**VTYS PROJE RAPORU**

**22. Grup**

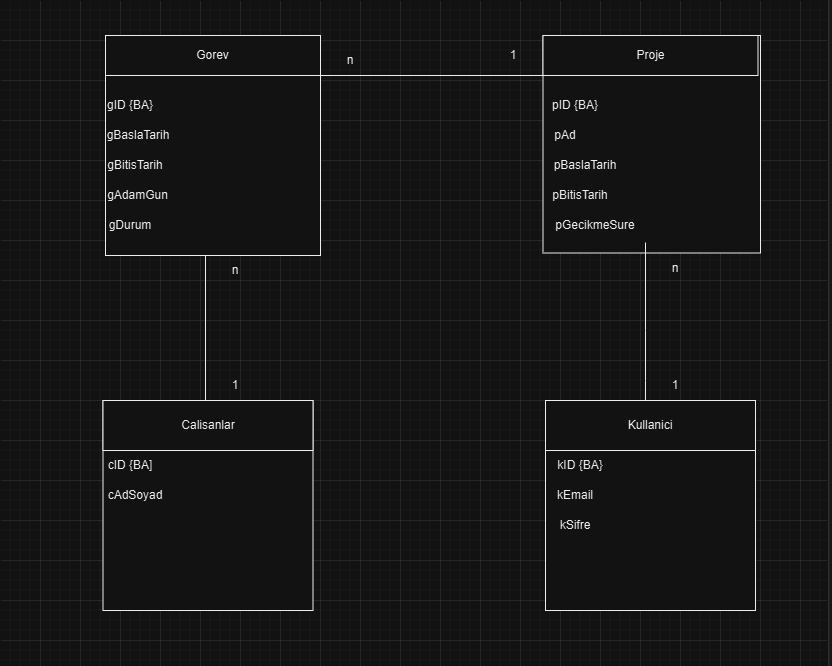
**Github linkine** [**buradan**](https://github.com/omerfderin/Veritabani-Proje-Odevi) **ulaşabilirsiniz.**

**Youtube linkine** [**buradan**](https://www.youtube.com/watch?v=bmfmbanhMKc) **ulaşabilirsiniz**

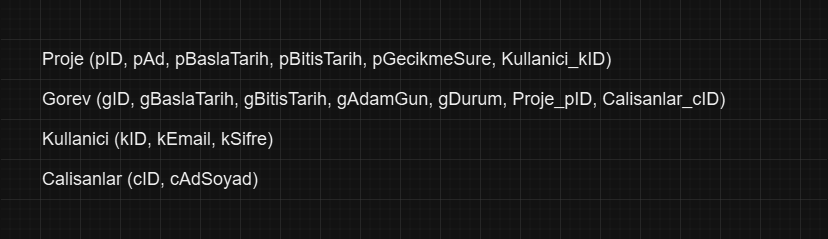
**1. Veri Tabanı Tasarımı**

Kullanılan veri tabanı yönetim sistemi : **MYSQL**

**1.1 Uml Diyagramı**

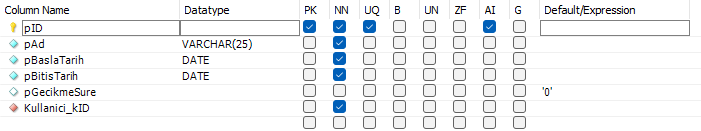


**1.2 ER Diyagramı**

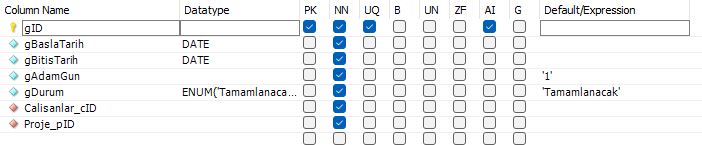


**1.3 Veri Tabanı Tabloları**

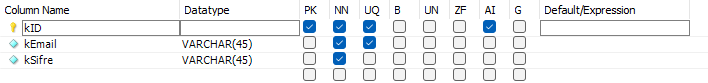
**1.3.1 Proje Tablosu**

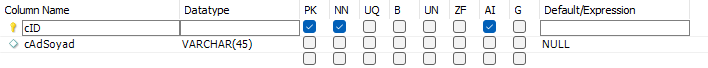


**1.3.2 Gorev Tablosu**

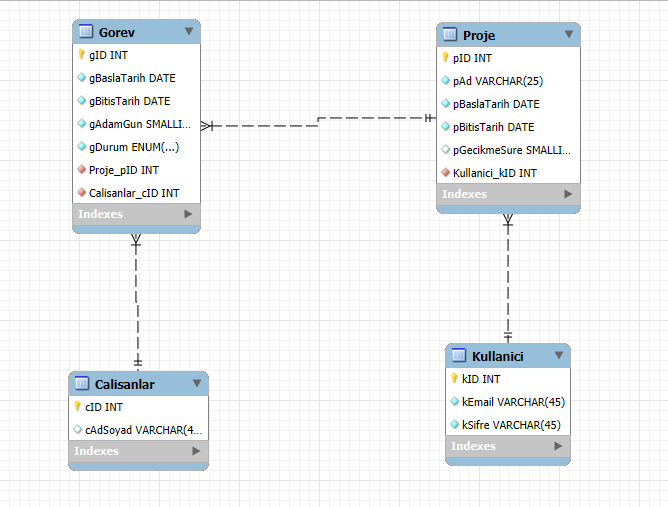


**1.3.3 Kullanici Tablosu**

**1.3.4 Calisanlar Tablosu**



**1.4 EER Diyagram**



**Tablolar Arası İlişkiler**

**Kullanıcı - Proje : 1:N**

Bir proje sadece bir kullanıcı tarafından oluşturulabilir fakat bir kullanıcı birden fazla projede bulunabilir.

**Proje - Görev : 1:N**

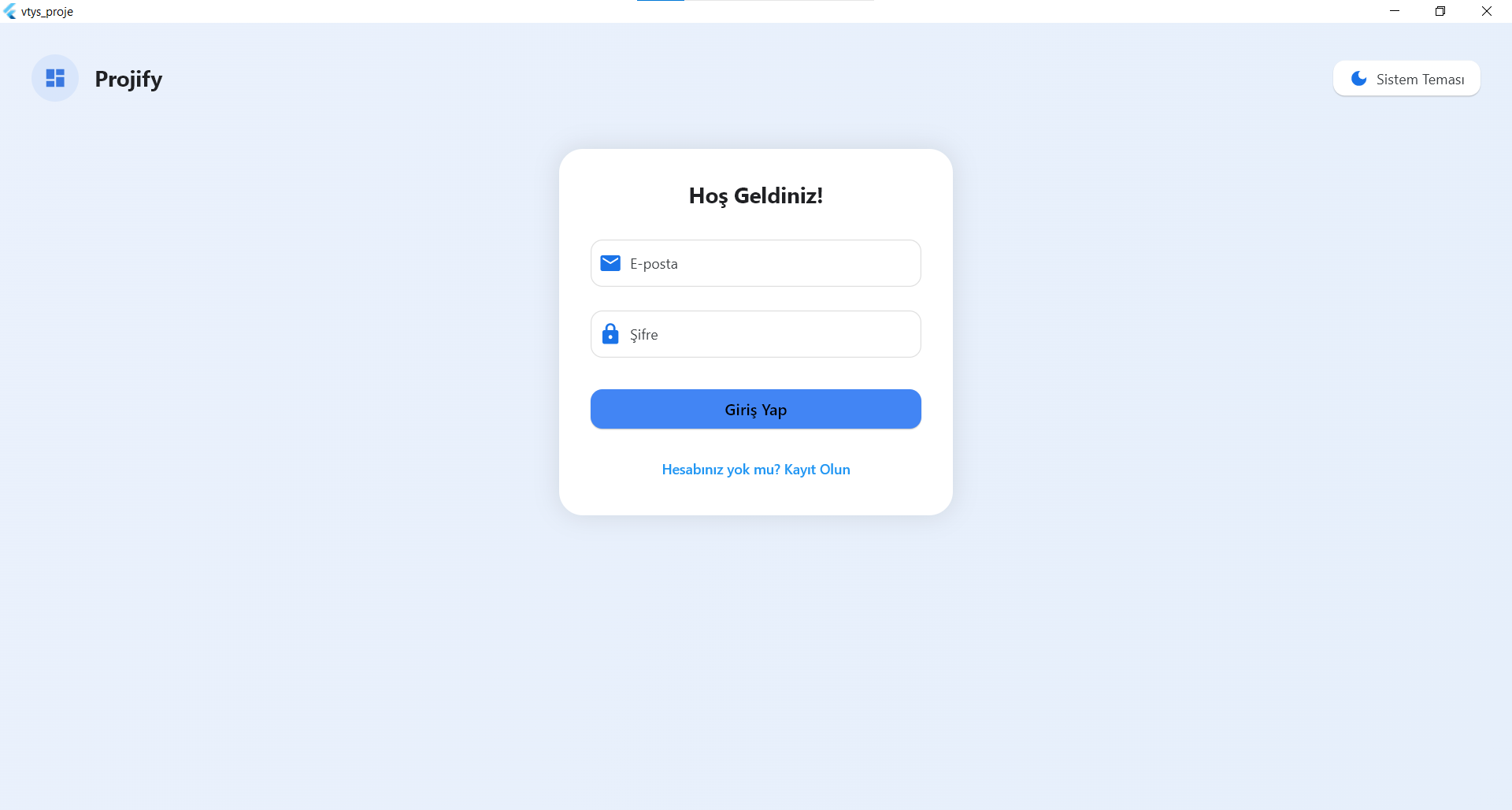
Bir projede birden fazla görev bulunabilir fakat bir görev sadece bir projede bulunabilir.

**Çalışanlar - Görev : 1:N**

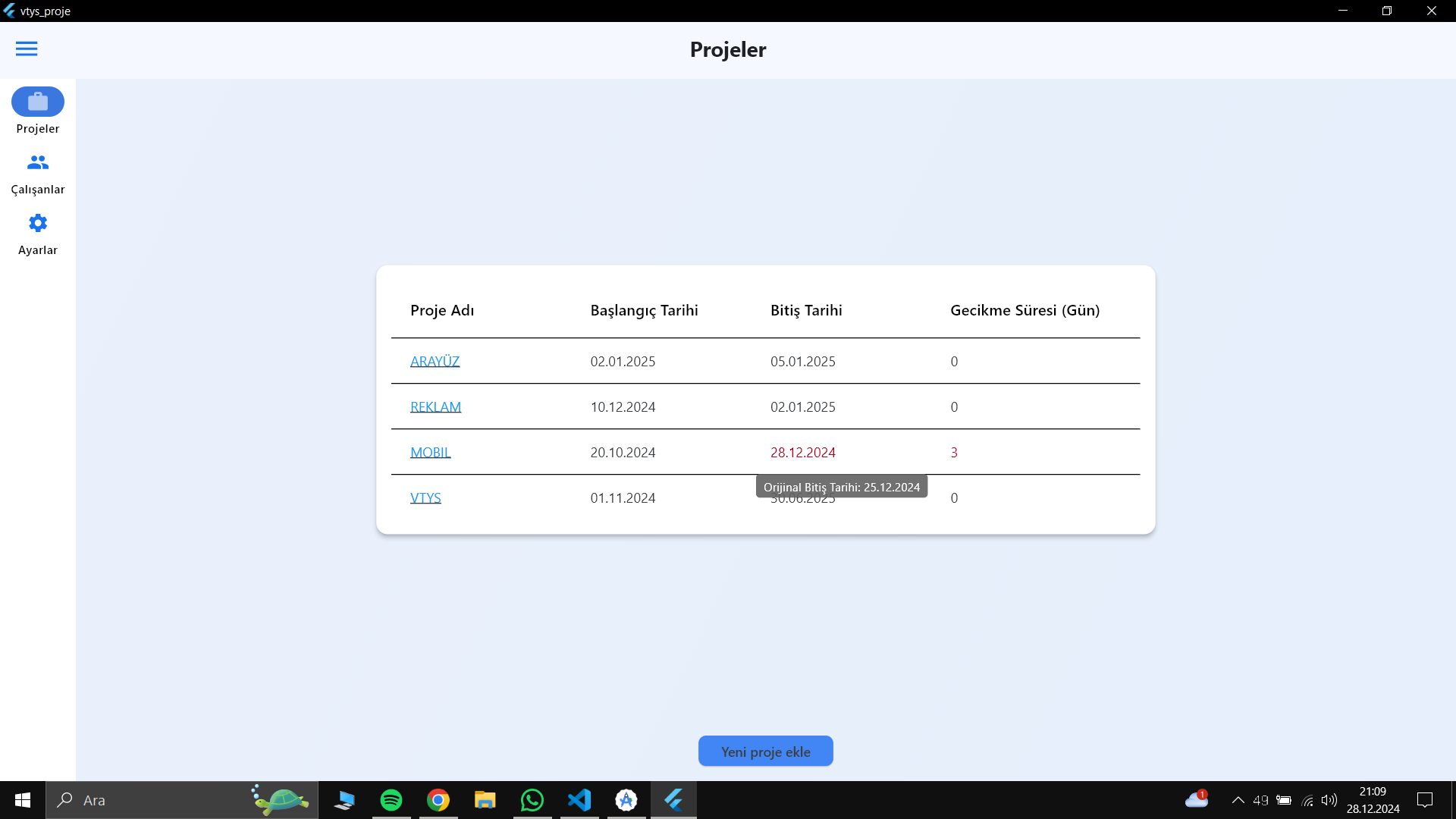
Bir görevde sadece bir çalışan bulunabilir fakat bir çalışan birden fazla görevde bulunabilir.

**2**. **ARAYÜZ TASARIMI**   
Kullanılan Diller : **FLUTTER (DART)**

**2.1 GİRİŞ EKRANI**



Bu arayüzde “Kayıt Olun” ile kullanıcı veritabanına kaydedilir.  
Giriş yapmak isteyen kullanıcıların bilgileri API üzerinden sorgulanarak kontrol edilir.  
Yanlış bilgiler girildiğinde uyarı mesajı gösterilir.

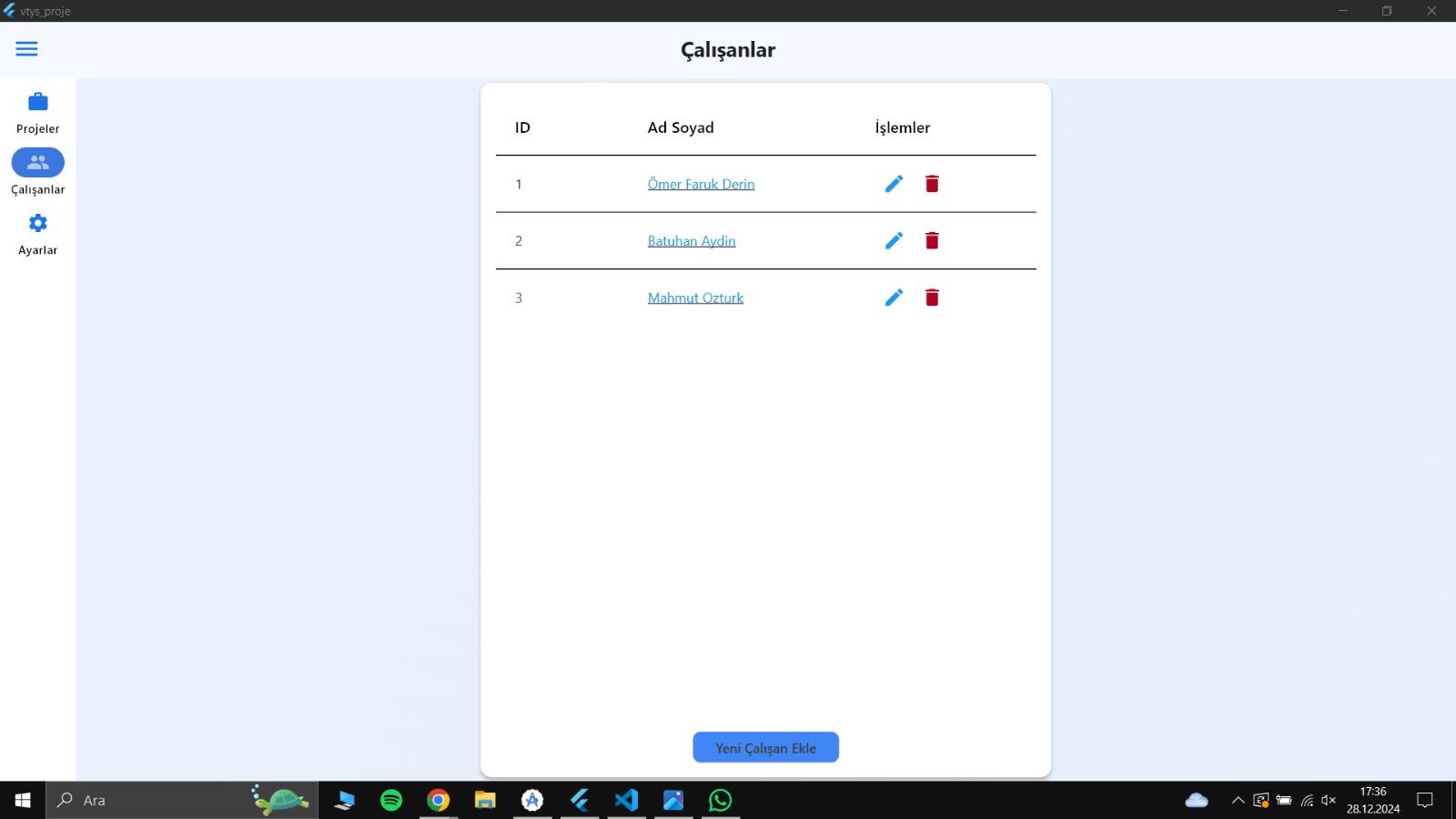
**2.2 PROJELER EKRANI**  
  
  
   
Bu arayüz giriş yapan kullanıcının projeleri görüntüleyip yönetebileceği sayfadır.

Projeler, API'den alınarak Datatable aracılığıyla dinamik olarak oluşturulur.  
Kullanıcılar, yeni bir proje eklemek için tasarlanmış özel bir dialog ekranı aracılığıyla proje bilgilerini girip kaydedebilir.  
Proje ekleme işlemi sırasında boş alan kontrolü yapılır, uygun hata mesajı ile yanıt verilir.

Projenin bitiş tarihi geçmişse gecikme süresi kırmızı renkle vurgulanır.

Proje adı üzerine tıklandığında, kullanıcı ilgili projenin görevlerini görüntüleyebileceği **Proje Görev Ekranı** sayfasına yönlendirilir.  
Bu sayfa, proje detaylarının yanı sıra çalışanlara ve ayarlara hızlı geçiş imkanı sunar.

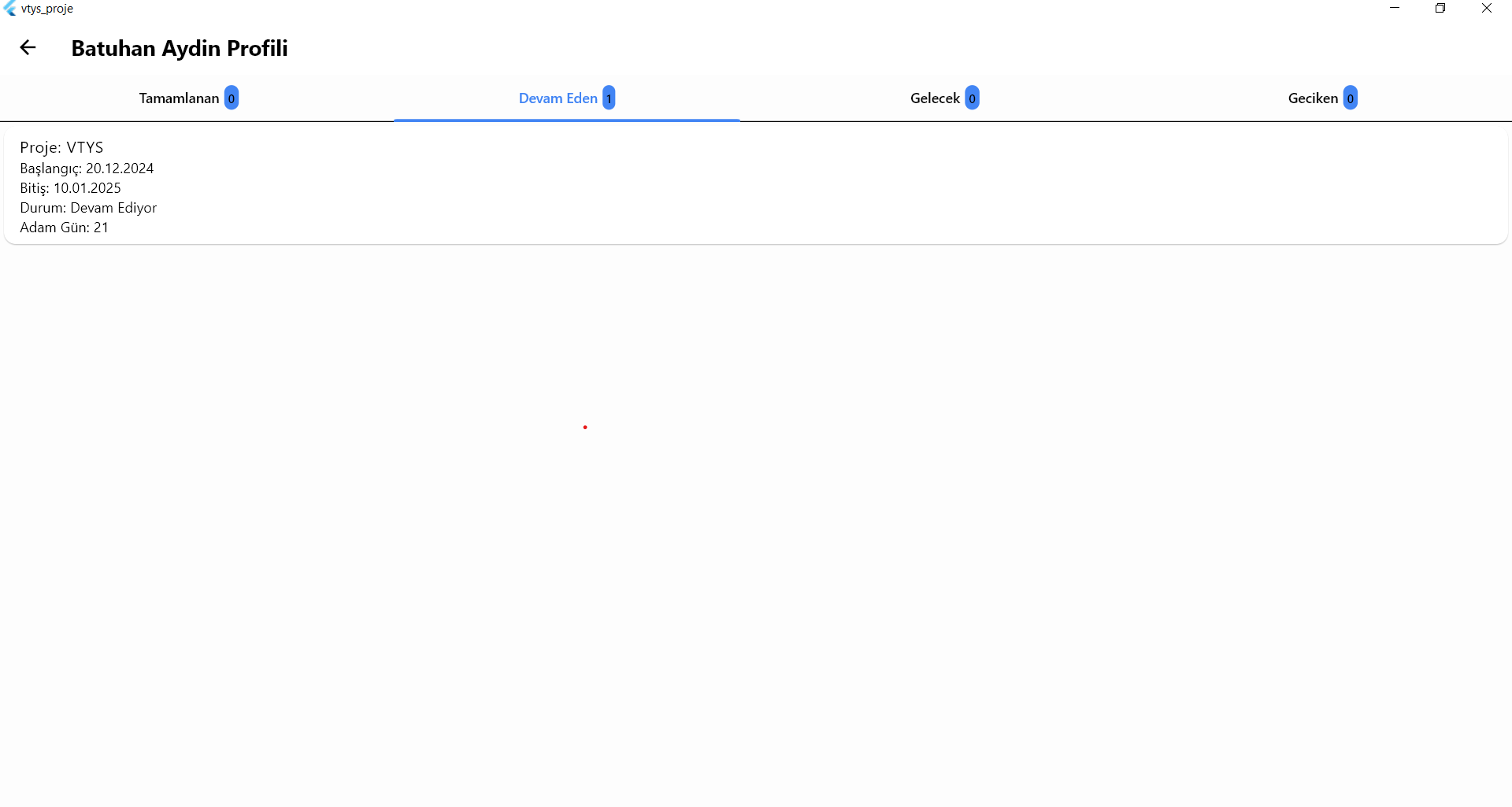
**2.3 ÇALIŞANLAR EKRANI**



Kullanıcıların çalışan bilgilerini görüntüleyip yönetebileceği bir sayfadır. Bu sayfa, çalışanların detaylarına erişim, düzenleme, silme ve yeni çalışan ekleme gibi işlemleri sağlar.  
Güncelleme, ekleme ve silme işlemleri başarılı olduğunda kullanıcıya bilgi mesajı gösterilir

Çalışan ad-soyadına tıklandığında, kullanıcı Çalışan Profil Ekranı ‘na yönlendirilir.

**2.4 ÇALIŞAN PROFİL EKRANI**

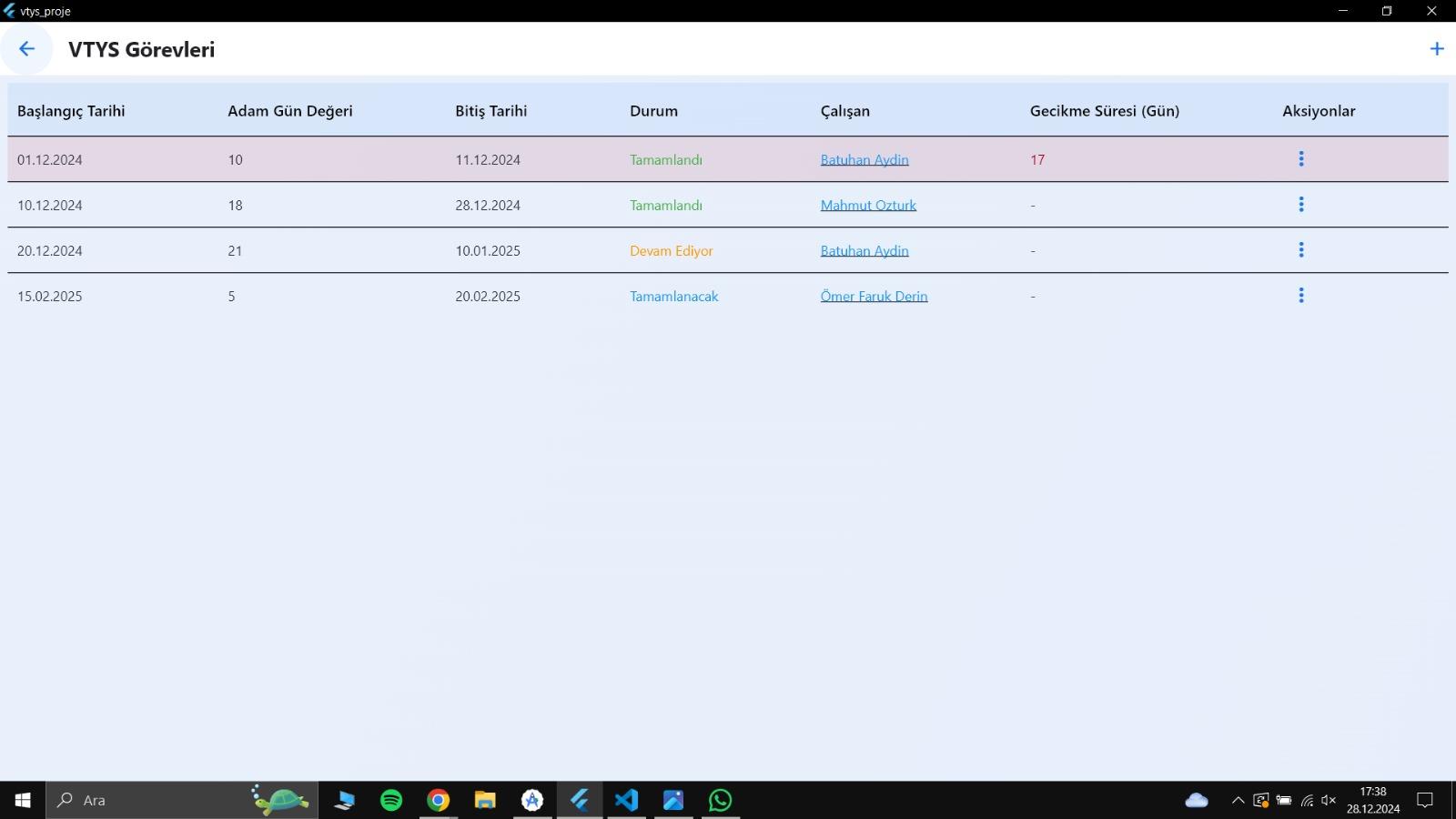


Bu sayfa, belirli bir çalışanın görevlerini görüntülemek için kullanılır. Görevler durumlarına göre farklı kategorilerde gösterilir

Her görev için proje adı, başlangıç ve bitiş tarihleri, durum bilgisi, adam gün ve gecikme süresi gibi detaylar gösterilir.

Kullanıcı, bir sekmeye tıklayarak yalnızca o kategorideki görevleri görüntüleyebilir.

**2.5 PROJE DETAY EKRANI**

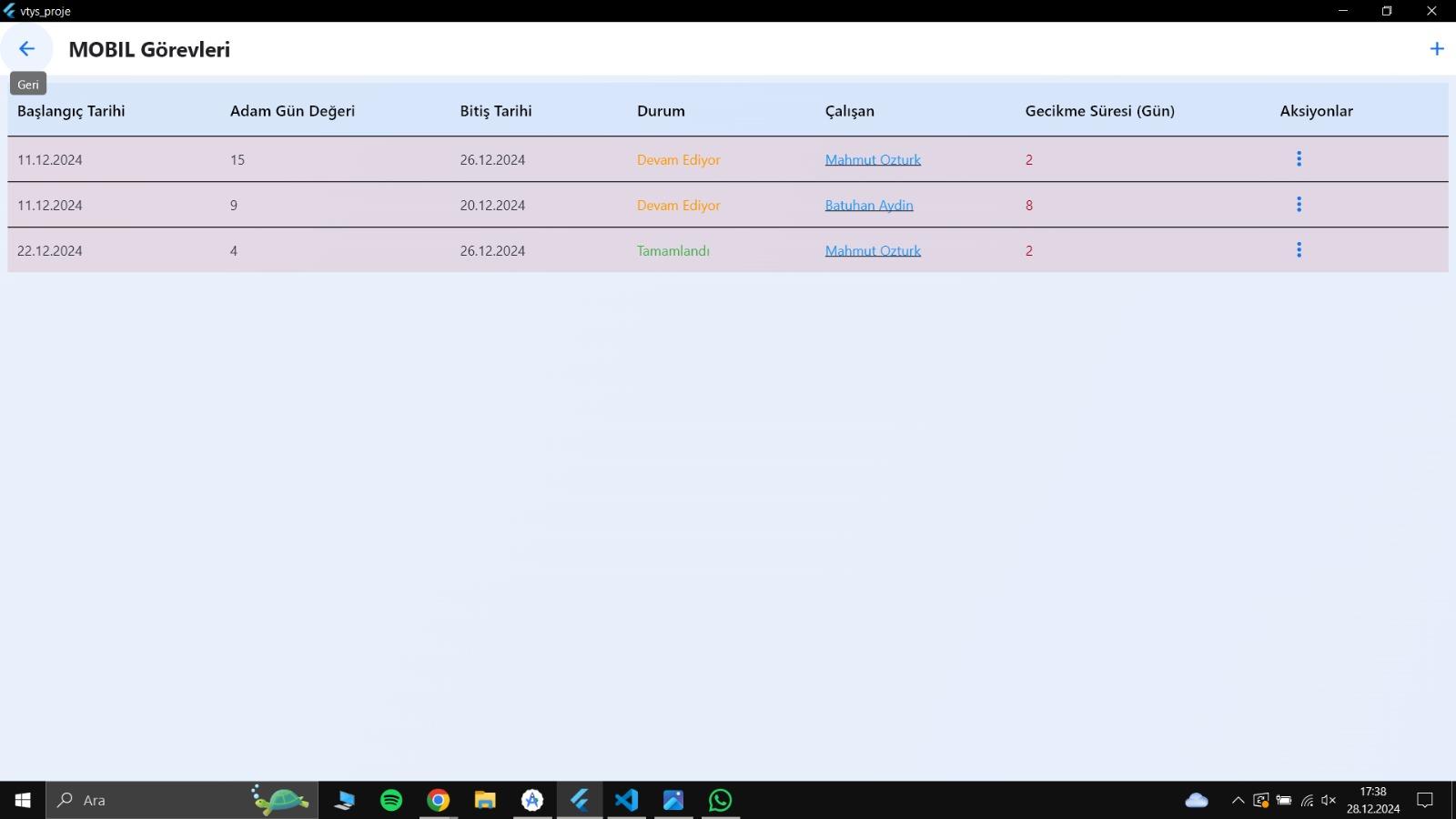


Bu sayfa, seçili bir projeye ait görevleri listelemek, düzenlemek, yeni görev eklemek ve görevlerin durumlarını güncellemek için kullanılır.

Görev durumu "Tamamlanacak", "Devam Ediyor" veya "Tamamlandı" olarak değiştirilebilir.

Çalışanın ismine tıklayarak Çalışan Profil Ekranı’na gidilir.

**2.6 PROJE DETAY EKRANI (GECİKMELİ)**

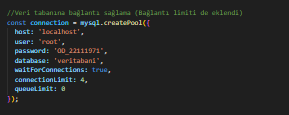


En ileri tarihli görevin bitiş zamanı ile projenin bitiş zamanı karşılaştırılarak projenin gecikip gecikmediği kontrol edilir, en ileri tarihte biten gecikmeli görevin tarihinden projenin bitiş tarihi çıkarılarak projenin gecikme süresi hesaplanır ve anında güncellenir.

**3. BACKEND**

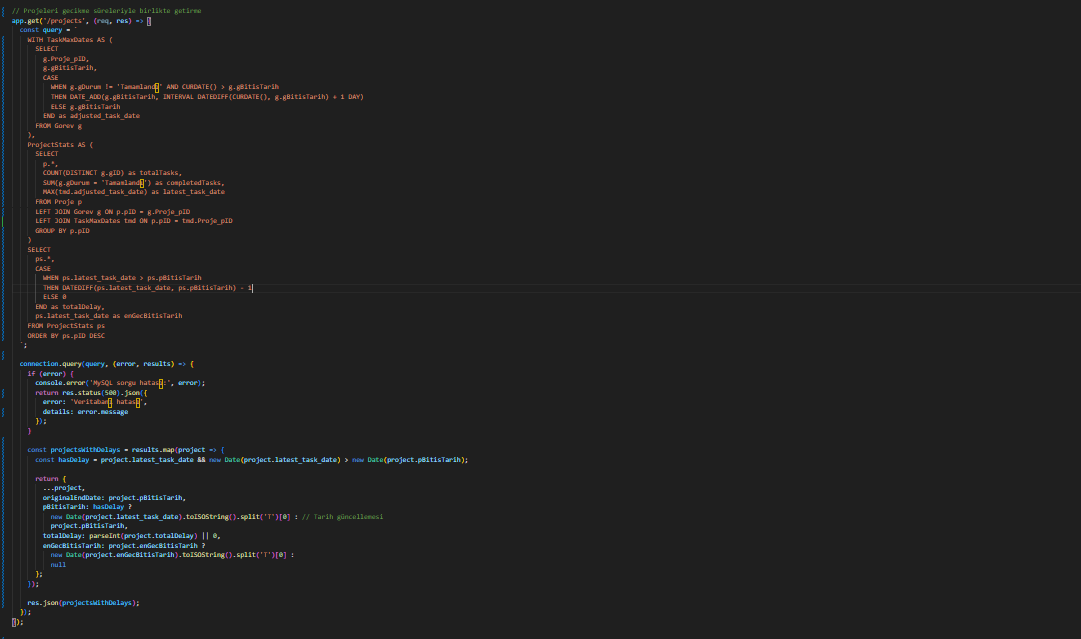
Kullanılan Çalışma Ortamı : **NODE.JS**

**3.1 MYSQL BAĞLANTISI KURMA**

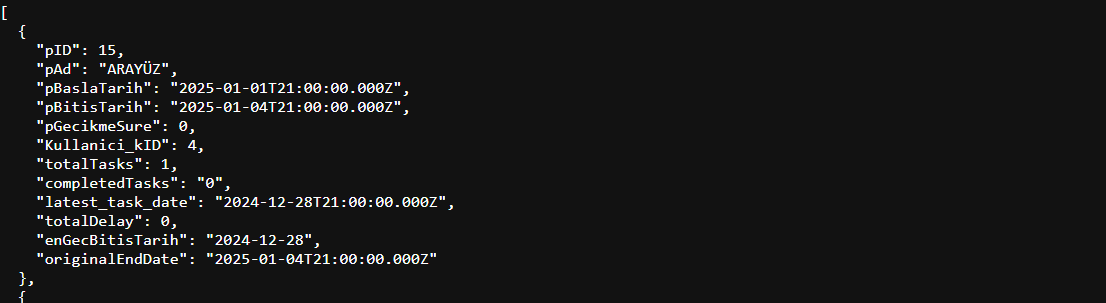
****

Bu kısım localde bulunan mysql veritabanı bağlantısını gerçekleştirmektedir. Şu anda localhost üzerinden çalışmaktadır fakat istenirse mysql in çalıştığı bir sunucu üzerinden de çalıştırılabilmektedir.

**3.2 PROJE GET İŞLEMİ**

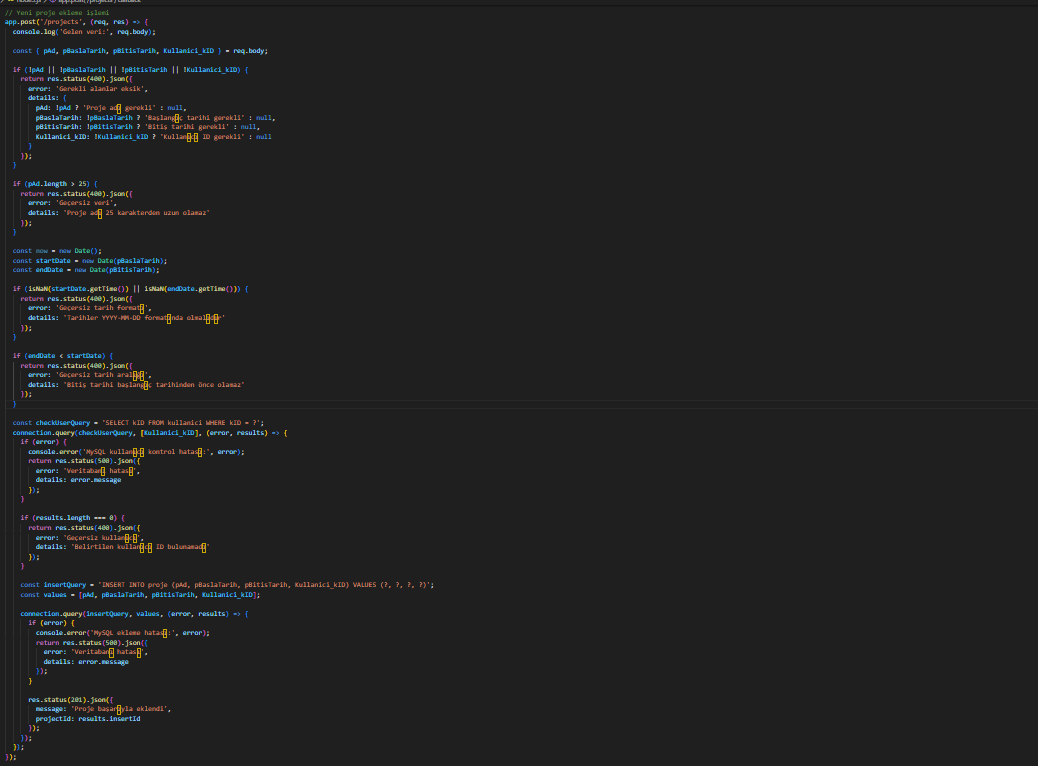
****

Projelerin projeler ekranına getirilmesi işlemini yapmaktadır. Flutter gerekli metodlar yardımıyla bu endpointe istek atar ve nodejs bu istek karşısında tanımlanan sql sorgusu sayesinde projeleri veritabanından çeker ve flutter’a gönderir. Hata durumlarında nodejs konsola hata mesajını yazmaktadır.



Flutter’a gelen json tipindeki bir proje verisi (Projenin gecikme durum kontrolü için fazladan alanlar tanımlanmıştır.)

**3.3 PROJE POST İŞLEMİ**



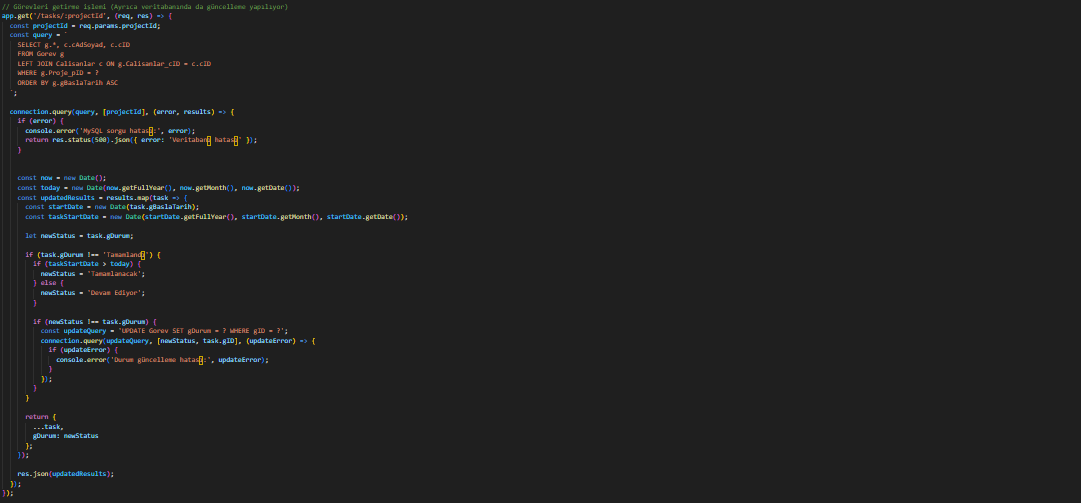
Bu endpoint ise veritabanına proje eklemek için kullanılmaktadır. Projeler ekranında proje ekleme işlemi gerçekleştirildikten sonra flutter üzerinden yollanan bilgiler burada tanımlanan “insert into” sql sorguları yardımıyla veritabanına kaydedilmektedir. Gönderilen verilerin tip kontrolleri de burada yapılmaktadır.

**3.4 PROJE PUT İŞLEMİ**

****

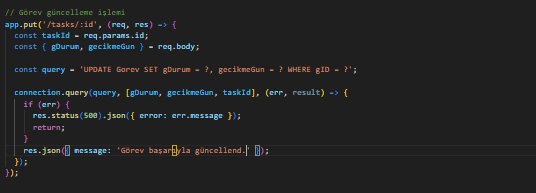
Projenin bitiş tarihinin güncelleme işleminden sorumlu endpointtir. Eğer 3.2 kısımdaki sorgudan gecikme gelirse bu endpoint çalışmaktadır.

**3.5 GOREV GET İŞLEMİ**

****

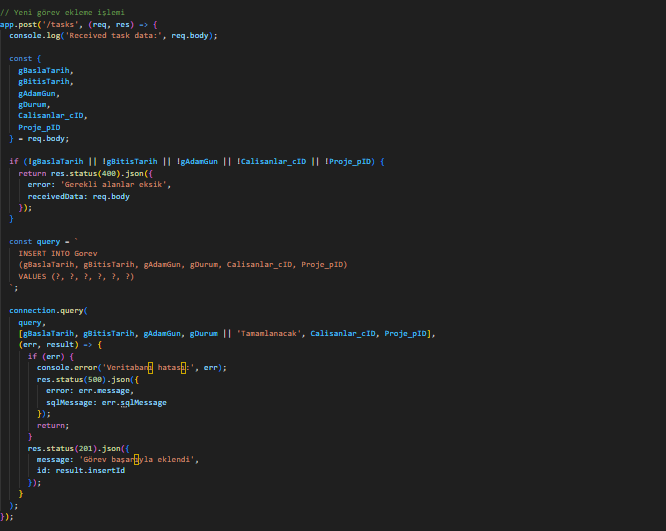
Projede tanımlanan görevleri veritabanından alıp flutter’a iletmekle sorumludur. Ayrıca görevler tablosunda bulunan proje durum değiştirme işlemi yapıldığı zaman bu değişimi sql sorgusu ile veritabanına iletmektedir.

**3.6 GOREV PUT İŞLEMİ**

****

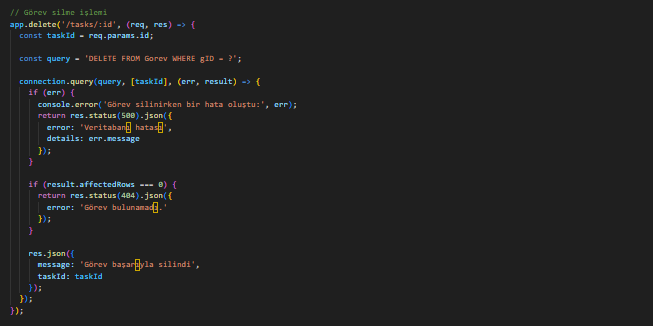
Bu endpoint ise görevin süresinin günler ilerledikçe kontrolü sonrasında durum değişikliği yaşamış olduğu durumlarda veritabanındaki durumunu güncellemek için kullanılmaktadır.

**3.7 GOREV POST İŞLEMİ**

****

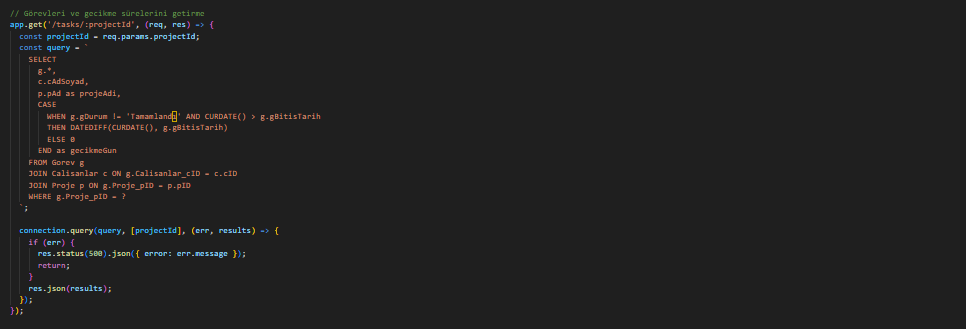
Proje detay ekranında bulunan görev ekleme butonundan alınan bilgileri sql sorguları yardımıyla veritabanına ekleme işlemini yapmaktadır.

**3.8 GOREV DELETE İŞLEMİ**

****

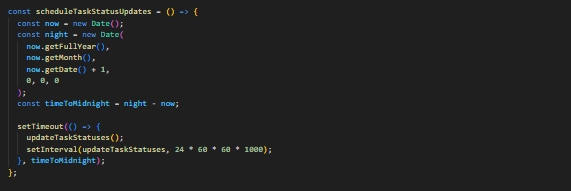
Proje detay ekranında bulunan görev silme işlemini yapmaktadır. Kullanıcı görev sil butonuna bastığı zaman bu görevin veritabanından silinmesi için gereken sql sorgusunu veritabanına iletir.

**3.9 GOREV GET İŞLEMİ**

****

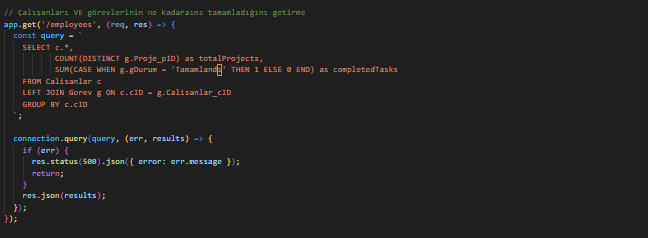
Bu endpoint ise görevlerin gecikme süreleriyle birlikte getirilmesinden sorumludur. Gün ilerlediği zaman gecikme süreleri, veri tabanından bu method yardımıyla alınmaktadır.

**3.10 GÖREVLERİN GECİKME SÜRESİ KONTROLÜ**

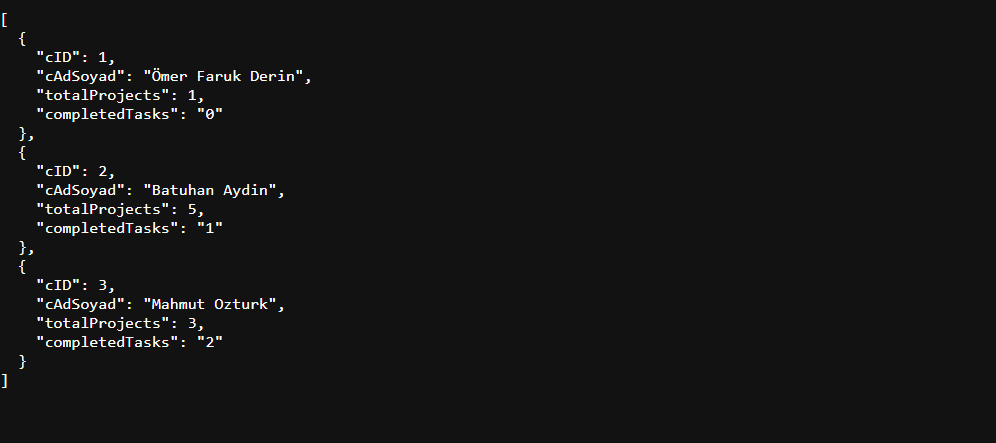
****

Görevlerin gecikme sürelerinin hesaplanmasında yardımcı bir metoddur. Her gün değişiminde görevlerin durumlarını ve gecikme sürelerini kontrol ederek görev durum kontrolüne yardımcı olur.

**3.11 CALİSANLAR GET İŞLEMİ**

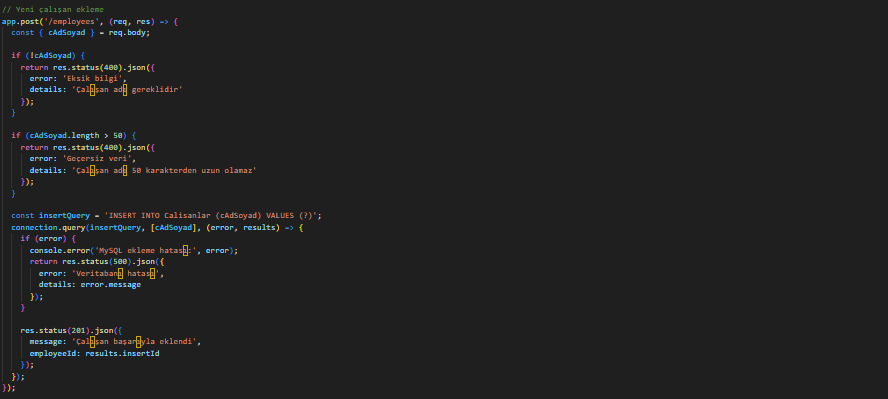
****

Bu endpoint ise çalışanların veritabanından getirilmesi işleminden sorumludur. Çalışanlar ekranında çalışanların id ve isimlerinin gösterimi için kullanılmaktadır. Ayrıca profilde tamamlanan ve tamamlanamamış projelerin gösterilmesinde kullanılmaktadır.



Flutter’a gelen json tipindeki veri

**3.12 CALİSANLAR POST İŞLEMİ**

****

Bu endpoint sisteme yeni çalışan ekleme görevini üstlenmektedir. Çalışanlar sayfasında yeni çalışan ekle işlemi gerçekleştirildiği zaman tanımlanan sorgular yardımıyla yeni çalışanı veritabanına kaydeder.

**3.13 CALİSANLAR DELETE İŞLEMİ**

****

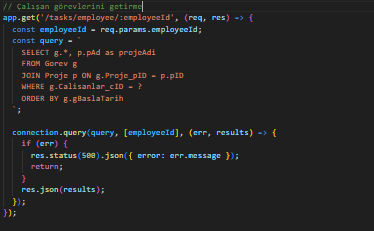
Bu endpoint çalışanlar sayfasında bir çalışan silme işlemi çağrıldığı zaman çalışanın herhangi bir projede görev alıp almadığını kontrol eder eğer almıyorsa bu çalışanı veritabanından kaldırır ancak çalışan bir projede görev alıyorsa ve görevi tamamlanmış değil ise çalışanın sistemden kaldırılmasını engeller aksi takdirde ise çalışanı sistemden siler.

**3.14 CALİSANLAR PUT İŞLEMİ**

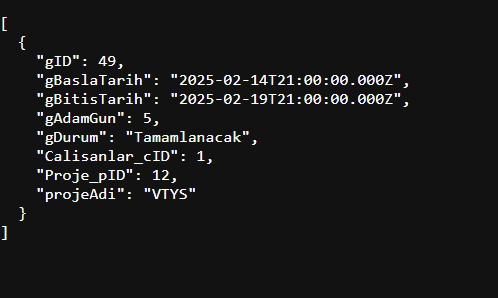
****

Çalışanlar sayfasında bulunan güncelleme işlemi sonrasında güncellenen çalışan ismi veritabanında güncelleme işlemini yapmaktadır

**3.15 CALİSANLAR GET İŞLEMİ**

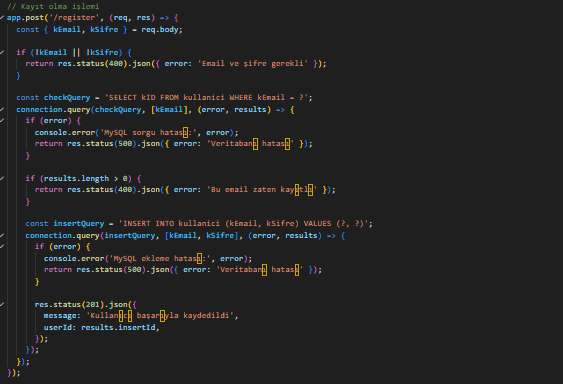


Bu kısım bir çalışanın bir projede yer aldığı bir görevin getirmek için kullanılmaktadır. (Sonrasında profil kısmında da kullanılacaktır.)



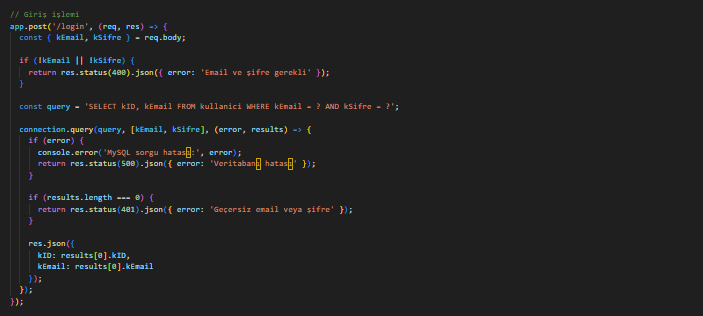
Flutter’a ulaşan json formatındaki veri.

**3.16 KULLANİCİ POST(KAYIT) İŞLEMİ**

****

Kullanıcıların sisteme kaydını üstlenmektedir. Kullanıcıların girdiği e posta ve şifre bilgilerini sisteme yeni kullanıcı olarak kaydetmektedir.

**3.17 KULLANİCİ POST(GİRİŞ) İŞLEMİ**



Kullanıcıların girdiği e posta ve şifre bilgilerini veritabanına göndererek e posta ve şifre kontrolü yapmaktadır. Gerektiği durumlarda kullanıcı hata mesajı almaktadır.

**3.18 SUNUCU VE BAĞLANTI İŞLEMLERİ**

****

**-** app.listen metodu nodejs sunucusunu başlatır ve görevlerin durumlarını kontrol eden metodları çalıştırır.

**-** app.use metodu ise genel bir hata yakalayıcıdır nodejs sunucusunda bir hata olduğu zaman bu hatayı detaylıca konsolda göstermek amacıyla kullanılmaktadır.

**-** process.on metodu ise nodejs sunucusu kapatıldığı zaman nodejs mysql arasındaki bağlantının güvenli bir şekilde kapatılmasını sağlar.