

PRO LAB 3.PROJE

1. Zeynep Palabıyık
Bilgisayar mühendisliği
Kocaeli Üniversitesi
Kocaeli, Türkiye
220202016@kocaeli.edu.tr

2. Asude Çetin
Bilgisayar mühendisliği
Kocaeli Üniversitesi
Ankara, Türkiye
220202060@kocaeli.edu.tr

Özetçe—Hasta Takip ve Yönetim Sistemi, bir hastane yönetim sistemini web tabanlı bir uygulama olarak sunar. Bu uygulama, hastaların kayıt oluşturabilmesini, doktorlarla randevu alabilmesini ve tıbbi raporları saklayabilmesini sağlar. Veri tabanı tasarımı, hastalar, doktorlar, yöneticiler, randevular ve tıbbi raporlar gibi temel tabloları içerir.

Güvenlik önlemleri arasında HTTPS protokolleri ve uygun şifreleme yöntemlerinin kullanımı bulunur. Arayüz geliştirmeleri, kullanıcıların dosya yükleme ve indirme işlemlerini sayfa yenilenmeden gerçekleştirebilmesini ve bildirim sistemlerinin kurulmasını içerir. Proje istekleri, arayüz üzerinden hasta, doktor, randevu ve tıbbi rapor işlemlerinin gerçekleştirilmesini ve veri tabanı değişikliklerinin anlık olarak raporlanmasını kapsar.

I. GİRİŞ

Hasta Takip ve Yönetim Sistemi, günümüz sağlık hizmetlerinde gereksinim duyulan karmaşık işlevleri hızlı ve etkin bir şekilde yerine getirmek üzere tasarlanmış bir web tabanlı uygulamadır. Bu uygulama, hastaların sağlık durumlarını takip etmelerini, doktorlarla iletişim kurmalarını, randevu alıp yönetmelerini, tıbbi raporları saklamalarını ve genel olarak sağlıkla ilgili süreçleri kolaylıkla yönetmelerini sağlar. Veri tabanı tasarımı, sistemin sağlam bir temelini oluştururken, hastalar, doktorlar, yöneticiler, randevular ve tıbbi raporlar gibi ana bileşenleri içerir. Ayrıca, her bir bileşenin özelliklerini ve ilişkilerini detaylı bir şekilde ele alarak veri tabanını ilişkisel bir yapıya dönüştürmek önemlidir.

Güvenlik açısından, kullanıcı verilerinin gizliliğini ve bütünlüğünü sağlamak için HTTPS protokolleri ve güçlü şifreleme yöntemleri kullanılmalıdır. Arayüz geliştirmeleri, kullanıcıların işlemleri kolayca gerçekleştirebilmelerini ve önemli bildirimlerden haberdar olmalarını sağlayarak kullanıcı deneyimini artırır. Proje istekleri, kullanıcıların arayüz üzerinden

çeşitli işlemleri gerçekleştirmelerini ve bu işlemlerin veri tabanında doğru bir şekilde yansıtılmasını sağlayarak sistemin verimliliğini artırır.

2. YÖNTEM

Bu projede, PHP, MySQL ve JavaScript teknolojilerini kullandık. Veri tabanı tasarımı için, hastalar, doktorlar, yöneticiler, randevular ve tıbbi raporlar gibi temel tabloları belirledik. Her bir tablo için primary key ve foreign key belirleyerek veri tabanını ilişkisel hale getirdik ve normalizasyon kurallarına uygun bir şekilde tabloları oluşturduk.

Nesne yönelimli programlama prensiplerini kullanarak, her bir bileşeni (hasta, doktor, randevu, tıbbi rapor vb.) bir sınıf olarak modellendik ve ilgili metotları tanımladık. Ayrıca, bir tabloya yapılan ekleme veya çıkarma işlemleri için ilgili diğer tüm tabloların güncellenmesini sağladık.

Güvenlik önlemleri olarak, hassas verilerin güvenliğini sağlamak için HTTPS protokollerinin yöntemlerini kullandık. Arayüz geliştirmelerinde, veri tabanında yapılan tüm değişikliklerin arayüz üzerinden takip edilebilmesini sağladık. Kullanıcı arayüzünü yenileyerek hastaların ve doktorların randevularını ve laboratuvar sonuçlarını görebilecekleri, yükleyebilecekleri ve indirebilecekleri bir arayüz ekledik.

AJAX çağrılarını kullanarak dosya yükleme ve indirme işlemlerini gerçekleştirdik ve kullanıcıların tıbbi geçmişlerini, tedavi notlarını ve randevu geçmişlerini görebilmeleri için dashboardlar ekledik. Bildirim sistemi kurarak kullanıcılara yeni laboratuvar sonuçları yüklendiğinde veya sonuçlarında önemli bir güncelleme

olduğunda bildirim gönderen bir sistem kurduk. Son olarak, arayüz üzerinden tüm sorguların sonuçlarının görüntülenebilmesini sağladık, doktorların hastalarını, hastaların randevularını ve tıbbi rapor sonuçlarını görebilmelerini sağladık.

3. DENEY SONUÇLARI

Bu projede, PHP, MySQL ve JavaScript teknolojilerini kullanarak bir hastane yönetim sistemi geliştirdik. Veri tabanı tasarımı için temel tabloları (hastalar, doktorlar, yöneticiler, randevular ve tıbbi raporlar) belirledik ve her bir tablo için primary key ve foreign key belirleyerek veri tabanını ilişkisel hale getirdik. Bu sayede veri tabanı yapısal olarak sağlam ve optimize edilmiş bir yapıya kavuşturuldu.

Nesne yönelimli programlama prensiplerini kullanarak, her bir bileşeni (hasta, doktor, randevu, tıbbi rapor vb.) bir sınıf olarak modellendik ve ilgili metotları tanımladık. Böylece, kodun daha modüler, yeniden kullanılabilir ve sürdürülebilir olmasını sağladık. Ayrıca, bir tabloya yapılan ekleme veya çıkarma işlemleri için ilgili diğer tüm tabloların güncellenmesini otomatik olarak sağladık, böylece veri bütünlüğünü koruduk.

Güvenlik önlemleri olarak, HTTPS protokollerini kullanarak veri iletişimini şifreledik ve hassas verilerin güvenliğini sağladık. Arayüz geliştirmelerinde, kullanıcıların veri tabanında yapılan tüm değişiklikleri arayüz üzerinden takip edebilmelerini sağladık. Yenilenen kullanıcı arayüzü ile hastaların ve doktorların randevularını ve laboratuvar sonuçlarını görüntüleyebilmesini, yükleyebilmesini ve indirebilmesini sağladık.

AJAX çağrılarını kullanarak dosya yükleme ve indirme işlemlerini sayfa yenilenmeden gerçekleştirdik ve kullanıcıların tıbbi geçmişlerini, tedavi notlarını ve randevu geçmişlerini görebilmeleri için dashboardlar ekledik. Ayrıca, kullanıcılara yeni laboratuvar sonuçları yüklendiğinde veya sonuçlarında önemli bir güncelleme olduğunda bildirim gönderebilecekleri bir sistem kurduk.

Son olarak, kullanıcıların arayüz üzerinden tüm sorguların sonuçlarını görüntüleyebilmelerini sağladık, böylece doktorların hastalarını, hastaların randevularını ve tıbbi rapor sonuçlarını kolayca görebilmelerini sağladık. Bu yöntemlerin kullanılmasıyla, projenin güvenliği, kullanılabilirliği ve kullanıcı deneyimi önemli ölçüde artırıldı.

4. SONUÇ

Bu proje, bir hastane yönetim sistemi olarak tasarlanmış bir web uygulaması olup, hastaların, doktorlarla randevu alabilmesi, tıbbi raporları saklayabilmesi ve genel sağlık işlemlerini yönetebilmesini sağlamaktadır. Veri tabanı tasarımında, temel tablolar belirlenmiş, ilişkisel yapı oluşturulmuş ve normalizasyon kurallarına uygun olarak tablolar oluşturulmuştur. Nesne yönelimli programlama prensipleri kullanılarak, her bir bileşenin sınıflar halinde modellenmesi ve ilgili metotların tanımlanması sağlanmıştır. Güvenlik önlemleri olarak, HTTPS protokollerinin kullanımı sağlanmıştır.

Arayüz geliştirmelerinde, tüm değişikliklerin takip edilebilmesi sağlanmış, kullanıcı arayüzü yenilenerek kullanıcıların işlemleri kolaylaştırılmıştır. AJAX çağrılarını kullanarak dosya yükleme ve indirme işlemleri gerçekleştirilmiş, kullanıcıların tıbbi geçmişlerini, tedavi notlarını ve randevu geçmişlerini görebilmeleri için dashboardlar eklenmiştir. Bildirim sistemi kurularak kullanıcılara yeni laboratuvar sonuçları yüklendiğinde veya sonuçlarında önemli bir güncelleme olduğunda bildirim gönderen bir sistem oluşturulmuştur. Son olarak, kullanıcıların tüm sorguların sonuçlarını görüntüleyebilmesi sağlanmış, doktorların hastalarını, hastaların randevularını ve tıbbi rapor sonuçlarını görebilmeleri sağlanmıştır.

Proje isteklerine uygun olarak, arayüz aracılığıyla hasta ekleme/silme, doktor ekleme/silme, randevu alma/iptal etme, tıbbi rapor ekleme ve bilgileri güncelleme gibi işlemler gerçekleştirilmiştir. Bu işlemlerin gerçekleştiğinde arayüzde bir onay mesajı gösterilmiş ve veri tabanı değişiklikleri anlık olarak raporlanmıştır. Bu proje, sağlık sektöründe verimliliği artırmak ve hastalar ile sağlık personeli arasındaki etkileşimi iyileştirmek için önemli bir adım olabilir.

5. KAYNAKÇA

1. <https://www.w3schools.com/php/>
2. <https://www.w3schools.com/MySQL/default.asp>
3. https://www.yusufsezer.com.tr/javascript-ajax/#google_vignette
4. <https://www.youtube.com/watch?v=d0zHIdmYNsU>

	demo Randevular
	RandevuID : int(11)
	RandevuTarihi : date
	RandevuSaati : time
	HastaID : int(11)
	DoktorID : int(11)
	YoneticilID : int(11)
	CalistigiHastane : varchar(255)
	UzmanlikAlani : varchar(255)

	demo TibbiRaporlar
	RaporID : int(11)
	RaporTarihi : date
	RaporIcerigi : text
	HastaID : int(11)
	DoktorID : int(11)
	YoneticilID : int(11)
	RaporURL : varchar(255)
	RaporJSON : longtext

	demo Hastalar
	HastaID : int(11)
	Ad : varchar(50)
	Soyad : varchar(50)
	DogumTarihi : date
	Cinsiyet : enum('Erkek','Kadin')
	Telefon : varchar(255)
	Adres : text
	Sifre : varchar(255)

	demo Yonetici
	YoneticilID : int(11)
	ad : varchar(255)
	Sifre : varchar(255)
	soyad : varchar(255)

	demo Doktorlar
	DoktorID : int(11)
	Ad : varchar(50)
	Soyad : varchar(50)
	UzmanlikAlani : varchar(100)
	CalistigiHastane : varchar(100)
	Sifre : varchar(255)



