# PROJE RAPORU

# Proje Tanımı

Bu proje, bir hastane yönetim sistemi geliştirmek için **JavaScript** Node.js ve kullanarak, MySQL veritabanı ile etkileşim geliştirmeyi kuran bir uygulama amaçlamaktadır. Sistem, doktor, hasta, randevu ve tıbbi rapor yönetimi işlevlerini içerir. Kullanıcılar (doktorlar ve hastalar) sisteme giriş görüntülevebilir bilgilerini yaparak yönetebilirler.

Kullanılan Teknolojiler

**Frontend** 

HTML: Web sayfası yapısı için kullanıldı

CSS: Web sayfası tasarımı ve stil vermek için kullanıldı.

JavaScript: Kullanıcı etkileşimlerini ve dinamik içerik güncellemelerini yönetmek için kullanıldı.

**Backend** 

Node.js: Sunucu tarafı işlemleri yürütmek için kullanıldı.

Express.js: Web sunucusu oluşturmak ve API oluşturmak için kullanıldı.

MySQL: Veritabanı yönetim sistemi olarak kullanıldı.

Faker.js: Geliştirme ve test amacıyla rastgele veri oluşturmak için kullanıldı.

Veritabanı Tasarımı:Veritabanı tasarımı, sistemin işlevselliğini sağlamak için dört temel tablo içerir.

**Doctors:** Doktor bilgilerini tutar

ID (INT, Primary Key, AUTO\_INCREMENT): Her doktor için benzersiz bir kimlik.

name (VARCHAR(255)): Doktorun adı.

department(VARCHAR(255)): Doktorun kendisinde çalıştığı bölüm.

username(VARCHAR(255),Unique): Doktorun kullanıcı adı

password (VARCHAR(255): Doktorun şifresi.

patients: Hasta bilgilerini tutar.

id (INT, Primary Key, AUTO\_INCREMENT): Her hasta için benzersiz bir kimlik.

FİRSTNAME (VARCHAR(255)): HASTANİN ADİ.

LASTNAME (VARCHAR(255): HASTANÎN SOYADÎ.

AGE(INT): HASTANÎN YAŞÎ.

GENDER (VARCHAR(10)):HASTANIN CINSIYETI.

PASSWORD(VARCHAR(255)):HASTANÎN SÎFRESÎ.

### **APPOINTMENTS:**

id (INT, Primary Key, AUTO\_INCREMENT): Her randevu için benzersiz bir kimlik.

APPOINTMENT DATE (DATE): RANDEVU TARIHI.

APPOINTMENTTIME (TIME): RANDEVU SAATI.

DOCTOR\_ID (INT, FOREIGN KEY): RANDEVU YAPILAN DOKTORUN KIMLIĞI.

PATIENT\_ID (INT, FOREIGN KEY): RANDEVU YAPILAN HASTANIN KIMLIĞI.

MEDİCALREPORTS: TİBBİ RAPOR BİLGİLERİNİ TUTAR.

reportDate (DATE): Rapor tarihi.

reportContent (TEXT): Rapor içeriği.

doctor\_id (INT, Foreign Key): Raporu hazırlayan doktorun kimliği.

patient\_id (INT, Foreign Key): Raporun hazırlandığı hastanın kimliği. İşlevsellikler

Doktor Girişi: Doktorlar kullanıcı adı ve şifre ile sisteme giriş yapabilirler. Giriş yapıldıktan sonra, randevularını ve tıbbi raporlarını yönetebilirler.

Hasta Girişi: Hastalar ad ve şifre ile sisteme giriş yapabilirler. Giriş yaptıktan sonra, randevularını ve tıbbi raporlarını görüntüleyebilirler.

Randevu Ekleme: Doktorlar, hastalar için yeni randevular ekleyebilir.

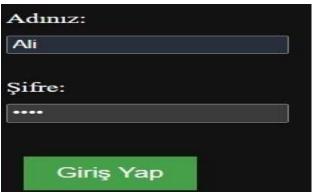
Randevu tarihi: saati, doktor ve hasta bilgileri girilerek oluşturulur. Hasta Kaydı Yeni hastalar, ad, soyad, yaş, cinsiyet ve şifre bilgileri ile sisteme kayıt olabilirler.

Hasta Bilgilerini Görüntüleme : Hastalar, randevularını ve tıbbi raporlarını görüntüleyebilirler. Tıbbi raporlar, doktor ve rapor içeriği bilgilerini içerir. Rastgele Veri Oluşturma Faker.js kütüphanesi kullanılarak, geliştirme ve test aşamalarında kullanmak üzere rastgele doktor ve hasta verileri oluşturulmuştur. Bu veriler, sistemin işlevselliğini test etmek ve doğrulamak için kullanılmıştır.

Kullanıcı Arayüzü : Kullanıcı arayüzü, HTML ve CSS kullanılarak olusturulmustur. Kullanıcılar, asağıdaki formlar aracılığıyla sisteme giriş yapabilir bilgilerini yönetebilirler Bu proje, hastane yönetim sistemi geliştirmek için gerekli temel işlevleri ve yapıları sağlamaktadır. Kullanıcıların (doktorlar hastalar) sisteme giriş yaparak randevu ve tıbbi raporlarını yönetebilmeleri sağlanmıştır. Projenin bir sonraki adımında, daha gelişmiş işlevler eklenebilir ve güvenlik önlemleri artırılabilir.

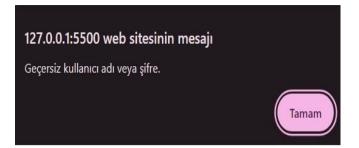
### PROJE İLE İLGİLİ GÖRSELLER:

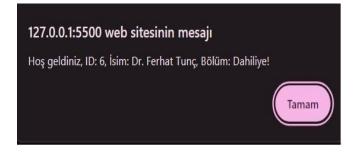


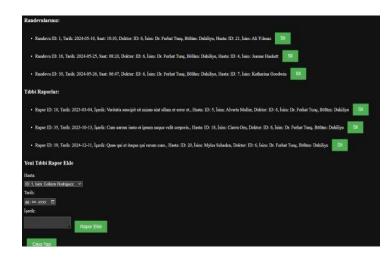












# Doktorlar: Doktor

# Sonuç

Bu proje, hastane yönetim sistemi geliştirmek için gerekli temel işlevleri ve yapıları sağlamaktadır. Kullanıcıların (doktorlar ve hastalar) sisteme giriş yaparak randevu ve tıbbi raporlarını yönetebilmeleri sağlanmıştır. Projenin bir sonraki adımında, daha gelişmiş işlevler eklenebilir ve güvenlik önlemleri artırılabilir.

## KAYNAKÇA

Node.js Resmi Dokümantasyonu MySQL Resmi Dokümantasyonu Faker.js Kütüphanesi MDN Web Docs