

**ANKARA ÜNİVERSİTESİ**  
**MÜHENDİSLİK FAKÜLTESİ**  
**BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**



**BLM 4531 PROJE RAPORU**  
**Hasta Takip ve Reçete Yönetim Sistemi**

**Github linki:**

**<https://github.com/NumalYI/Blm4531-Proje>**

**Youtube Linki:**

**<https://youtu.be/uYKuSVDNpVo>**

**Öğrenci Adı, Soyadı: Ahmed Numan ÇİFTÇİ**

**Öğrenci No: 22290462**

**Öğretmen Adı, Soyadı: Enver Bağcı**

**OCAK, 2026**

## İçindekiler

1. GİRİŞ: .....	3
2. KULLANICI DENEYİMİ VE ROL TABANLI YETKİLENDİRME .....	3
2.1. Doktor Paneli: Kontrol Merkezi .....	3
2.2. Hasta Paneli: Kişisel Sağlık Arşivi.....	3
3. SİSTEMİN KALBI: YAPAY ZEKA DESTEKLİ ÖNERİ MOTORU .....	4
4. TEKNİK MİMARİ: PERDE ARKASINDA NELER OLUYOR?.....	4
4.1. Sunucu ve İş Mantığı (Backend - .NET API).....	4
4.2. Kullanıcı Arayüzü (Frontend - React).....	5
4.3. Veri Saklama Merkezi (Database - SQL).....	5
4.4. Yapay Zeka Entegrasyonu (OneriAi).....	6
5. GÜVENLİK VE VERİ GİZLİLİĞİ .....	6
6. SÜREÇ AKIŞI: BİR MUAYENE GÜNÜ .....	7
7. SONUÇ VE GELECEK VİZYONU .....	7

## 1. GİRİŞ:

Günümüzde sağlık sektörü, bilginin hızı ve doğruluğu üzerine kuruludur. Hasta Takip ve Reçete Yönetim Sistemi geleneksel dosyalama yönteminin getirdiği zaman kaybını ve hata payını ortadan kaldırmak amacıyla geliştirilmiş bütünleşik bir platformdur. Proje bir hastanenin sadece kayıt tutma ihtiyacını değil aynı zamanda veriyi işleme ve doktora karar destek mekanizması sunma ihtiyacını karşılar.

Sistem, en güncel yazılım teknolojilerini (React ve .NET) kullanarak verilerin güvenli bir şekilde saklanması ve yetkili kişilerce her an her yerden erişilmesini sağlar.

## 2. KULLANICI DENEYİMİ VE ROL TABANLI YETKİLENDİRME

Sistem, karmaşayı önlemek adına "ihtiyacın olanı gör" prensibiyle iki farklı kullanıcı arayüzü sunar:

### 2.1. Doktor Paneli: Kontrol Merkezi

Doktorlar için hazırlanan arayüz, bir hastanın tüm tıbbi yaşam döngüsünü yönetebilecekleri bir kontrol paneli niteliğindedir.

- ❖ Hızlı Arama ve Filtreleme: Onlarca hasta arasından TC Kimlik No veya isimle saniyeler içinde arama yapabilirler.
- ❖ Yeni Kayıt Oluşturma: Sisteme yeni bir hasta geldiğinde, demografik bilgileri hızlıca kaydedilir.
- ❖ Muayene Geçmişi: Doktor, hastayı muayene etmeden önce hastanın 1 yıl önce hangi şikayetle geldiğini ve hangi ilaçları kullandığını tek tıkla görebilir.

### 2.2. Hasta Paneli: Kişisel Sağlık Arşivi

Hastalar, kendilerine özel tanımlanan giriş bilgileriyle sisteme dahil olurlar.

- ❖ Şeffaflık: Hasta, doktorun kendisi için koyduğu teşhisleri ve yazdığı reçeteleri dijital olarak görebilir.
- ❖ Süreklilik: Başka bir kuruma gittiğinde veya ilacını unuttuğunda, sistemdeki kayıtları üzerinden kendi tedavi sürecini takip edebilir.

### 3. SİSTEMİN KALBİ: YAPAY ZEKA DESTEKLİ ÖNERİ MOTORU

Bu projeyi standart kayıt sistemlerinden ayıran en büyük fark, arka planda çalışan akıllı öneri sistemidir.

Nasıl Çalışır? Doktor muayene sırasında "Teşhis" alanına bir hastalık ismi girdiğinde sistem veritabanındaki OneriAi tablosuyla anlık iletişim kurar.

Klinik Destek: Sistem o teşhis için önceden tanımlanmış en uygun ilaçları doktora "Öneri" olarak sunar.

Hata Payının Azaltılması: Bu özellik özellikle yoğun çalışma saatlerinde doktorun karar verme sürecini hızlandırır ve rutin tedavilerde unutkanlık payını minimize eder.

### 4. TEKNİK MİMARİ: PERDE ARKASINDA NELER OLUYOR?

Proje "Modüler Katmanlı Mimari" prensibiyle inşa edilmiştir. Bu yaklaşım, sistemin bir parçasında yapılan değişikliğin diğerlerini bozmamasını sağlar. Yani yarın bir gün hastaneye yeni bir birim eklendiğinde tüm sistemi yıkıp yeniden yapmaya gerek kalmaz; sadece ilgili parça güncellenir.

#### **4.1. Sunucu ve İş Mantığı (Backend - .NET API)**

Sistemin "İşlem Merkezi" olan bu katman GitHub'daki Hospital.Api klasöründe yer alır. Burası sadece veri taşımaz aynı zamanda veriyi denetler ve korur.

- ❖ **Güvenlik Protokolü (Middleware & Auth):** Sisteme gelen her talep önce bu "güvenlik kapısından" geçer. Kullanıcı giriş yaptığında ona özel bir JWT

atanır. Bu katman her işlemde bu kimliği kontrol eder. Örneğin; bir hasta kendi reçetesini silebilir mi? Hayır. Sistem burada devreye girer ve "Yetkiniz yok" diyerek işlemi durdurur.

- ❖ **Servis Odaklı Yapı (Business Logic Layer):** PatientService gibi dosyalar, uygulamanın zekasını temsil eder. Bir hastaya yeni bir muayene kaydı girildiğinde bu servis otomatik olarak tarihlerde çakışma olup olmadığını kontrol edip teşhis verisini AI motoruna gönderir ve sonucu düzenleyerek kaydeder.
- ❖ **API (Uygulama Arayüzü):** Backend verileri ham metin (JSON) formatında sunar. Bu sayede gelecekte sisteme bir mobil uygulama (iOS/Android) eklemek isterseniz backend'i hiç değiştirmeden sadece yeni bir Mobil App ekleyerek sistemi büyütebilirsiniz.

## **4.2. Kullanıcı Arayüzü (Frontend - React)**

Kullanıcının gördüğü her şey burada, frontend/src klasöründe şekillenir. Modern ve hızlı bir deneyim sunmak için "Tek Sayfa Uygulaması" (SPA) teknolojisi kullanılmıştır.

**Bileşen (Component) Tabanlı Tasarım:** Uygulama, lego parçaları gibi küçük bileşenlerden oluşur. Örneğin; "Hasta Listesi" bir parçadır "Reçete Formu" başka bir parçadır. Bu kodun tekrar kullanılmasını sağlar; bir formda yapılan iyileştirme, sistemin her yerindeki formları otomatik olarak iyileştirir.

**Global Hafıza (State Management - Context API):** AuthContext.js dosyası sistemin "kısa süreli hafızası" gibidir. Kullanıcı hangi sayfaya giderse gitsin sistem onun kim olduğunu doktor mu yoksa hasta mı olduğunu unutmaz. Bu sayede her sayfada tekrar giriş yapmaya gerek kalmaz.

**Dinamik Veri Akışı (Axios & Interceptors):** Sistem arka planla sürekli konuşur. Bir doktor teşhis yazdığı anda sayfa yenilenmeden yapay zeka önerisi ekranda belirir. Bu kullanıcıya masaüstü programı kadar hızlı ve akıcı bir deneyim sunar.

## **4.3. Veri Saklama Merkezi (Database - SQL)**

Sistemin "Kurumsal Arşivi"dir. Veriler gelişigüzel değil, aralarında mantıksal bağlar kurularak saklanır.

**İlişkisel Veri Modeli:** Veritabanında bir "Hasta" birden fazla "Muayene"ye sahip olabilir. Bir "Muayene" ise birden fazla "Reçete" içerebilir. Bu hiyerarşi sayesinde bir hastanın 5 yıl önceki kaydına ulaşmak binlerce veri arasından bir iğne aramaya değil, düzenli bir kütüphaneden kitap almaya benzer.

**Veri Bütünlüğü (Data Integrity):** Eğer bir hasta kaydı silinirse sistem o hastaya bağlı reçetelerin havada asılı kalmasını engeller. Her şey birbirine bağlı olduğu için veri kirliliği yaşanmaz.

**Entity Framework Core (Aracı Katman):** Kod yazarken karmaşık SQL komutları yerine C# dili kullanılır. Bu teknoloji veritabanını kodun bir parçası gibi görmemizi sağlar.

#### **4.4. Yapay Zeka Entegrasyonu (OneriAi)**

Teknik mimarinin en üstünde yer alan bu katman OneriAi kontrolcüsü üzerinden çalışır.

**Zeka Köprüsü:** Doktorun girdiği metinler Backend içindeki özel bir algoritmaya gönderilir. Bu algoritma veritabanındaki "Teşhis-İlaç" eşleşmelerini tarar.

**Gelecek Vizyonu:** Mevcut yapıda bu sistem sabit bir tablodan beslenmektedir; ancak altyapı gerçek bir makine öğrenmesi modeline bağlanmaya hazır şekilde tasarlanmıştır.

### **5. GÜVENLİK VE VERİ GİZLİLİĞİ**

Sağlık verileri en hassas verilerdir. Proje bu konuda modern standartları uygular:

**JWT :** Sisteme giriş yapıldığında kullanıcıya bir "dijital mühür" verilir. Bu mühür olmadan hiçbir veri talebi karşılanmaz.

**Şifreleme:** Kullanıcı şifreleri veritabanında olduğu gibi değil, karmaşık matematiksel algoritmalarla saklanır. Veritabanı ele geçirilse bile şifreler okunamaz.

**İzlenebilirlik:** Hangi işlemin hangi rol tarafından yapıldığı sistem tarafından her zaman bilinir.

## 6. SÜREÇ AKIŞI: BİR MUAYENE GÜNÜ

Sistemin işleyişini bir örnekle açıklayalım:

1. Giriş: Doktor güvenli bir şekilde sisteme giriş yapar.
2. Hasta Seçimi: Gelen hastayı listeden bulur veya yeni kayıt oluşturur.
3. Muayene Kaydı: Doktor şikayetleri dinler, sisteme notlarını düşer.
4. Akıllı Öneri: Teşhis konulduğu an sistem ilaç önerilerini getirir.
5. Reçeteleme: Doktor önerileri onaylar veya kendi ilaçlarını ekleyerek reçeteyi tamamlar.
6. Erişim: Hasta evine gittiğinde kendi hesabına girerek reçetesini ve doktorun notlarını inceler.

## 7. SONUÇ VE GELECEK VİZYONU

Projem bir hastanenin dijital dönüşümündeki en kritik adımları başarıyla atar. Temiz kod yapısı ve modüler mimarisi sayesinde sistem; laboratuvar sonuçları entegrasyonu, randevu sistemi veya detaylı istatistik raporlama gibi özelliklerle genişletilmeye tamamen müsaittir.

Bu proje, sadece bir kayıt aracı değil; doktor, hasta ve hastane yönetimi arasındaki bağı güçlendiren güvenilir bir dijital köprüdür.