

TeamFlow

Görev ve Takım Yönetim Uygulaması

Proje Tanımı: Bu projede, birden fazla takımın görevlerini yönetebileceği, kullanıcıların rollere göre sisteme erişebileceği bir görev yönetim sistemi geliştireceksiniz. Projede hem backend (Spring Boot) hem de frontend (React) teknolojilerini kullanarak uçtan uca bir uygulama geliştirmeniz beklenmektedir.

Proje Hedefleri:

- JWT ile güvenli kullanıcı giriş sistemi kurmak
- Rol bazlı yetkilendirme mekanizması oluşturmak (Admin, Manager, Member)
- Takım bazlı görev yönetimi geliştirmek
- Modern bir kullanıcı arayüzü ile görevlerin kontrolünü sağlamak
- Github üzerinden proje versiyonlaması yapmak

Teknoloji Gereksinimleri:

Backend

- Spring Boot (REST API)
- Spring Security (JWT ile)
- Role-Based Authorization
- JPA (Hibernate)
- DTO kullanımı
- Validation
- Global Exception Handling
- PostgreSQL

Frontend

- React
- TypeScript
- Axios
- React Router (sayfa yönlendirme için)
- Form validasyonları

Yapılacaklar Listesi

- **Admin:** Tüm takımları ve kullanıcıları görebilir, yönetebilir.
- **Manager:** Kendi takımındaki kullanıcıları ve görevleri yönetebilir.
- **Member:** Sadece kendisine atanan görevleri görebilir ve durumlarını değiştirebilir.

Özellikler:

- Kullanıcı kayıt ve giriş sistemi (JWT token üretimi)
- Kullanıcı rolleri (Admin, Manager, Member)
- Takım oluşturma ve takım üyeleri ekleme
- Görev oluşturma, listeleme, güncelleme ve silme
- Görev detayları: başlık, açıklama, durum, öncelik, teslim tarihi
- Her kullanıcının sadece kendi takımındaki görevleri görebilmesi
- Görevleri duruma göre filtreleme
- Hataların anlamlı şekilde gösterilmesi (global exception handling)

Roller ve Yetkileri

1- Admin (Sistem Yöneticisi)

Sistemdeki en üst düzey yetkilidir. Tüm takımları, kullanıcıları ve görevleri görebilir ve yönetebilir.

Yetkileri:

- Tüm takımları görüntüleyebilir.
- Tüm kullanıcıları listeleyebilir ve silme yetkisine sahiptir.
- Kullanıcılara rol atayabilir (örneğin bir kullanıcıyı Manager yapabilir).
- Takım oluşturabilir, silebilir.
- Sistemdeki tüm görevleri görüntüleyebilir ve silebilir.
- Sistem üzerinde tam yetkilidir (gizli/log kayıtlarını görüntüleme gibi opsiyonel gelişmiş yetkiler verilebilir).

Kullanım Senaryoları:

- Yeni kullanıcı oluşturulmuşsa ona uygun rol ataması yapar.
- Gerekğinde bir takımı veya kullanıcıyı sistemden kaldırır.
- Sistemsel hata veya tutarsızlık olduğunda müdahale eder.

2- Manager (Takım Yöneticisi)

Kendi takımının yönetiminden sorumludur. Takıma kullanıcı ekleyebilir, görev atayabilir ve takım içindeki görevleri düzenleyebilir.

Yetkileri:

- Kendi takımındaki üyeleri listeleyebilir.
- Yeni takım üyesi oluşturabilir
- Takım üyelerine görev atayabilir.
- Takım görevlerini listeleyebilir, güncelleyebilir ve silebilir.
- Takım içi görev durumlarını takip edebilir.
- Kendi takımındaki kullanıcıların rollerini **değiştiremez**, sadece admin bunu yapabilir.

Kısıtlar:

- Başka takımların verilerine erişemez.
- Admin yetkilerine sahip değildir.

Kullanım Senaryoları:

- Takımına yeni bir çalışan eklendiğinde sistemden oluşturur.
- Görevleri üyelerine dağıtır ve durumlarını izler.
- Gerekliğinde görev güncellemesi veya yeniden atama yapar.

3- Member (Takım Üyesi)

Kendi görevlerini yönetebilir, ancak takımda yönetici rolü yoktur. Sadece atanmış görevlerle ilgilenir.

Yetkileri:

- Sadece **kendi görevlerini** görebilir.
- Görev durumu güncelleyebilir (örn: “Devam Ediyor”, “Tamamlandı”).
- Kendisine atanmış görevlerin detaylarını görüntüleyebilir.

Kısıtlar:

- Başka kullanıcıların görevlerine erişemez.
- Yeni görev oluşturamaz veya başkasına görev atayamaz.
- Kullanıcı veya takım yönetimi yapamaz.

Kullanım Senaryoları:

- Üzerine atanan görevleri günlük olarak kontrol eder.
- Görev tamamlandığında durumu günceller.
- Görevle ilgili açıklamaları takip eder.

Uygulamada Roller Nasıl Test Edilmeli?

Her kullanıcı rolü için örnek login bilgileri oluşturun ve frontend tarafında:

- Admin giriş yaptığında admin paneline yönlendirilmeli.
- Manager giriş yaptığında kendi takımına özel sayfaya yönlendirilmeli.
- Member giriş yaptığında sadece kendi görevlerini göreceği sade bir dashboard gösterilmeli.

Teslim Kuralları

GitHub Kullanımı:

- Projenizi **public GitHub repository** olarak paylaşılmalıdır.
- Git geçmişiniz anlamlı commit mesajları içermelidir.
- README.md dosyası içinde:
 - Proje açıklaması
 - Kurulum adımları
 - Giriş bilgileri (test için kullanıcı adı / şifre)
 - Kullanılan teknolojiler mutlaka yer almalıdır.
 - Uygulama ekran görüntüleri

Teslim Süresi:

- Proje teslim tarihi: **15/08/2025**
- Geç teslimler puan kaybına neden olur.