



**KOCAELİ ÜNİVERSİTESİ**  
**BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**  
**BLM210 PROGRAMLAMA LAB. II**  
**PROJE 3**

**Proje İlan Tarihi:** 22/04/2024

**Proje Teslim Tarihi:** 17/05/2024

**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, Doğum Tarihi, Cinsiyet, Telefon Numarası, Adres vb.
2. **Doktorlar:** DoktorID, Ad, Soyad, Uzmanlık Alanı, Çalıştığı Hastane vb.
3. **Yönetici:** YöneticiID vb.
  - a. Bir yönetici, hasta ekleyebilir.
  - b. Bir yönetici, doktor ekleyebilir.
  - c. Bir yönetici, tıbbi rapor ekleyebilir.
4. **Randevular:** RandevuID, Randevu Tarihi, Randevu Saati vb.
5. **Tıbbi Raporlar:** RaporID, Rapor Tarihi, Rapor İçeriği vb.
  - a. Tıbbi raporlar görüntü dosyalarıdır.
  - 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. Ayrıca bu raporlar json formatı ile de saklanmalıdır.
  - d. Tıbbi raporlar; hasta, doktor veya yönetici tarafından eklenebilir/ silinebilir.

**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 ve 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 ve indirebileceği bir arayüz eklenmelidir. Bu arayüzde sonuçların tarihleri ve türleri gibi bilgileri de gösterecek şekilde düzenleyin.
2. **Dinamik Bileşenler Kullanma:** Arayüzde AJAX çağrıları kullanarak sayfa yenilenmeden dosya yükleme ve indirme işlemlerini gerçekleştirmelidir. Hasta ve doktor profillerine, kullanıcıların tıbbi geçmişlerini, tedavi notlarını ve randevu geçmişlerini görebilmeleri için dashboardlar eklenmelidir.
3. **Bildirim Sistemi Kurulumu:** Kullanıcılara yeni laboratuvar sonuçları yüklendiğinde veya sonuçlarında önemli bir güncelleme olduğunda bildirim gönderen bir sistem kurulmalıdır.
4. **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 soyadı ile 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/ Silme:** Arayüz aracılığıyla veri tabanına yeni bir hasta girişi veya hasta silme işlemi yapılabilirdir ve ekleme/ silme yapılan tüm ilgili tablolardaki güncellemeler veri tabanında proje sunum esnasında gösterilmelidir.
2. **Doktor Ekleme/ Silme:** Arayüz aracılığıyla veri tabanına yeni bir doktor girişi veya doktor silme işlemi yapılabilirdir ve ekleme/ silme yapılan tüm ilgili tablolardaki güncellemeler veri tabanında proje sunum esnasında gösterilmelidir. Aktif randevusu bulunan bir doktor silinemeyecektir.
3. **Randevu Alma/ İptal Etme:** Bir hastanın belirli bir doktorla randevu alarak veya randevusunu iptal ederek 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.

**7. Randevu Bilgilerini Güncelleme:** Mevcut bir randevunun bilgilerini güncelleyerek bu değişikliğin veri tabanında doğru bir şekilde güncellenip güncellenmediğini gösterilmelidir.

**8. Tıbbi Rapor Bilgilerini Güncelleme:** Mevcut bir tıbbi raporun bilgilerini güncelleyerek bu değişikliğin veri tabanında doğru bir şekilde güncellenip güncellenmediğini kontrol edin.

**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.
- Proje raporu IEEE formatında 4 sayfa uzunluğunda olmalıdır. Rapor; akış diyagramı veya yalancı kod içermeli, özet, giriş, yöntem, deneysel sonuçlar, sonuç, kaynakça ve ER diyagramından oluşmalıdır. **Bu bir veri tabanı projesi olup ER diyagramı olmayan raporlar değerlendirmeye alınmayacaktır!**
- 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 **Arş. Gör. Kübra Erat** veya **Arş. Gör. Dilara Görmez Açık**'a sorulabilir. Proje teslimine 2 gün kala sorulan hiçbir soruya cevap verilmeyecektir.
- Sunum tarihleri daha sonra duyurulacaktır.
- Sunum sırasında;
  - o Algoritma, geliştirdiğiniz kodun çeşitli kısımlarının ne amaçla yazıldığı ve geliştirme ortamı hakkında sorular sorulabilir.
  - o Veri tabanında oluşturduğunuz tablolar ve normalizasyon işlemleri hakkında sorular sorulabilir.
  - o Kullandığınız herhangi bir satır kodu açıklamanız istenebilir.
- Projenin tanıtım toplantısı detayları e-destek üzerinden duyurulacaktır.
- Proje grupları **en fazla 2 kişiden** oluşmalıdır. Proje grup bilgileri e-destekte paylaşılabilecek link üzerinden en geç **26 Nisan Cuma** gününe kadar girilmelidir. Bu tarihten sonra gruplarda herhangi bir değişiklik yapılmayacaktır.