

**ANKARA ÜNİVERSİTESİ MÜHENDİSLİK
FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**



**BLM4538 - IOS İle Mobil Uygulama Geliştirme II
Görev Yönetim Uygulaması
Süleyman Sertaç Üst – 21290682**

Github: ss-ust

Video:

<https://youtube.com/shorts/d2qaGzqCqt8>

https://github.com/ss-ust/react_taskManagement

1. Açıklama

Uygulamanın veritabanı Render.com üzerinden deploy edilmiş durumda. Uzun süre kullanılmadığı zaman bu deployu bekleme moduna alınabiliyor. O nedenle ilk kayıt / giriş işlemi yaptıktan sonra birkaç saniye beklemek gerekebiliyor.

Git'te "Lokal" ve "APK" olmak üzere iki klasör var:

- Lokal: Lokal veritabanı ile çalışıyor. Backend kaynak koduna buradan erişilebilir. Index.js çalıştırıldığı zaman backend çalışıyor.
- APK: APK'nın derlendiği kaynak kodlar. Lokalden tek farkı local veritabanı yerine Render'daki veritabanına erişiyor. Kullandığı backend lokal ile birebir aynı olsa da yine de görmek isterseniz şu repoda:

Render Backend Repo'su: <https://github.com/ss-ust/task-manager-backend>

2. Proje Tanımı

Bu proje, kullanıcıların görevlerini etkili bir şekilde yönetmelerini sağlayan bir görev yönetim (task management) uygulamasıdır. Kullanıcılar, hesap oluşturduktan sonra yeni görevler ekleyebilir, bu görevleri kategorilere ayırabilir, ilerleme durumlarını takip edebilir ve diğer kullanıcılarla işbirliği yapabilirler. Yöneticiler, sistemdeki tüm görevleri ve kullanıcıları yönetebilirken, kullanıcılar sadece kendi görevlerini düzenleyebilir ve takip edebilir. Görevler; öncelik seviyelerine, kategorilerine ve ilerleme durumlarına göre organize edilebilir.

3. Projede Yapılanlar

3.1. Temel Özellikler:

- Kullanıcı kayıt ve giriş işlemleri
- Token tabanlı yetkilendirme
- Rol bazlı erişim kontrolü
- Görev oluşturma, güncelleme, silme, atama
- Görev filtreleme
- Yorum ekleme, güncelleme, silme

3.2. Sistem Mimarisi:

React Native ile geliştirilen mobil istemci, görev yönetim sistemine ait sunucu tarafı (backend) ile haberleşmek için <https://task-manager-backend-2k59.onrender.com> adresine RESTful API istekleri gönderir. Bu istemci, kullanıcıların görev oluşturma, görevleri görüntüleme, yorum yapma, kullanıcı listelerini görme gibi çeşitli işlemleri gerçekleştirmesine olanak tanır.

Sistem, kimlik doğrulama mekanizması olarak JWT (JSON Web Token) tabanlı bir yapı kullanır. Kullanıcı sisteme giriş yaptıktan sonra backend tarafından bir JWT token üretilir ve istemciye iletilir. Bu token, sonraki tüm isteklerde Authorization başlığı altında "Bearer <token>" formatında gönderilir. Backend, bu token'ı doğrulayarak kullanıcının kimliğini ve yetkisini kontrol eder.

Kullanıcılar sistemde iki farklı rol ile tanımlanabilir:

- **admin:** Tüm görevleri görüp düzenleyebilir, kullanıcı listesini görüntüleyebilir ve görev atamaları yapabilir.
- **user:** Sadece kendisine atanan görevleri görebilir ve yorum yapabilir.

Tüm API erişim noktaları kullanıcının sahip olduğu role göre yetkilendirilmiştir. Örneğin bir user rolündeki kullanıcı başka bir kullanıcıya görev atayamazken, bir admin kullanıcı tüm sistemde yetkilidir. Bu sayede güvenli ve kontrollü bir işlem akışı sağlanmaktadır.

3.3. Backend, API Rotaları ve Veritabanı Rotaları:

Uygulama, Express.js çatısı altında geliştirilmiş bir Node.js sunucusu kullanır. Sunucu, görev yönetim sistemine ait tüm iş mantığını ve veri erişim katmanını barındırır. RESTful mimariye uygun olarak yapılandırılmış bu sunucu mobil istemciden gelen HTTP isteklerine yanıt verir.

Veri yönetimi için better-sqlite3 adlı senkron, hızlı ve verimli SQLite bağlayıcısı tercih edilmiştir. Bu sayede düşük kaynak tüketimiyle birlikte veritabanına doğrudan erişim ve işlem yapılması sağlanmıştır. Tüm kullanıcı, görev ve yorum verileri SQLite veritabanında tutulur.

Sistemde kullanılan temel rotalar şunlardır:

Route	Method	Açıklama
/api/auth/register	POST	Yeni kullanıcı kaydı yapar.
/api/auth/login	POST	Kullanıcı giriş yapar ve JWT token alır.
/api/tasks	GET	Kullanıcının yetkisine göre görevleri listeler.
/api/tasks	POST	Yeni görev oluşturur.
/api/tasks/:id	PUT	Mevcut görevi günceller.

/api/tasks/:id	DELETE	Görevi siler (admin veya oluşturan).
/api/tasks/categories	GET	Tüm görev kategorilerini listeler.
/api/tasks/comments/:taskId	POST	Belirli göreve yorum ekler.
/api/tasks/comments/:taskId	GET	Belirli görevin yorumlarını getirir.
/api/tasks/comments/:commentId	PUT	Yorumu günceller (yetkili kullanıcı).
/api/tasks/comments/:commentId	DELETE	Yorumu siler (admin veya sahibi).
/api/users	GET	Tüm kullanıcıları listeler (sadece admin).

Sistem üç ana tablo üzerinde çalışmaktadır:

- users: id, username, password, role
- tasks: id, title, description, category, assignedTo, createdBy, priority, status, progress, startDate, dueDate
- comments: id, taskId, userId, text, createdAt

3.4. Derleme ve Deploy:

Mobil istemci tarafı React Native ve Expo kullanılarak geliştirilmiştir. Android platformu için .apk dosyası oluşturulurken EAS CLI (Expo Application Services) aracı kullanılmıştır.

Uygulamanın sunucu tarafı Node.js ortamında geliştirildikten sonra Render.com platformuna sürekli barındırma (continuous deployment) prensibiyle yüklenmiştir. Render, kaynak kod deposuna bağlı olarak otomatik olarak build ve deploy işlemlerini yürütmektedir. Veritabanı olarak kullanılan SQLite dosyası (better-sqlite3 ile erişilen taskmanager.db) sunucuda yerel olarak barındırılmaktadır ve yeniden başlatma sonrası tablolar otomatik olarak oluşturulmaktadır.