PROGRAMLAMA LABORATUVARI-2

PROJE-3

1st Bilge Çeşme
Kocaeli Üniversitesi
Kocaeli/Turkiye
220201045

2nd Selim Eren Kaya
Kocaeli Üniversitesi
Kocaeli/Turkiye
230201127

I.ÖZET

Bu projede hastaların kayıt oluşturabileceği, doktorlara randevu alabileceği, tıbbi raporları saklayabileceği ve genel olarak sağlıkla ilgili işlemleri yönetebileceği bir hastane yönetim sistemi platformu hazırlanması istenmektedir.

II.PROJE TANIMI

Asağıdaki isterlere göre web tabanlı bir hastane sistemi oluşturulmuştur:

II.İSTERLER

Hasta İşlemleri:

•Hasta Ekleme: Yeni hasta girişi ve veri tabanına güncelleme.

•Hasta Silme: Hasta silme ve tüm ilgili tablolarda güncelleme. (Aktif randevusu olan hastalar silinemez.)

Doktor İşlemleri:

•Doktor Ekleme: Yeni doktor girişi ve veri tabanına güncelleme.

•Doktor Silme: Doktor silme ve tüm ilgili tablolarda güncelleme. (Aktif randevusu olan doktorlar silinemez.)

Randevu İşlemleri:

•Randevu Alma: Randevu oluşturma ve veri tabanına güncelleme.

•Randevu İptal Etme: Randevuyu iptal etme ve veri tabanına güncelleme.

Tıbbi Rapor İşlemleri:

•Tıbbi Rapor Ekleme: Rapor ekleme ve veri tabanına kaydetme.

•Tıbbi Rapor Bilgilerini Güncelleme: Mevcut rapor bilgilerini güncelleme.

Hasta/Doktor/Randevu/Rapor Bilgilerini Güncelleme:

•Mevcut hasta, doktor, randevu veya rapor bilgilerini güncelleme ve veri tabanında değişiklikleri anında raporlama.

Genel:

•Tüm işlemler arayüz aracılığıyla yapılmalıdır.

•Her işlemde bir onay mesajı gösterilmelidir.

•Veri tabanı değişiklikleri anlık olarak raporlanmalıdır.

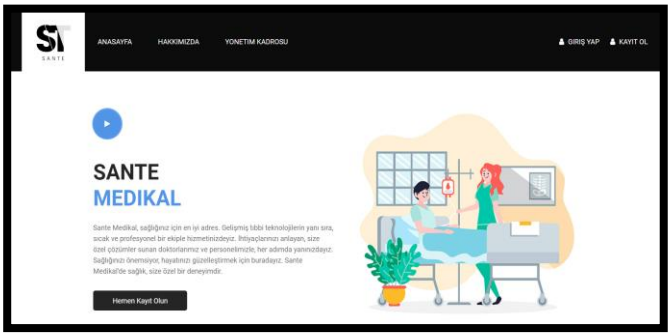

Fig.1: Ana ekran

IV. FONKSİYONLAR

A) app.py Fonksiyonları

Kullanıcı İşlemleri:

- load_user(user_id)

Bu fonksiyon, kullanıcı kimliğine (user_id) göre kullanıcı bilgilerini yükler. Kullanıcı üç farklı türden biri olabilir: Hasta, Yönetici veya Doktor. Fonksiyon sırasıyla her türü kontrol eder ve ilgili sınıfın bir örneğini döndürür. Eğer kullanıcı bulunamazsa None döndürür.

- logout()

Kullanıcı oturumunu sonlandırır (logout_user() fonksiyonu ile) ve kullanıcıyı anasayfaya (home fonksiyonu) yönlendirir.

- home()

Anasayfa fonksiyonudur. Eğer veritabanı tabloları oluşturulmamışsa, bu fonksiyon tabloOlustur ve yoneticileriOlustur fonksiyonlarını çağırarak gerekli tabloları ve yöneticileri oluşturur. Son olarak home.html şablonunu render eder.

- about()

"Hakkımızda" sayfası fonksiyonu. Bu rota, about.html şablonunu render eder.

- yoneticiler()

"Yöneticiler" sayfası fonksiyonu. Bu fonksiyon, yoneticiler.html şablonunu render eder.

- girisYap()

Giriş yapmak için gerekli işlevler ve kontrolleri içerir.

- kayitOl()

Kayıt olmak için gerekli işlevler ve kontrolleri içerir.

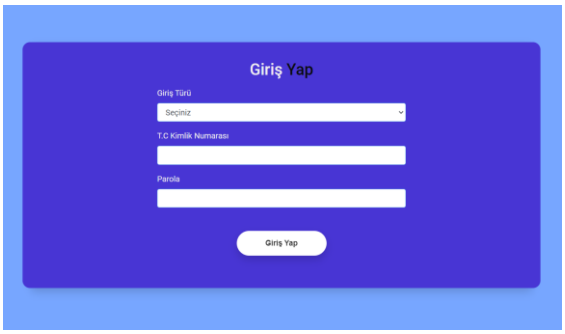

Fig.2: giriş ekranı

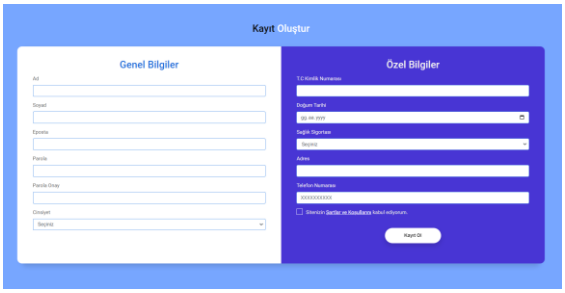

Fig.3: kayıt ekranı

Hasta Menüleri:

- hastaMenu()

Bu fonksiyon, "Hasta" sınıfına ait kullanıcıların hastaMenu.html şablonunu görüntülemesini sağlar.

Kontrol: Kullanıcının sınıfının "Hasta" olup olmadığını kontrol eder. Değilse, girisYap sayfasına yönlendirir.

Dönüş: Eğer kullanıcı "Hasta" ise hastaMenu.html şablonunu title parametresi ile birlikte render eder.

- hastaProfil()

Bu fonksiyon, "Hasta" sınıfına ait kullanıcıların profil bilgilerinin görüntülenmesi ve güncellenmesi işlevini yerine getirir.

GET:Kullanıcıyı hastaMenuler/hastaProfil.html şablonuna yönlendirir.

POST: Formdan gelen durum parametresine göre farklı işlemler yapar:

bilgiDegisme: Boş bir JSON yanıtı döner.

bilgiİptal: Kullanıcının mevcut bilgilerini JSON formatında döner.

bilgiGuncelleme: Kullanıcı bilgilerini günceller. Öncelikle formdaki bilgilerin doğruluğunu kontrol eder. Geçersiz bilgi varsa hata mesajı ile JSON formatında döner. Bilgiler geçerliyse, bilgileri günceller ve başarılı bir JSON yanıtı döner.

- hastaRandevuMenu()

Bu fonksiyon, "Hasta" sınıfına ait kullanıcıların randevu menüsünü kullanarak randevularını yönetmesini sağlar.

GET:Kullanıcıyı hastaMenuler/hastaRandevuMenu.html şablonuna yönlendirir.

POST: Formdan gelen islem parametresine göre farklı işlemler yapar:

İşlem 0: Kullanıcının randevularını sayfalama ile listeler. page ve per_page parametrelerine göre randevuları veritabanından çeker ve JSON formatında döner.

İşlem 1: Şubelerin listesini döner.

İşlem 2: Seçilen şubeye göre departmanların listesini döner.

İşlem 3: Seçilen departmana göre doktorların listesini döner.

İşlem 4: Yeni randevu oluşturur. Randevu bilgilerinin geçerliliğini kontrol eder. Eğer bilgiler geçerliyse randevuyu veritabanına ekler ve başarılı bir JSON yanıtı döner.

İşlem 5: Belirtilen randevuID'ye göre randevuyu iptal eder. Kullanıcının bu randevuya sahip olup olmadığını kontrol eder ve randevuyu siler.

İşlem 6: Mevcut bir randevuyu günceller. Randevu bilgilerini kontrol eder ve geçerliyse günceller.

- hastaRaporMenu()

hastaRaporMenu fonksiyonu, hastaların rapor menüsüne erişimini sağlar.

İlk olarak, giriş yapan kullanıcının sınıfının "Hasta" olup olmadığını kontrol eder. Eğer

kullanıcı sınıfı "Hasta" değilse, kullanıcıyı giriş yapma sayfasına yönlendirir (girisYap). Eğer kullanıcı sınıfı "Hasta" ise, hastaRaporMenu.html şablonunu yükler ve ekranda gösterir.

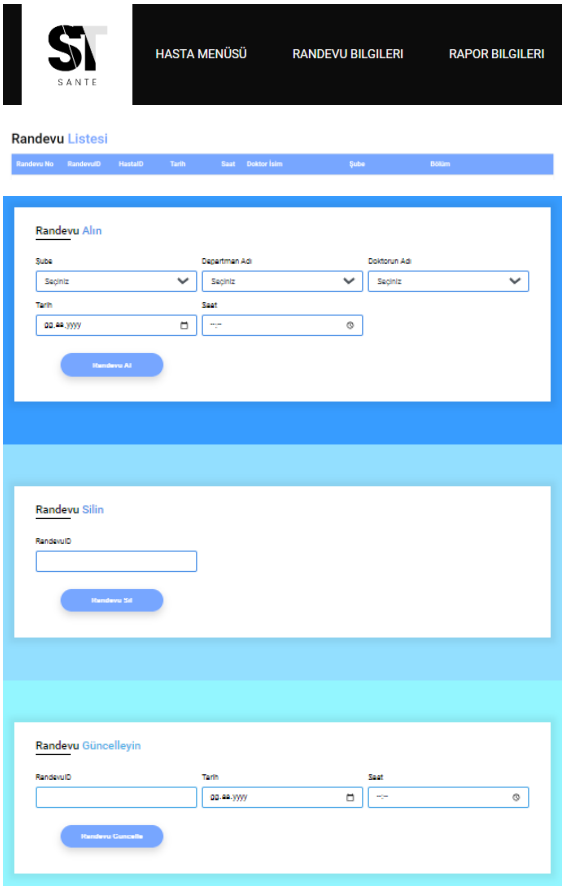

Fig.4: hasta işlemleri

Doktor Menüleri:

- doktorMenu()

doktorMenu fonksiyonu, doktorların ana menüsüne erişimini sağlar.

İlk olarak, giriş yapan kullanıcının sınıfının "Doktor" olup olmadığını kontrol eder. Eğer kullanıcı sınıfı "Doktor" değilse, kullanıcıyı giriş yapma sayfasına yönlendirir (girisYap). Eğer kullanıcı sınıfı "Doktor" ise, doktorMenu.html şablonunu yükler ve ekranda gösterir.

- doktorProfil()

doktorProfil fonksiyonu, doktorların profil bilgilerini görüntülemesi ve güncellemesi için kullanılır.

İlk olarak, giriş yapan kullanıcının sınıfının "Doktor" olup olmadığını kontrol eder.

Eğer kullanıcı sınıfı "Doktor" değilse, kullanıcıyı giriş yapma sayfasına yönlendirir (girisYap).

Eğer HTTP POST isteği yapılmışsa durum parametresi kontrol edilir:

"bilgiDegisme" durumunda, boş bir JSON yanıtı döner (işlem başarılı).

"bilgiİptal" durumunda, mevcut kullanıcı bilgilerini içeren bir JSON yanıtı döner.

"bilgiGuncelleme" durumunda, formdan alınan veriler doğrulanır ve hatasızsa güncellenir.

Eksik veya hatalı bilgi varsa uygun hata mesajı ile 400 hatası döner.

Her şey yolundaysa, bilgileri günceller ve başarılı bir JSON yanıtı döner.

GET isteğinde ise doktorProfil.html şablonunu yükler ve ekranda gösterir.

- doktorRandevuMenu()

Fonksiyonun başında, giriş yapan kullanıcının sınıfının "Doktor" olup olmadığı kontrol edilir. Eğer kullanıcı "Doktor" değilse, giriş yapma sayfasına yönlendirilir.

request.method kontrol edilerek, yalnızca POST istekleri işlenir.

islem parametresi kullanılarak yapılacak işlem belirlenir:

a. islem == 0: Randevu Listesini Getirme:

Sayfa numarası (page) alınır ve her sayfa için kaç randevu gösterileceği (per_page) belirlenir.Veritabanına bağlantı kurulur ve doktorID'ye göre randevular sorgulanır.

Sorgulanan randevular JSON formatında döndürülür.

Her randevu için ilgili hasta bilgileri de eklenir.

b. islem == 1: Randevu Silme:

randevuID alınır ve geçerliliği kontrol edilir.

Eğer geçerli bir randevuID değilse hata mesajı döner.

doktorID ve randevuID'ye göre veritabanından randevu silinir ve başarılı işlem yanıtı döner.

c. islem == 2: Randevu Güncelleme:

randevuID, saat ve tarih alınır ve geçerliliği kontrol edilir.

Eğer geçerli bilgiler değilse hata mesajı döner.

randevuID, doktorID ve hastaID'ye göre randevu güncellenir.

Belirtilen tarih ve saat için hasta veya doktorun başka bir randevusu olup olmadığı kontrol edilir.

Eğer başka bir randevu yoksa, randevu güncellenir ve başarılı işlem yanıtı döner.

GET İstekleri:

GET isteği yapıldığında, doktorRandevuMenu.html şablonu yüklenir ve ekranda gösterilir.

Randevu Listesi: Doktorun tüm randevularını tarih sırasına göre görüntülemesini sağlar.

Randevu Silme: Doktorun belirli bir randevuyu

silmesini sağlar.

Randevu Güncelleme: Doktorun mevcut bir randevunun tarih ve saatini güncellemesini sağlar.

Kullanıcı Doğrulaması: Sadece doktor sınıfına ait kullanıcıların bu işlemleri yapabilmesini sağlar.

Bu fonksiyon, doktorların randevu yönetimini kolaylaştırarak, hasta ve randevu bilgilerini etkin bir şekilde görüntüleme, güncelleme ve silme işlemlerini gerçekleştirmelerine olanak tanır.

Yönetici Menüleri:

- yoneticiMenu()

Bu kod, yönetici kullanıcılarının randevu bilgilerini görüntüleyebileceği bir menü oluşturur.

Kullanıcının sınıfı "Yonetici" değilse, giriş yapma sayfasına yönlendirilir.

POST isteği yapılırsa sayfa numarası (page) alınır ve sayfalama için gerekli offset hesaplanır.

Veritabanından belirli sayıda (per_page) randevu bilgisi çekilir.

Randevular JSON formatında döndürülür.

GET isteği yapılırsa, yoneticiMenu.html şablonu yüklenir ve görüntülenir.

Yardımcı Fonksiyonlar:

- kullanici_bilgi_sorgula()

Bu fonksiyon, giriş türüne (girisTur) ve T.C. Kimlik Numarası'na (tcKimlikNo) göre kullanıcıyı veritabanında arar. Girilen tür "Yönetici", "Doktor" veya "Hasta" olabilir ve buna göre ilgili arama fonksiyonunu (sqlFonksiyonlar.YoneticiAra, sqlFonksiyonlar.DoktorAra, sqlFonksiyonlar.HastaAra) çağırır.

- kullanici_olustur()

Yeni kullanıcı oluşturulmasını sağlar.

- kullanici_menu_yonlendir()

girisTur paramtresini kontrol ederek kullanıcıyı gerekli menüye yönlendirir.

B) SqlFonksiyonlar.py

Bu kısım, hastane yönetim sistemi için gerekli veritabanı tablolarını oluşturur ve bazı yardımcı değişkenleri tanımlar. İlk olarak, turkish_cities, mails, sigortalar, sube_isimleri ve uzmanlik_alanlari listeleri, Türkiye'deki şehirler, e-posta alan adları, sağlık sigortası şirketleri, hastane şubeleri ve tıbbi uzmanlık alanlarını içerir.

Kod, connect_db() fonksiyonu ile SQLite veritabanına (database.db) bağlanarak yabancı anahtar kısıtlamalarını etkinleştirir. tabloOlustur() fonksiyonu ise Flask uygulama bağlamında çalışarak veritabanına bağlanır ve çeşitli tabloları oluşturur: Hastalar tablosu hasta bilgilerini, Hastaneler tablosu hastane şubelerini, UzmanlikAlanlar tablosu tıbbi uzmanlık alanlarını, HastaneBilgiler tablosu hastane şubeleri ve uzmanlık alanları arasındaki ilişkileri, Doktorlar tablosu doktor bilgilerini, Yoneticiler tablosu yönetici bilgilerini ve Randevular tablosu randevu bilgilerini içerir. Tüm tablolar yabancı anahtar kısıtlamaları ile birbirine bağlanmıştır. Fonksiyon, veritabanı bağlantısını kapatarak ve değişiklikleri kaydederek işlemi tamamlar.

Bu kısmın geri kalanında doktor, hasta ve yöneticilerin yaptığı ekleme, silme vb. işleri idare eden fonksiyonlar bulunuyor. Yapılan her bir değişikliğin veritabanı üzerine de kaydedilmesi sağlanıyor.

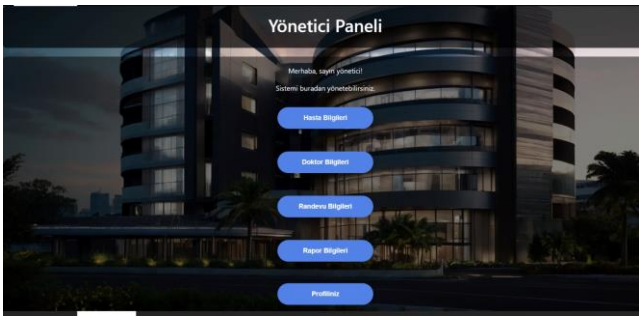

Fig5: yönetici paneli

V. KAYNAKÇA

https://www.youtube.com/watch?v=DTCru23XCZ0
https://academy.patika.dev/courses/sql/PrimaryKeyAndForeignKey
https://www.youtube.com/watch?v=ZVzbmV6sYlM

VI.KATKILAR

Bu ekip çalışmasında Bilge Çeşme tablo oluşturma , web tasarımı ve rapor yazmada görev almıştır. Selim Eren Kaya ise menülerin çalışması, giriş , kayıt, randevu alma gibi işlemler ile ilgilenmiştir.

VI. ER DİYAGRAMI