

# HASTANE YÖNETİM SİSTEMİ

**Hasta Takip ve Yönetim Sistemi:** Bu proje, bir hastane yönetim sistemi olarak tasarlanmış bir web uygulamasıdır. Bu uygulama, hastaların kayıt oluşturabileceği, doktorlarla randevu alabileceği, tıbbi raporları saklayabileceği ve genel olarak sağlıkla ilgili işlemleri yönetebileceği bir platform sunması beklenmektedir.

## Veri Tabanı Tasarımı:

Aşağıda veri tabanında bulunması gereken temel tablolar verilmiştir. Proje geliştirme aşamasında yeni tablolar ve bu tablolar arasında ilişkiler kurabilirsiniz. Her bir tablo için *primary key* ve *foreign key* belirlenerek veri tabanı ilişkisel hale getirilmelidir. Tablolarınızı *normalizasyon kurallarına göre* (1NF, 2NF ve 3NF vb.) oluşturmanız beklenmektedir.

- 1. Hastalar:**HastaID,Ad,Soyad,DoğumTarihi,Cinsiyet,TelefonNumarası,Adresvb.
- 2. Doktorlar:**DoktorID,Ad,Soyad,UzmanlıkAlanı,ÇalıştığıHastanevb.
- 3. Yönetici:**YöneticiIDvb.
  - a. Biryönetici,hastaekleyebilir.
  - b. Biryönetici,doktorekleyebilir.
  - c. Biryönetici,tıbbiraporekleyebilir.
- 4. Randevular:**RandevuID,RandevuTarihi,RandevuSaativb.
- 5. TıbbiRaporlar:**RaporID,RaporTarihi,Raporİçeriğivb.
  - a. Tıbbi raporlar görüntüdosyalarıdır.
  - b. *Resim Dosyalarını Saklama:* Laboratuvar sonuçlarını görüntü olarak saklamak için dosyaları bir dosya depolama sistemine yükleyerek, veri tabanında bu dosyalara işaret eden URL'leri saklamalısınız. Bu URL'ler Laboratuvar Sonuçları tablosunda bir sütun olarak eklenmelidir.
  - c. Tıbbi raporlar;hasta,doktor veya yönetici tarafından eklenebilir.

## Nesne Yönelimli Programlama

Bu projenin geliştirilmesi sırasında nesne yönelimli programlama prensiplerini kullanmak zorunludur. Her bir bileşenin (hasta, doktor, randevu, tıbbi rapor vb.) bir sınıf olarak modellenmesi ve her bir sınıf için uygun metotların tanımlanması beklenmektedir. Bir tabloya yapılan ekleme veya çıkarma işlemleri için trigger fonksiyonları yazılarak ilgili diğer tüm tabloların güncellenmesi beklenmektedir.

### Güvenlik Önlemleri:

Laboratuvar sonuçları gibi hassas verilerin güvenliğini sağlamak için HTTPS protokollerini veya uygun şifreleme yöntemlerini kullanılmalıdır.

### Arayüz Geliştirmeleri:

Veritabanında yapılan tüm değişiklikler arayüz üzerinden takip edilebilmelidir.

**1. Kullanıcı Arayüzünü Yenileme:** Hastaların ve doktorların randevularını, laboratuvar sonuçlarını görebileceği, yükleyebileceği bir arayüz eklenmelidir. Bu arayüzde sonuçların tarihleri ve türleri gibi içerik bilgileri de gösterecek şekilde düzenleyin.

**2. Dinamik Bileşenler Kullanma:** Arayüzde AJAX kullanarak sayfa yenilenmeden dosya yükleme işlemini gerçekleştirmelidir. Hasta ve doktor profillerine, kullanıcıların tıbbi geçmişlerini, tedavi notlarını ve randevu geçmişlerini görebilmeleri için dashboardlar eklenmelidir.

**3. Sorgular:** Arayüz üzerinden tüm sorguların sonuçları görüntülenebilmelidir. Örneğin, bir doktor, tüm hastalarını ve bir hastasına ait tıbbi rapor sonuçlarını veya bir hasta tüm randevularını ve tıbbi rapor sonuçlarını görüntüleyebilmelidir. Örneğin;

- a. Bir doktor, tüm hastalarını görüntüleyebilmelidir. Her hastanın adı, soyadı, doğum tarihi, cinsiyeti, telefon numarası, adresi gibi temel bilgiler listelenmelidir.
- b. Bir doktor, belirli bir hastasına ait tıbbi rapor sonuçlarını görüntüleyebilmelidir. Hastanın adı ve soyadıyla birlikte raporların tarihi, içeriği, sonuçları gibi bilgiler sunulmalıdır.
- c. Bir hasta, tüm randevularını görüntüleyebilmelidir. Her randevunun tarihi, saati, doktoru gibi bilgiler listelenmelidir.
- d. Bir hasta, tüm tıbbi rapor sonuçlarını görüntüleyebilmelidir. Raporların tarihi, içeriği, sonuçları gibi bilgiler sunulmalıdır.

### Proje İsterleri:

**1. Hasta Ekleme:** Arayüz aracılığıyla veri tabanına yeni bir hasta girişi işlemi yapılabilir ve ekleme yapılan tüm ilgili tablolardaki güncellemeler veri tabanında proje sunum esnasında gösterilmelidir.

**2. Doktor Ekleme:** Arayüz aracılığıyla veri tabanına yeni bir doktor girişi

işlemi yapılabilir ve ekleme yapılan tüm ilgili tablolardaki güncellemeler veri tabanında proje sunum esnasında gösterilmelidir.

**3. Randevu Alma:** Bir hastanın belirli bir doktorla randevu alarak ve bu randevunun veri tabanına doğru bir şekilde güncellenip güncellenmediği gösterilmelidir.

**4. Tıbbi Rapor Ekleme:** Bir hastanın tıbbi raporunu ekleyerek raporun veri tabanına doğru bir şekilde kaydedilip kaydedilmediğini gösterilmelidir.

**5. Hasta Bilgilerini Güncelleme:** Arayüz üzerinden mevcut bir hastanın bilgilerini güncelleyerek bu değişikliğin veri tabanında doğru bir şekilde güncellenip güncellenmediğini gösterilmelidir.

**6. Doktor Bilgilerini Güncelleme:** Arayüz üzerinden mevcut bir doktorun bilgilerini güncelleyerek bu değişikliğin veri tabanında doğru bir şekilde güncellenip güncellenmediğini gösterilmelidir.

NOT: İsterlerde belirtilen işlemler gerçekleştiğinde, arayüzde bir onay mesajı gösterilmeli ve veri tabanı değişiklikleri anlık olarak raporlanmalıdır.

### Ödev Teslimi

- Önerilen Programlama Dilleri: Python, PHP, JavaScript (Node.js).
- Önerilen VTYS: MSSQL, PostgreSQL, MySQL, SQLite.
- Dersin takibi projenin teslimi dâhil edestek2.kocaeli.edu.tr sistemi üzerinden yapılacaktır. edestek2.kocaeli.edu.tr sitesinde belirtilen tarihten sonra teslim edilen projeler kabul edilmeyecektir. Proje ile ilgili sorular edestek2.kocaeli.edu.tr sitesindeki forum üzerinden
- Algoritma, geliştirdiğiniz kodun çeşitli kısımlarının ne amaçla yazıldığı ve geliştirme ortamı hakkında sorular sorulabilir.
- Veri tabanında oluşturduğunuz tablolar ve normalizasyon işlemleri hakkında sorular sorulabilir.
- Kullandığınız herhangi bir satır kodu açıklamanız istenebilir.