

ANKARA ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ



BLM4538 - IOS İle Mobil Uygulama Geliştirme II

PROJE RAPORU

PLANTPAL – Bitki Bakımı Mobil Uygulama Projesi

İlayda Öcal

21290519

github.com/illyoci

Mayıs 2025

GİTHUB VE VIDEO BAĞLANTI LİNKLERİ

- **GitHub Frontend Dosyalarının Proje Linki:**
<https://github.com/illyocl/plantpal-react-native>
- **GitHub Backend Dosyalarının Proje Linki:**
<https://github.com/illyocl/plantpal-backend>
- **Uygulama Tanıtım Video Linki:**
<https://drive.google.com/file/d/1hHTy-h7VN49uk4oHtGN1QSH6bmnwAhm2/view?usp=sharing>
- **Database Video Linki:**
https://drive.google.com/file/d/1brNku70hGJtAJTwUNuuPNAF-7_uHYC1F/view?usp=sharing
- **Apk Dosya Linki:**
<https://drive.google.com/file/d/1MbHLFSwYyyabsuwGNFIfMRLHuX7umOGq/view?usp=sharing>

ÖZET

PlantPal, bitki bakımına ilgi duyan kullanıcıların iç ve dış mekân bitkileri hakkında bilgiye kolayca erişmelerini ve bitki bakım süreçlerini daha verimli hale getirmelerini sağlamak amacıyla geliştirilmiş bir mobil platformdur. Projenin temel amacı, kullanıcıların bitki türlerine göre bakım ihtiyaçlarına ulaşmalarını sağlarken bitki bakımına dair rehberlik sunmak ve kullanıcıların bitkilerinin bakımlarını izleyebilecekleri bir takvim oluşturmaktır.

Geliştirilen mobil uygulama, bitki bilgilerini iç ve dış mekân türlerine göre kategorize ederek kullanıcılara filtreleme imkânı sunmaktadır. Kullanıcılar, ilgili bitki kartlarını seçerek bitkilerin ihtiyaç duyduğu sulama, ışık, toprak türü gibi bakım detaylarına erişebilmektedir. Ayrıca, kullanıcı yönetimi için API tabanlı bir sistemle desteklenen uygulama, güvenli giriş ve kayıt işlemleri sunarak kullanıcı verilerinin güvenliğini sağlamaktadır.

Proje, MSSQL veri tabanı ile güvenli bir veri depolama sağlarken, .NET Framework kullanılarak API'ler aracılığıyla verilerin işlenmesi gerçekleştirilmiştir. Flutter ve Dart teknolojileriyle çapraz platform desteği sunan kullanıcı dostu bir mobil arayüz tasarlanmıştır.

Bu platform, kullanıcı dostu tasarımı ve veri güvenliğine odaklı altyapısı ile bitki bakım süreçlerini kolaylaştırmayı, bitki severlere pratik çözümler sunmayı ve çevrimiçi bitki bakım rehberliği konusunda etkili bir çözüm sağlamayı hedeflemiştir.

İÇİNDEKİLER

ÖZET	i
İÇİNDEKİLER	ii
1. GİRİŞ	1
2. TEKNOLOJİK ALTYAPI	2
2.1. Veritabanı: MSSQL (Microsoft SQL Server)	2
2.2. Backend: .NET Framework	3
2.3. Frontend: Flutter, Dart	6
3. VERİ TABANI TASARIMI VE TABLOLAR	7
4. UYGULAMANIN GELİŞTİRİLMESİ	8
4.1. Ana Sayfa	9
4.2. Bitkilerim Sayfası	10
4.3. Bitki Detay Kartları	11
4.4. İç Mekan Bitkileri Sayfası	12
4.5. Dış Mekan Bitkileri Sayfası	13
4.6. Hakkında Sayfası	14
4.7. Gübreler Sayfası	15
4.8. Bakım Takvimi Sayfası	16
4.9. Giriş ve Kayıt Sayfası	17
4.10. Bitki Arama ve Filtreleme Sayfası	18
SONUÇ	19
EKLER	20
KAYNAKLAR	21

1. GİRİŞ

Doğayla iç içe olma isteği, şehir hayatının karmaşasında insanların bitki bakımına olan ilgisini artırmıştır. Bitkiler, hem yaşam alanlarını güzelleştiren hem de insanlara huzur veren canlılardır. Ancak, farklı bitki türlerinin doğru şekilde bakım gereksinimlerini karşılamak, özellikle bitki bakımı konusunda deneyimsiz kullanıcılar için zorlu olabilmektedir.

PlantPal, bu ihtiyaca yönelik olarak geliştirilmiş, kullanıcıların bitki bakım süreçlerini kolaylaştırmayı amaçlayan bir mobil uygulamadır. Proje kapsamında, kullanıcıların bitki türleri ve gübreler hakkında detaylı bilgi edinmelerine, doğru bakım yöntemlerini öğrenmelerine ve bitki kartları üzerinden bakım talimatlarına kolayca ulaşmalarına olanak sağlanmıştır.

Bu çalışmada, PlantPal uygulamasının tasarım süreci, kullanılan teknolojiler ve sağlanan işlevsellikler detaylı olarak ele alınmıştır. Kullanıcı deneyimi, veri güvenliği ve dinamik içerik yönetimi gibi temel unsurların nasıl entegre edildiği açıklanarak, projenin kullanıcı ihtiyaçlarına sunduğu çözümler üzerinde durulmuştur.

PlantPal projesi, kullanıcıların bitki bakım süreçlerini kolaylaştırmayı ve mobil uygulama kullanımını verimli bir bilgi merkezi haline getirmeyi amaçlayan iki ana hedefe odaklanmıştır:

- **Kullanıcı Deneyimi:** Uygulama, bitki bakımına ilgi duyan kullanıcılar için hızlı ve kolay erişim sağlamak amacıyla kullanıcı dostu bir arayüzle tasarlanmıştır. Kullanıcılar, bitki türlerini iç ve dış mekân olarak filtreleyebilir ve seçilen bitkinin bakım detaylarına kolayca ulaşabilirler.
- **Bitki Bakım Takibi:** Giriş yapan kullanıcılar, platformda kişisel bir takvim oluşturarak kendi bitkilerini ekleyebilir ve bu takvim üzerinden sulama, gübreleme gibi bakım hatırlatmalarını takip edebilir ve bildirim alabilirler. Bu özellik, kullanıcıların bitki bakım rutinlerini daha düzenli ve verimli bir şekilde yönetmelerine imkân tanımaktadır.

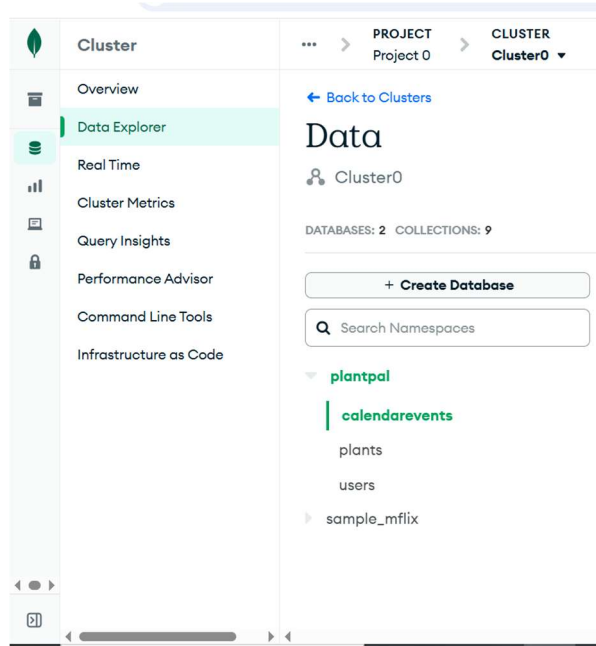
2. TEKNOLOJİK ALTYAPI

Bu projede, veritabanı yönetimi, backend işlevselliği, frontend tasarımı ve güvenlik önlemleri gibi çeşitli alanlarda güncel ve güçlü teknolojiler kullanılmıştır. Her bir teknoloji, projenin gereksinimlerine ve kullanıcı ihtiyaçlarına göre seçilmiştir.

2.1. Veritabanı: MongoDB Atlas (NoSQL Bulut Veritabanı)

PlantPal projesinde, kullanıcılar, bitkiler ve bakım takvimi gibi kritik veriler, bulut tabanlı NoSQL veritabanı olan MongoDB Atlas üzerinde güvenli bir şekilde saklanmaktadır.

MongoDB, belge tabanlı (JSON yapısında) veri modeliyle esnek ve hızlı erişim imkanı sunarken, mongoose kütüphanesi aracılığıyla Node.js backend ile bütünleşik olarak kullanılmıştır.



Şekil 2.1. Veri Tabanı Table yapısı

- **Bitkiler Tablosu:** Bitkilerin adı, türü (iç mekân/dış mekân), bakım gereksinimleri (sulama sıklığı, ışık ihtiyacı vb.) gibi bilgileri içerir. Kullanıcılar bu bilgiler sayesinde bitkileri filtreleyerek ihtiyaç duydukları bakım detaylarına ulaşabilirler.
- **Kullanıcılar Tablosu:** Kullanıcıların kişisel bilgileri (ad, soyad, e-posta vb.) ve sistemde kaydedilen bitki listeleri saklanır. Kullanıcılar sisteme giriş yaparak bitki bakım takvimlerini düzenleyebilirler.

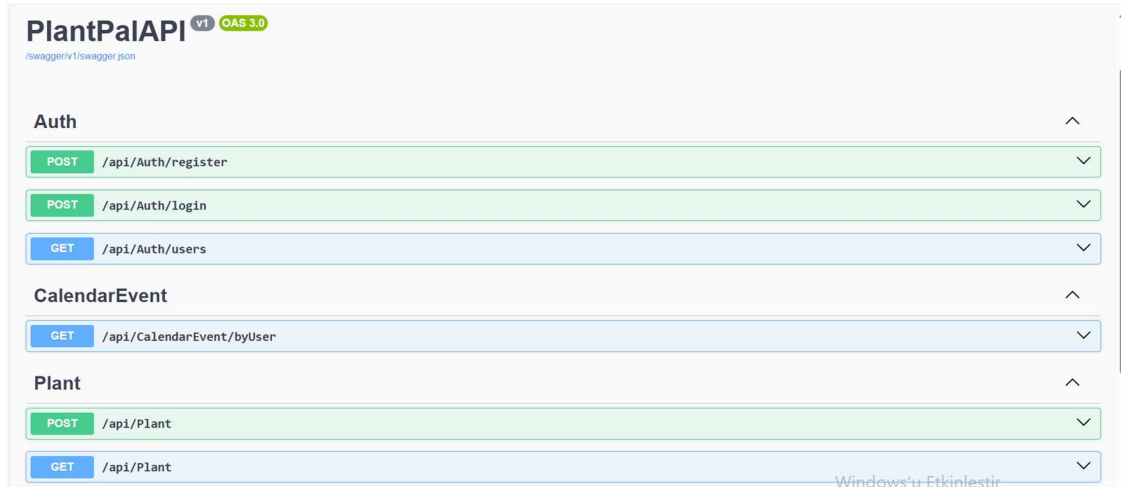
- **Takvim Tablosu:** Kullanıcıların ekledikleri bitkilerin bakım zamanlamaları bu tabloda saklanır. Sulama, gübreleme gibi bakım hatırlatmaları bu tablo aracılığıyla takip edilir ve kullanıcıların bakım süreçleri daha düzenli hale getirilir.

Bu yapı sayesinde PlantPal, kullanıcıların bitki bakım ihtiyaçlarını kolayca yönetmelerine olanak tanıyan güvenli ve verimli bir sistem sunmaktadır.

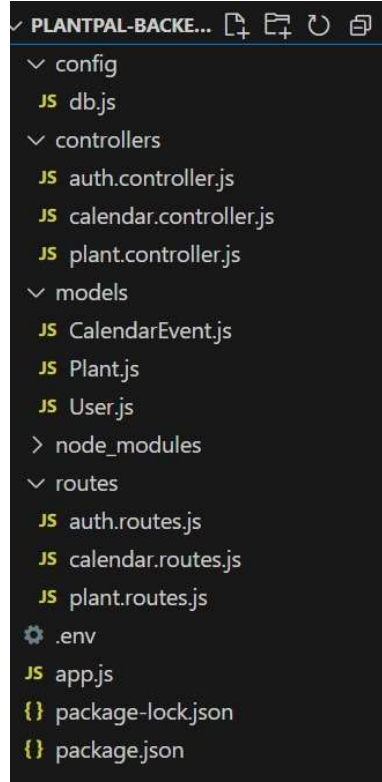
2.2. Backend: Node.js

Proje backend tarafı, Node.js ve Express framework'ü kullanılarak geliştirilmiştir. Node.js, hızlı ve etkin bir sunucu ortamı sağlarken, Express framework'ü ise API geliştirme süreçlerini kolaylaştırarak proje yapısının daha modüler ve yönetilebilir olmasına yardımcı olmuştur.

- **API Geliştirme:** Veritabanına eklenen, güncellenen ve silinen veriler, geliştirilen RESTful API'ler aracılığıyla işlenmektedir. Node.js backend, API güvenliğini artırmak amacıyla kimlik doğrulama, veri doğrulama ve şifreleme gibi mekanizmalar kullanmaktadır. API'ler sayesinde, kullanıcılar bitki bilgilerini görüntüleyebilir, takvim oluşturabilir ve kendi kullanıcı bilgilerini güncelleyebilir.



Şekil 2.2. Swagger yapısı



Şekil 2.3. Backend API dosya yapısı

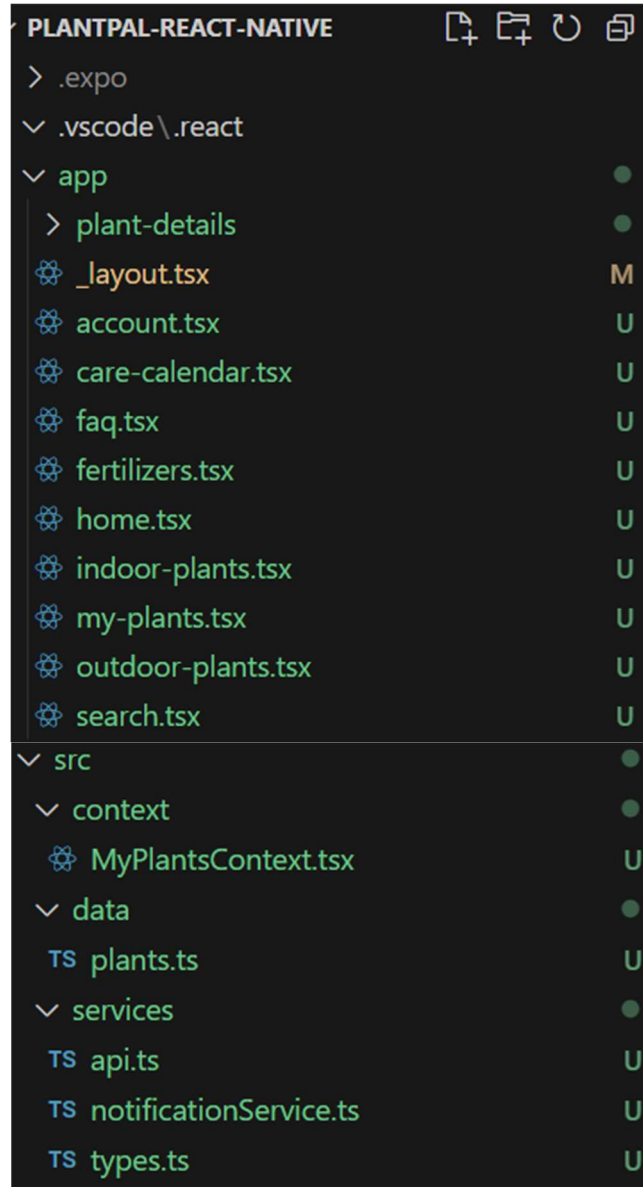
- **config/db.js:** Bu dosya, MongoDB veritabanı ile Node.js uygulaması arasında bağlantıyı kurmak için kullanılır.
- **controllers/ authController.js:** Bu dosya, kullanıcının kayıt ve giriş işlemleri için kullanılan API kontrolüdür. Şifrelerin güvenli bir şekilde saklanıp doğrulanmasını sağlar. Ayrıca, kullanıcı bilgilerini veri tabanına kaydeder.
- **controllers/calender.controller.js:** Bu dosya, kullanıcılara ait takvim olaylarının yönetilmesi için oluşturulmuş bir API kontrolüdür. Kullanıcıların takvimdeki olaylarına erişim sağlamak amacıyla yazılmıştır.
- **controllers/plant.controller.js:** Bu dosya, kullanıcıların bitkilerini eklemeleri, listelemeleri ve bitkilerine ait takvim olayları eklemeleri için bir API kontrolü sağlar.
- **models/CalendarEvent.js:** Bu dosya, CalendarEvent modelini tanımlar. Bu model, her bir takvim olayı (örneğin, bitki sulama, gübreleme vb.) için gerekli bilgileri içerir.
- **models/Plant.js:** Bu model, Bitkilerin veritabanındaki yapısını tanımlar. bitki ekleme, listeleme gibi işlemlerde kullanılır.
- **models/User.js:** Bu model, kullanıcıların sisteme kaydolması, giriş yapması ve diğer işlemleri için temel bilgileri sağlar.

- **routes/auth.routes.js:** Kullanıcının kayıt olma, giriş yapma gibi işlemlerini yönlendiren endpoint'leri tanımlar. /api/auth yolundaki HTTP isteklerini auth.controller.js dosyasındaki işlemlerle eşleştirir.
- **routes/calendar.routes.js:** Takvimle ilgili HTTP isteklerini karşılar. /api/calendar yolundaki istekleri calendar.controller.js ile bağlantılar.
- **routes/plant.routes.js:** Bitkilere ait API uç noktalarını yönetir. /api/plants yoluna gelen istekler burada tanımlanır ve plant.controller.js ile eşleştirilir.
- **.env:** Uygulama içindeki hassas bilgiler (.env üzerinden alınan MONGO_URI, port bilgisi vb.) burada saklanır. Bu sayede ortam değişkenleri güvenli ve merkezi bir yapı içinde yönetilir.
- **app.js:** Uygulamanın ana dosyasıdır. Express uygulaması başlatılır, middleware'ler tanımlanır, yönlendirmeler (routes) yüklenir ve sunucu portu dinlemeye başlar.

2.3. Frontend: React Native

PlantPal mobil uygulamasının frontend geliştirmesi, kullanıcı dostu ve etkileşimli bir deneyim sunmak amacıyla React Native framework'ü ve JavaScript (TypeScript) kullanılarak gerçekleştirilmiştir. Bu teknoloji seçimi, uygulamanın Android platformlarda sorunsuz çalışmasını sağlamıştır.

- **Kullanıcı Arayüzü:** React Native'in zengin bileşen (component) yapısı sayesinde, tarım temalı ve kullanıcı dostu bir tasarım oluşturulmuştur. Kullanıcılar, bitki bakım bilgilerine kolayca erişip bitkileri türlerine göre filtreleyebilmektedir.
- **Responsive Tasarım:** Uygulama, farklı cihaz boyutlarına otomatik olarak uyum sağlayarak kullanıcı deneyimini optimize etmektedir.
- **Dinamik İçerik:** JavaScript ile desteklenen yapı, kullanıcıların etkileşimlerine anlık yanıt veren dinamik içeriklerin geliştirilmesini mümkün kılmıştır.

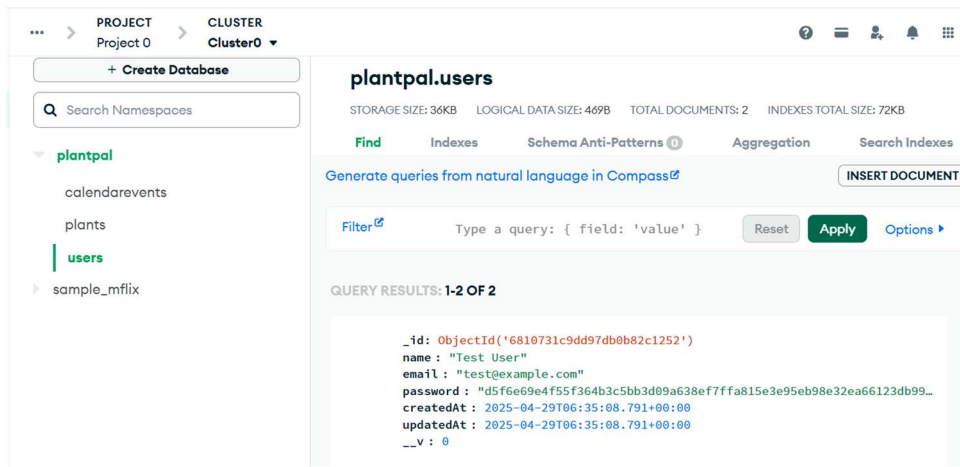


Şekil 2.4. Frontend dosya yapısı

3. VERİ TABANI TASARIMI VE TABLOLAR

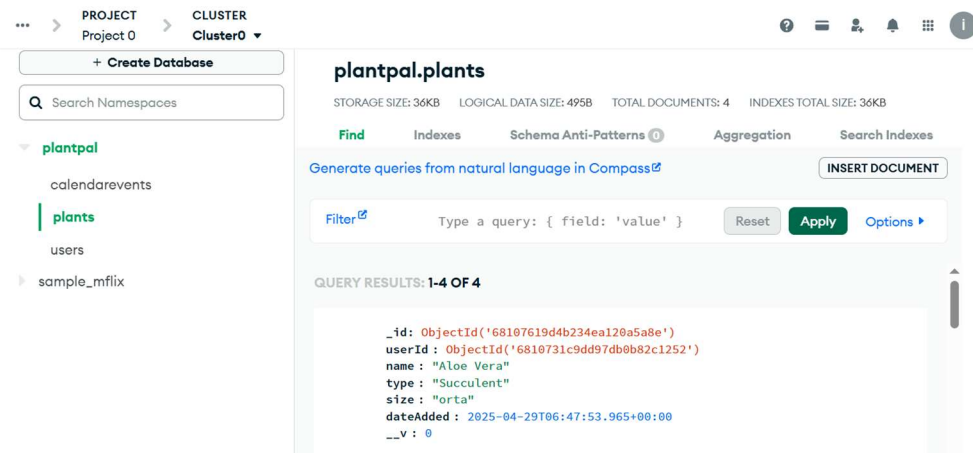
Projede kullanılan veritabanı yapısı, ilişkisel veritabanı modeline dayalı olarak tasarlanmıştır. Veritabanında üç ana tablo yer almaktadır: Bitkiler, Kullanıcılar ve Takvim.

- **Kullanıcılar Tablosu:**



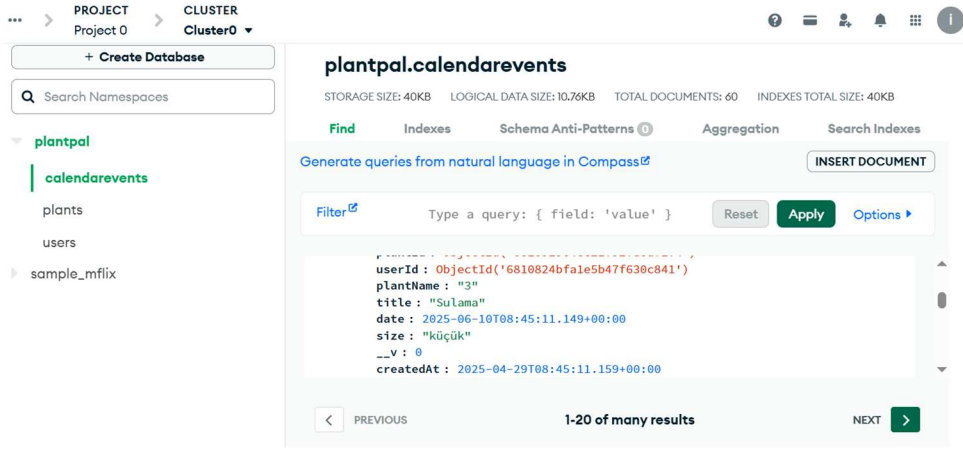
Şekil 3.1. Veri tabanına kayıtlı bir kullanıcı

- **Bitkiler Tablosu:**



Şekil 3.2. Veri tabanına kayıtlı bir bitki

- **Takvim Tablosu:**



Şekil 3.3. Veri tabanına kayıtlı bir bitkinin bakım bilgileri

4. UYGULAMANIN GELİŞTİRİLMESİ

Bu bölümde, Projenin her bir sayfası ve ilgili fonksiyonları detaylı bir şekilde açıklanacaktır. Her sayfa, kullanıcıların ihtiyaçlarına göre işlevsel ve erişilebilir olacak şekilde tasarlanmıştır.

- **app/ layout.tsx:**

Bu dosya, React Native + Expo Router yapısında uygulamanın temel navigasyon ve global ayarlarını tanımlayan ana düzen (layout) bileşenidir. Ana tema ayarları, başlangıç rotası ve diğer sayfa yönlendirmeleri burada tanımlanır. Ayrıca, uygulama başlatılmadan önce bildirim servisinin (NotificationService) başlatılması sağlanır.

- **src/context/MyPlantContext.tsx:**

Bu dosya, kullanıcıların eklediği bitkileri uygulama genelinde yönetmek için oluşturulmuş bir React Context yapısıdır. plants adlı dizi içinde bitkiler tutulur ve addPlant fonksiyonu ile yeni bitkiler eklenir. MyPlantsProvider bileşeni ile bu veri, tüm ekranlara aktarılır. useMyPlants hook'u sayesinde farklı bileşenler bu verilere kolayca erişebilir. Bu yapı, bitki verilerinin tek bir merkezden yönetilmesini sağlar.

- **src/data/plants.ts:**

Bu dosya, uygulamada kullanılan bitki verilerini statik olarak tanımlar. Her bitki için adı, resmi, açıklaması, sulama ve ışık ihtiyacı gibi temel bilgiler ile isteğe bağlı olarak botanik adı, toprak türü, bakım ipuçları ve karşılaşılabilecek zararlılar/hastalıklar gibi detaylar eklenmiştir. Ayrıca slugify fonksiyonu kullanılarak bitki adları URL uyumlu hale getirilir. Bu yapı sayesinde bitki kartları ve detay sayfaları dinamik bir şekilde oluşturulur.

src/sevices/api.ts:

Bu dosya, mobil uygulama ile backend API arasındaki veri iletişimini yönetir. axios kullanılarak, bitkiler, kullanıcılar ve takvim etkinlikleri için HTTP istekleri (GET, POST, PUT, DELETE) gerçekleştirilir. Böylece frontend ile backend arasında güvenli ve düzenli bir veri akışı sağlanır.

src/services/notificationService.ts:

Bu dosya, Expo Notifications modülünü kullanarak uygulama içinde yerel bildirimler göndermek için oluşturulmuştur. Bildirim izni kontrol edilir, zamanlanmış hatırlatıcı bildirimler ayarlanır ve kullanıcıların bakım zamanı gibi önemli konularda bilgilendirilmesi sağlanır.

src/services/types.ts:

Bu dosyada, uygulamada kullanılan veri yapıları için TypeScript ile tip tanımlamaları yapılmıştır. Bitkiler, kullanıcılar ve takvim etkinlikleri gibi nesneler için güvenli veri modeli oluşturularak uygulama genelinde kodun daha hatasız ve okunabilir olması sağlanmıştır.

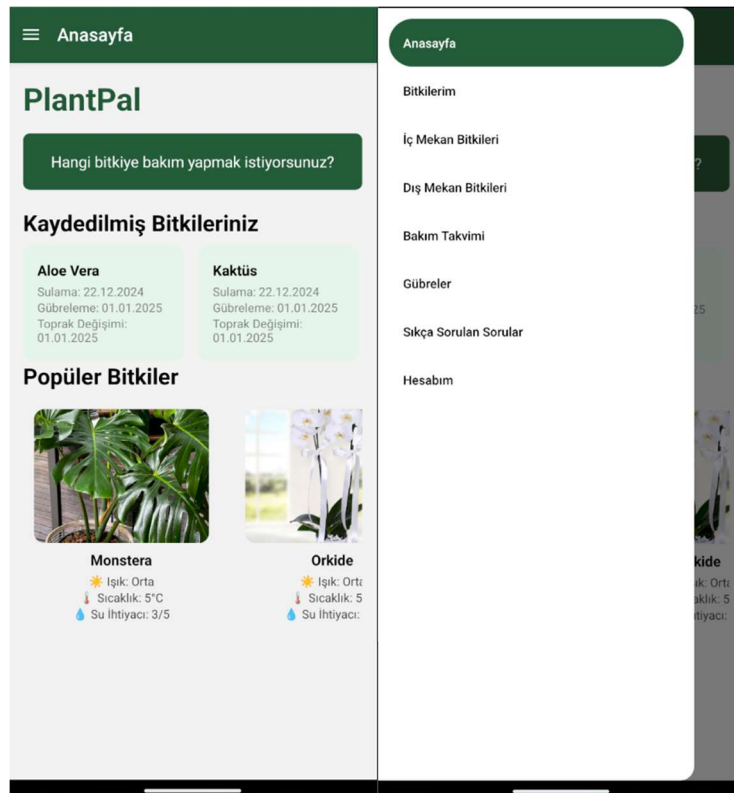
4.1. Ana Sayfa:

Ana sayfa (home.tsx), Bu dosya, PlantPal uygulamasının ana sayfasını tanımlar ve kullanıcıların bitkilerle ilgili bilgilere kolayca erişmesini sağlayan bir arayüz sunar.

Kullanıcı dostu bir tasarım ile geliştirilen ana sayfa, şu önemli bileşenlerden oluşur:

- **Yan Menü (Drawer):** Kullanıcıların uygulamanın farklı bölümlerine erişimini sağlar.
- **Arama Barı:** Kullanıcıların bilgi almak istedikleri bitkiye direkt ulaşmalarına olanak tanır.
- **Popüler Bitkiler Köşesi:** En çok aratılan bitkilerin bitki kartlarından oluşan bir kısımdır.
- **Kaydedilmiş Bitkiler:** Kullanıcının daha önce kaydettiği bitkiler, sulama, gübreleme ve toprak değişim tarihleri ile birlikte listelenir.

Ana sayfa, tüm kullanıcıların ihtiyaç duyduğu temel yönlendirmeleri sağlamak için geliştirilmiştir ve sitenin diğer bölümlerine erişimi kolaylaştırır.



Şekil 4.1. Ana Sayfa ve Yan Menü

4.2. Bitkilerim Sayfası:

Bu sayfa (my-plants.tsx), Bu dosya, PlantPal uygulamasında kullanıcıların kendi bitkilerini kaydedebileceği bir sayfa oluşturur. Kullanıcılar, bitki adı, türü ve boyutunu girerek bitkilerini ekleyebilir ve listeleyebilirler.

Android Emulator - Medium_Phone_API_35:5554

≡ Bitkilerim

Bitki Ekle

Bitki Adı

Örn: Menekşe

Bitki Türü

Aloe Vera

Bitki Boyutu

Küçük

BITKİYİ EKLE

Eklenen Bitkiler

Henüz hiç bitkiniz yok.

Şekil 4.2. Bitki ekleme sayfası

4.3. Bitki Detay Kartları:

Bitki Kartları, kullanıcıların seçtikleri bitki hakkında daha ayrıntılı bilgi aldığı sayfadır. Burada “Bakım İpuçları, Çiçeklenme Süreci, Zararlılar/Hastalıklar” gibi çeşitli konu başlıkları altında detaylı bilgiler bulunur.



Monstera

Monstera, tropikal ormanlarda yetişen popüler bir iç mekan bitkisidir. Geniş, keskin yapraklarıyla ünlüdür. İç mekan dekorasyonunda sıkça tercih edilir.

 Su İhtiyacı: Her hafta 1 kez sulama

 Işık İhtiyacı: Orta dolaylı güneş ışığı

 Ortam: Tropikal, nemli ortam

 Botanik Adı: Monstera Deliciosa

Bakım İpuçları

- Monstera'yı bol dolaylı güneş ışığında tutun.
- Toprağın iyi drene olduğundan emin olun.
- Yüksek nem seviyeleri Monstera için önemlidir.

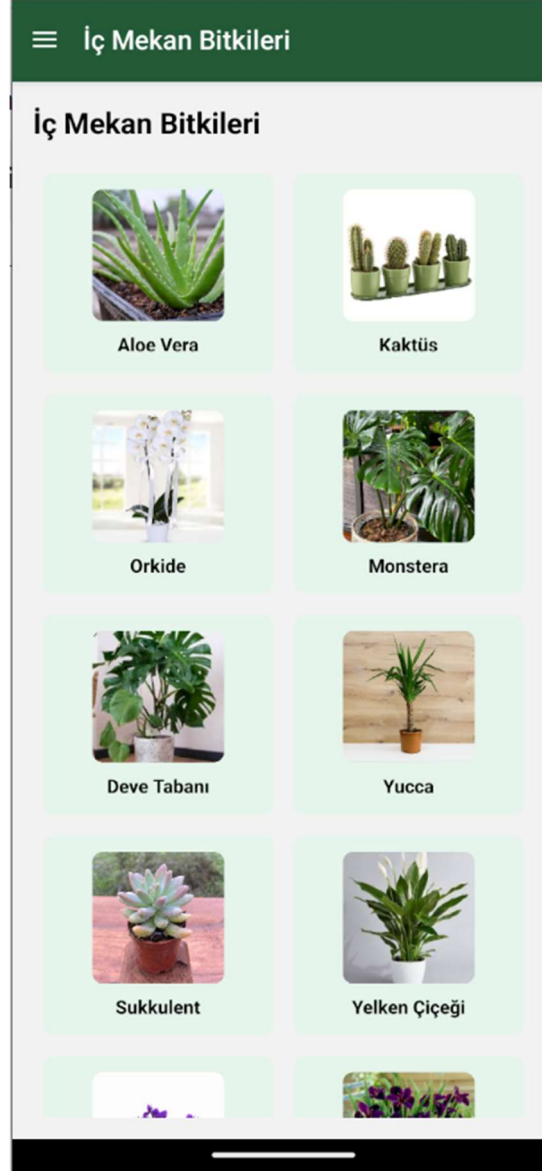
Zararlılar ve Hastalıklar

Zararlı/Hastalık	Tip	Belirtiler	Çözüm
Kırmızı Örümcekler	İnsansız	Yapraklarda ince ağlar, yaprakların sararması	Yüksek nem ile bitkiyi sulamak ve yaprakları su ile yıkamak.
Kök Çürüklüğü	Hastalık	Köklerin çürümesi; bitkinin zayıflaması	Toprağın iyi drene olduğundan emin olun.

Şekil 4.3. Örnek bir bitki detay sayfası

4.4. İç Mekan Bitkileri Sayfası:

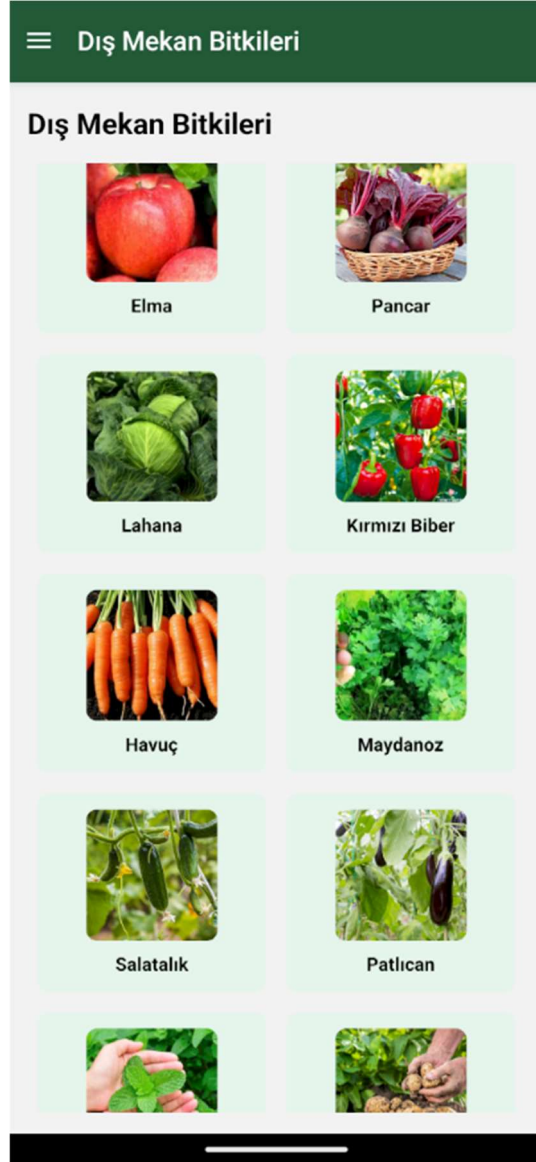
İç mekan bitkiler sayfası (indoor-plants.tsx), kullanıcılara kolaylık olması amacıyla 12 adet evde beslenebilen, iç mekan bitkisinin detay kartlarının bulunduğu sayfadır.



Şekil 4.4. İç Mekan Bitkileri sayfası

4.5. Dış Mekan Bitkileri Sayfası:

Dış mekan bitkiler sayfası (outdoor-plants.tsx), kullanıcılara kolaylık olması amacıyla 12 adet bahçede beslenebilen, dış mekan bitkisinin detay kartlarının bulunduğu sayfadır.



Şekil 4.5. Dış Mekan Bitkileri sayfası

4.6. Hakkında Sayfası:

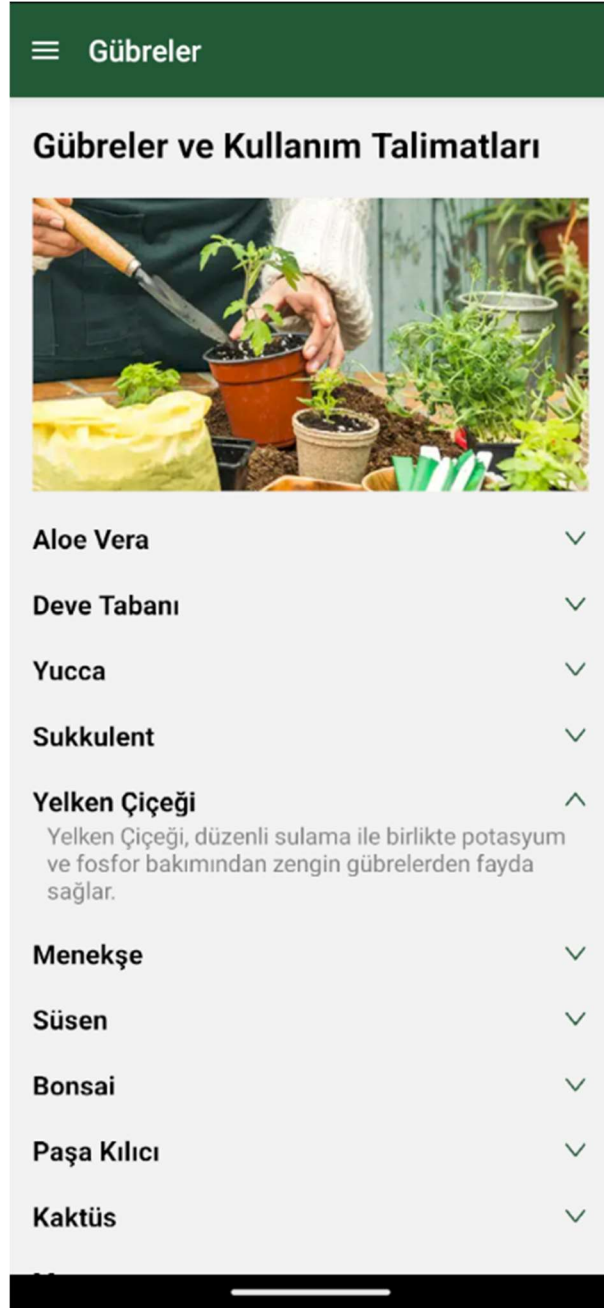
Hakkında sayfası (faq.tsx), kullanıcıların bitki bakımına dair yaygın sorularına yanıt bulabileceği ve site hakkında genel bilgilere erişebileceği SSS (Sıkça Sorulan Sorular) bölümünü barındırmaktadır. Girişte bir görsel kullanılarak dikkat çekmek amaçlanmış ve Kullanıcıların her bir soruya tıklayarak ilgili cevabı genişletmesini veya daraltmasını sağlayan akordeon tarzı tasarım uygulanmıştır. Bu sayede içerik daha derli toplu ve kullanıcı dostu hale getirilmiştir.



Şekil 4.6. Hakkında sayfası ve örnek olarak açılmış bir soru

4.7. Gübreler Sayfası:

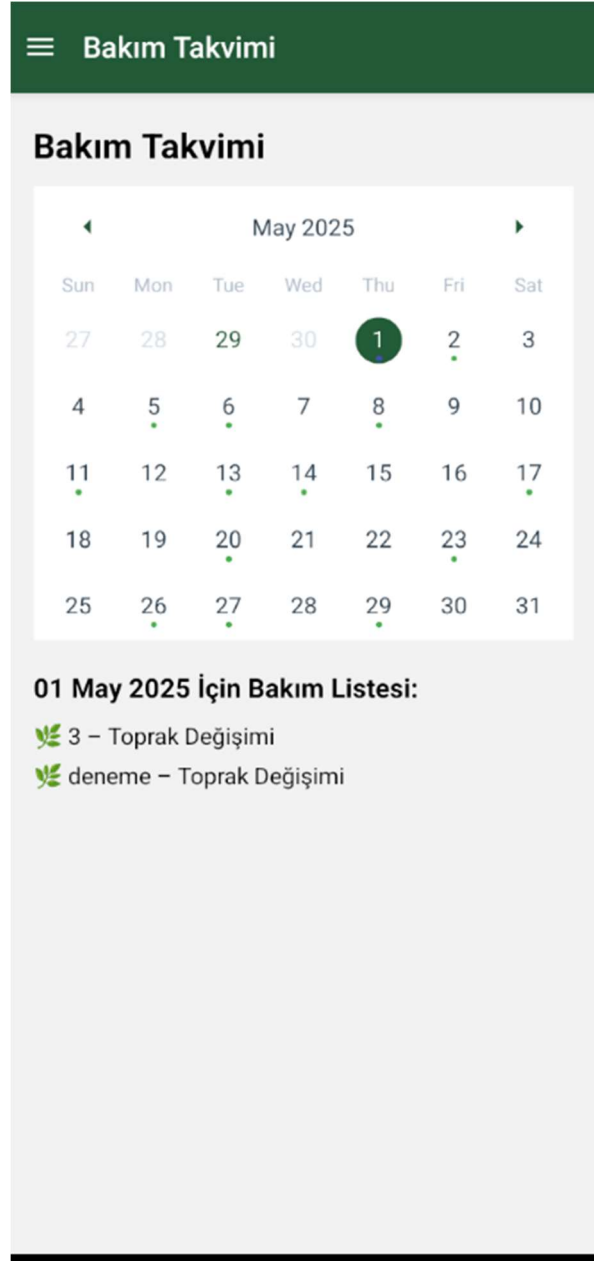
Bu dosya, PlantPal uygulamasında bitkilerin gübreleme bilgilerini ve kullanım talimatlarını içeren bir sayfa (fertilizer.tsx) sunar. Kullanıcılar, farklı bitkiler için uygun gübreler ve kullanım talimatlarını burada görebilirler.



Şekil 4.7. Gübreler sayfası ve örnek olarak açılmış talimatlar

4.8. Bakım Takvimi Sayfası:

Bakım Takvimi Sayfası (care-calender.tsx), PlantPal uygulamasındaki Bakım Takvimi sayfasını yönetir. Sayfa, kullanıcılara bitkilerinin bakımını takip etme imkanı sunar, özellikle sulama, gübreleme ve toprak değişimi gibi işlemleri hatırlatır, bildirim gönderir.



Şekil 4.8. Giriş yapmış bir kullanıcının Bakım Takvimi sayfası ve bildirim

4.9. Giriş ve Kayıt Sayfası:

Bu dosya (account.tsx), PlantPal uygulamasının Hesap Sayfasını tanımlar ve kullanıcının giriş yapmasını veya yeni bir hesap oluşturmasını sağlar. Sayfa, iki ana moddan biri olan Giriş Yap veya Kayıt Ol ekranlarını içerir.

Kayıt Ol

KAYIT OL

Zaten hesabınız var mı? [Giriş Yap](#)

Demo Hesap: demo@plantpal.com / 123456

Giriş Yap

GİRİŞ YAP

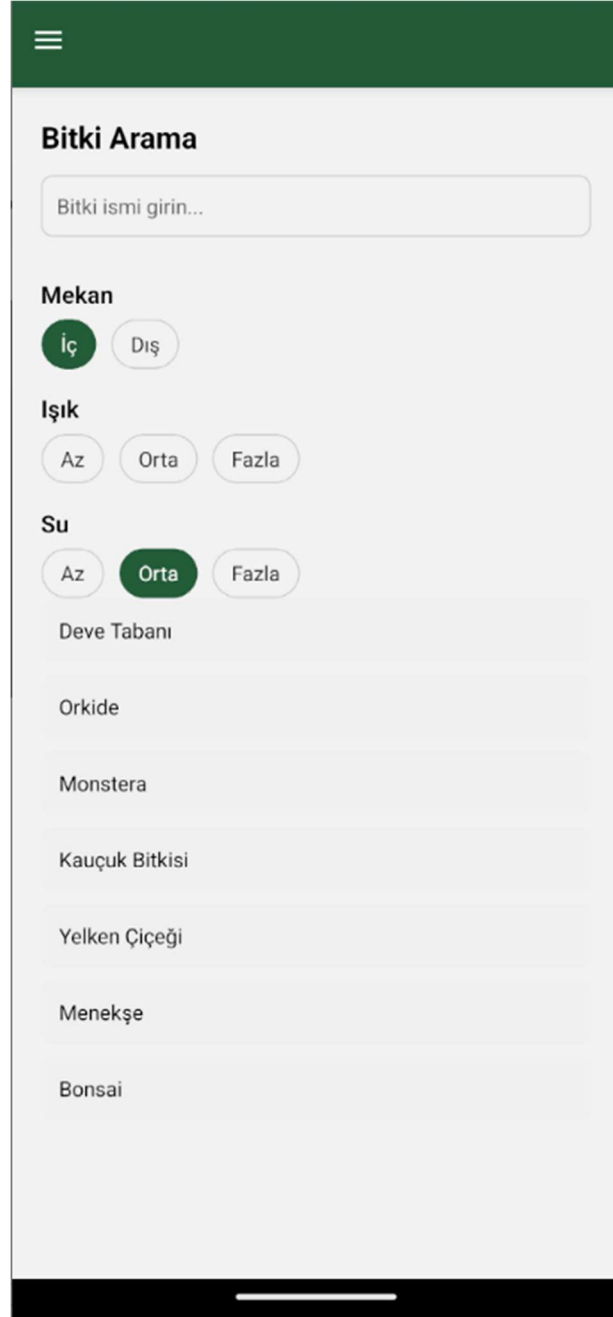
Hesabınız yok mu? [Kayıt Ol](#)

Demo Hesap: demo@plantpal.com / 123456

Şekil 4.9. Giriş ve Kayıt sayfası

4.10. Bitki Arama ve Filtreleme Sayfası:

Bu sayfa (search.tsx), Kullanıcının uygulama içinde arama yaparak bitki isimlerini bulabilmesini ve her bir bitkinin detay sayfasına yönlendirilmesini sağlar.



Şekil 4.10. Filtreleme sayfası

SONUÇ

PlantPal Bitki Bakımı Mobil Uygulama Projesi, belirlenen hedefler doğrultusunda başarıyla tamamlanmış ve kullanıcı dostu bir platform oluşturulmuştur. Geliştirilen sistem sayesinde kullanıcılar bitki bilgilerine kolayca ulaşabilir, bitki ekleyebilir ve bakımlarını düzenli bir şekilde planlayabilirler. Takvim entegrasyonu sayesinde bitki bakımı ile ilgili önemli tarihleri takip edebilmek ve bildirim alabilmek kullanıcı deneyimini artırmıştır.

Projede kullanılan teknolojiler, verilerin güvenli şekilde işlenmesi, hızlı erişim sağlanması ve kullanıcı dostu bir deneyim sunulması amacıyla dikkatle seçilmiştir. API entegrasyonu ile veri iletişimi etkin hale getirilmiş, backend kısmında Node.js, frontend kısmında React Native teknolojileri kullanılarak kullanıcı odaklı bir uygulama tasarlanmıştır.

KAYNAKLAR

Meta Platforms, Inc. (2024). *React Native Documentation*. <https://reactnative.dev/>

OpenJS Foundation. (2024). *Node.js Documentation*. <https://nodejs.org/>

MongoDB, Inc. (2024). *MongoDB Manual*. <https://www.mongodb.com/docs/>

TypeScript Team. (2024). *TypeScript Handbook*. <https://www.typescriptlang.org/docs/>

Expo. (2024). *Expo Documentation*. <https://docs.expo.dev/>

Udemy. (2024). *React Native ile Mobil Uygulama Geliştirme Kursu*. Udemy.

Udemy. (2024). *Node.js & Express.js ile RESTful API Geliştirme*. Udemy.

Johnson, S. (2021). *Full Stack Development with React and Node.js*. Packt Publishing.