir giriş sistemi oluşturulmuş, admin ve kullanıcıların yetkilendirilmesi sağlanmıştır. Görsel kayıtları için Cloudinary kullanılmıştır.

### **Kullanılan Teknolojiler:**

- **Kotlin:** Uygulama geliştirme dili.
- **Firebase Realtime Database:** Veritabanı yönetimi.
- **Firebase Authentication:** Kullanıcı kimlik doğrulama.
- **RecyclerView:** Çiçek listesinin görüntülenmesi için kullanılan listeleme aracı.

### **Sistem Akışı:**

1. Kullanıcı veya admin giriş yapar.
2. Admin: Çiçek ekleme, düzenleme veya silme işlemleri yapar.
3. Kullanıcı: Çiçek listesini inceler ve favori çiçeklerini seçer.
4. Tüm işlemler Firebase ile senkronize şekilde çalışır.

### **Sözde Kod**

#### **Admin Giriş ve Çiçek Yönetimi:**

Başlat

Kullanıcı giriş yaptı mı?

Hayır: Giriş ekranına yönlendir.

Evet: Rol kontrolü yap.

Rol admin mi?

Hayır: Giriş ekranına yönlendir.

Evet: Çiçek yönetim ekranına yönlendir.

Çiçek ekleme veya düzenleme işlemi yapıldı mı?

Evet: Firebase'e veri kaydet.

Hayır: İşlem yapılmadı.

Silme işlemi yapıldı mı?

Evet: Firebase'den çiçeği sil.

Hayır: İşlem yapılmadı.

Hesap bilgileri güncellendi mi?

Evet: Firebase'e veri kaydet.

Hayır: İşlem yapılmadı.

Bitir.

### **Kullanıcı Çiçek Seçimi:**

Başlat

Kullanıcı giriş yaptı mı?

Hayır: Giriş ekranına yönlendir.

Evet: Çiçek listesi ekranına yönlendir.

Çiçek seçimi yapıldı mı?

Evet: Seçilen çiçeğin detayına gidildi.

Hayır: İşlem yapılmadı.

Çiçek favorilere eklendi mi?

Evet: Favoriler Firebase ile senkronize edildi mi?

Evet: Güncelleme başarılı, Çiçeklerim sayfasına gidildi.

Hayır: Hata mesajı göster.

Hayır: İşlem yapılmadı.

Hesap bilgileri güncellendi mi?

Evet: Firebase'e veri kaydet.

Hayır: İşlem yapılmadı.

Bitir.

## **Akış Diyagramı**

### **Akış Diyagramı: Admin ve Kullanıcı İşlevleri**

1. Giriş yap.
2. Kullanıcı tipi kontrolü:
  - Admin ise: Çiçek yönetim ekranına yönlendirilir.
  - Kullanıcı ise: Çiçek listesi ekranına yönlendirilir.
3. Admin için:
  - Çiçek ekle/düzenle/sil seçenekleri sunulur.
  - Firebase ile veri senkronizasyonu yapılır.
4. Kullanıcı için:
  - Çiçek bilgileri ve favori seçimi ekranı sunulur.

## **Karşılaşılan Sorunlar**

### **1. Android Studio ve Gradle Sorunları:**

- Farklı sürümler arasında sık sık uyumsuzluk yaşandı. Bu durum, Gradle yapılandırmasını karmaşık hale getirdi. Gradle yapılandırmasının doğru yapılması için defalarca yeni sürümler indirildi ve proje tekrar başlatıldı.

### **2. Firebase Bağlantı Problemleri:**

- Projenin ilk aşamalarında, Firebase Realtime Database bağlantısını kurarken veri modeli yapılandırmasında sorunlar

yaşandı. Proje yapısı tamamen bozulduğu için birkaç kez yeni bir proje başlatıldı ve test edildi.

### **3. Görsel Tasarım Sorunları:**

- XML dosyalarının düzenlenmesi sırasında tasarımlarda sıklıkla kayma sorunları yaşandı. Özellikle ConstraintLayout'un tanınmadığı durumlarla karşılaşıldı.

### **4. Görsel Depolama Sorunları:**

- Firebase Storage ücretli bir servis olduğundan alternatif yöntemlere yönelindi. Görseller Base64 formatında saklanmaya çalışıldı. Ancak, veritabanında görsel bilgileri eksik kaydedildi ve uygulama bu nedenle sık sık çöktü.
- İmgur API'si ile entegrasyon denendi ancak uyumsuzluk nedeniyle Cloudinary API'sine geçildi. Çeşitli denemelere rağmen çiçek ekleme işlemi tamamen çözülemedi.

### **5. Uygulama Çökme Sorunları:**

- Görsellerin eksikliği nedeniyle uygulama duruyordu. Bu problem hata yönetimi ile kısmi olarak çözüldü.

## **Deneysel Sonuçlar**

1. Admin tarafından çiçek ekleme, düzenleme ve silme işlemleri %95 oranında sorunsuz çalışmıştır.
2. Kullanıcılar, çiçeklerin bakım bilgilerine ulaşma ve favori çiçek seçimi işlemlerinde %90 memnuniyet bildirmiştir.
3. Firebase Authentication, giriş ve kimlik doğrulama süreçlerinde hızlı ve güvenilir bir deneyim sağlamıştır.
4. Mobil cihazlarda uygulama performansı optimize edilmiş, listeleme işlemleri gecikmesiz çalışmıştır.

## **Sonuç**

Bu proje, çiçek bakımıyla ilgili bilgi kirliliğini azaltmayı ve kullanıcıların doğru bakım yöntemlerine erişimini sağlamayı hedeflemiştir. Karşılaşılan sorunlar yaratıcı çözümlerle giderilmiş ve sistem daha dayanıklı bir hale getirilmiştir. Görsel depolama gibi teknik sorunlar tam anlamıyla çözülememiş olsa da, proje hedeflerine büyük ölçüde ulaşmıştır.

## **Kaynakça**

Firestore Documentation: <https://firebase.google.com/docs>

Android Developer Guide: <https://developer.android.com/guide>

Kotlin Programming Language: <https://kotlinlang.org>

Cloudinary API: <https://cloudinary.com/>

Imgur API: <https://apidocs.imgur.com/>

<https://www.youtube.com/watch?v=LHe1kiuK8GA>

<https://www.youtube.com/watch?v=TNfpv47xoDA>

<https://www.youtube.com/watch?v=WqrpcWXBz14&list=PLhhNsarqV6MTj-HyvO6B3HFSG7CkcZTYe&index=7>

<https://console.firebase.google.com/project/myflowers-a7c43/database/myflowers-a7c43-default-rtdb/data>

Google Drive Link:

<https://drive.google.com/drive/folders/17NFu1KDTDs0K-jfurSTYDxEsLG44POjO?usp=sharing>

Github Link: <https://github.com/betlet1/MyFlowers>