



T.C.
SAKARYA ÜNİVERSİTESİ

BİLGİSAYAR VE BİLİŞİM BİLİMLERİ FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

WEB PROGRAMLAMA PROJE RAPORU

HASTANE RANDEVU SİSTEMİ

G201210571 - Ali Alhamdou - 2 C Grubu

https://github.com/ali-alhamdou/WEB_PROGRAMLAMA_PROJE2023_2024

SAKARYA

Ocak 1, 2024

WEB PROGRAMLAMA Dersi

Hastane Randevu Sistemi

Ali Alhamdou, G201210571

2.Öğretim C grubu

Özet

Proje bir hastanenin randevu sistemini kapsamaktadır. Randevu alınabilmesi için üye olunması ve authentication yapılması gerekmektedir. Proje içinde admin, doktor ve hasta olmak üzere üç rol bulunmaktadır. Hastalar randevu alabilir ve kendisine randevuları görüntüleyebilir. Doktorlar hastaların yaptığı işlemlere ek olarak kendisinden randevu alan hastaları görüntüleyebilir. Adminler hastalara ve doktorlara ek olarak sisteme doktor ekleyebilir ve sayfa içeriklerini güncelleyip silebilir ya da ekleyebilir.

© 2023Sakarya Üniversitesi.

Bu rapor benim özgün çalışmamdır. Faydalanmış olduğum kaynakları içerisinde belirttim. Herhangi bir kopya işleminde sorumluluk bana aittir.

Anahtar Kelimeler: asp.net 6, SQLServer, Identity, Localization, LINQ, API

1. GELİŞTİRİLEN YAZILIM

1.1. Amaç

Identity, localization, linq, entity framework vb. kullanarak ayağa kalkabilen hastane randevu web projesi geliştirilmesi.

1.2. Algoritma

Kullanıcılar “Register” bölümünden üye olur, “Login” bölümünden giriş yapar. İşlemleri sağlayan Identity sınıfıdır. Admin,doctor ve patient olmak üzere üç tip yetki vardır. Adminler için yetkilendirme veritabanından, doktorlar için admin panelinden yapılır ve kayıt olan her kullanıcı için patient yetkisi otomatik oluşur.

1.2.1.Online Randevu İşlemleri

Online randevu sekmesinde kullanıcıdan öncelikle departman seçmesi beklenmektedir. Seçilen departmandaki doktorlar sonrasında listelenmektedir. Ajax ile cascade işlemi uygulanmıştır. Doktor seçiminden sonra kullanıcıdan bir tarih seçmesi beklenmektedir. Saatleri göster butonuna tıklanıldığında yapılan işlemler şu şekildedir :

```

var doctor = _docMan.ListDoctorWithDepartment().FirstOrDefault(x => x.DoctorID == docid);
int? shipID = doctor.ShipID;
var ship = _shipMan.GetList().FirstOrDefault(x => x.ShipID == shipID);
TimeSpan shipS = ship.ShipStart;
var min = new TimeSpan(0, 0, 30, 00, 0000000);
var hour = new TimeSpan(0, 1, 30, 00, 0000000);
List<TimeSpan> shipTime = new List<TimeSpan>();
for (int i = 1; i <= 16; i++)
{
    shipTime.Add(shipS);

    if (i != 8)
    {
        shipS += min;
    }
    else
    {
        shipS += hour;
    }
}

```

Görsel : Randevu Saatleri Liste Oluşturma

- Öncelikle seçilen doktorun shift'ine ulaşılır ve shift'teki mesai saatleri dikkate alınarak randevu saatlerinin bulunduğu bir liste oluşturulur. Mesai saatinin başlangıcı - bitişi ve mola saatinin başlangıcı - bitişi dikkate alınarak bu saat aralıklarında yarımşar saat arayla veriler listeye eklenir. (Görsel : Randevu Saatleri Liste Oluşturma)

```

var reservartions = _resMan.GetList().Where(x => x.DoctorID == docid);
reservartions = reservartions.Where(y => y.ReservationDay == res);
reservartions = reservartions.Where(y => y.ReservationStatus == true);
if (reservartions != null)
{
    foreach (var item in reservartions)
    {
        foreach (var item2 in shipTime)
        {
            if (item.ReservationTime == item2)
            {
                shipTime.Remove(item2);
                break;
            }
        }
    }
}
var values = shipTime;
var jsonDepartments = JsonConvert.SerializeObject(values);
return Json(jsonDepartments);

```

Görsel : Randevu Saatleri Düzenleme

- Sonrasında seçilen doktor ve seçilen tarihe ait veriler veritabanındaki Reservation tablosundan alınarak bu verilerdeki saatler randevu saatlerinin bulunduğu listedekiyle karşılaştırılır. Burada uyuşan saatler varsa listeden çıkarılır. (Görsel : Randevu Saatleri Düzenleme)

Görsel : Arayüzdeki Son Durum

Online Randevu

Departman
Dermatoloji

Doktor
Doctor Doctor

Tarih
05.01.2024

Saatleri Göster

15:00:00	15:30:00	16:30:00	17:00:00
17:30:00	18:00:00	18:30:00	20:00:00
20:30:00	21:00:00	21:30:00	22:00:00
22:30:00	23:00:00	23:30:00	

Onayla / Accept

Randevu Al

- Sonuç olarak saatleri göster butonuna basıldığında doktora ait o tarihteki uygun randevu saatleri gösterilir. (Görsel : Arayüzdeki Son Durum)

1.2.2. Shift İşlemleri

Vardiya ID

Sil

Vardiya İsmi

Vardiya Başlangıcı

--:--

Vardiya Bitişi

--:--

Mola Başlangıcı

--:--

Mola Bitişi

--:--

Ekle

Vardiya ID : 5

Sabah Vardiyası

Vardiya Başlangıcı 08:00:00

Vardiya Bitişi 00:00:00

Mola Başlangıcı 12:00:00

Mola Bitişi 13:00:00

Vardiya ID : 6

Akşam Vardiyası

Vardiya Başlangıcı 15:00:00

Vardiya Bitişi 00:00:00

Mola Başlangıcı 19:00:00

Mola Bitişi 20:00:00

Shift ekleme ve silme işlemleri Admin Panel üzerinden yapılmaktadır. Her doktor kendi shiftini Doktor Panel üzerinden seçebilir. Shift işlemlerinde mesainin başlama ve bitiş saatiyle beraber mola zamanlarında belirtilmelidir. Shift oluştururken dikkat edilmesi gereken hususlar şu şekildedir :

- Her doktor en fazla mola zamanıyla beraber dokuz saat çalışacaktır.
- Mesaide ilk dört saatten sonra bir saatlik mola verilmesi zorunludur.
- Vardiyanın başlangıç zamanı ile bitiş zamanı farklı günler olmamalıdır.

1.2.3. Localization

Serdivan Hastanesi

Admin Panel

Randevularım

Merhaba admin@gmail.com!

Çıkış yap

Anasayfa

Hakkımızda

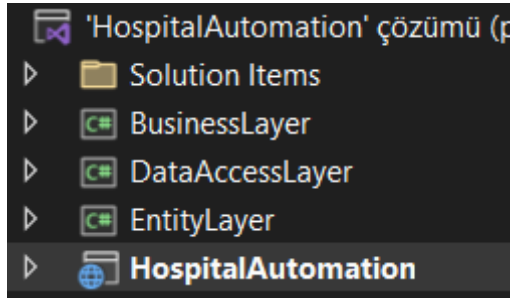
Doktorlarımız

Galeri

Online Randevu

Bu proje ingilizce ve türkçe dillerini desteklemektedir. Header üzerindeki bayrak simgesi olan dropdown üzerinden dil seçeneği değiştirilebilir.

1.2.4 Dosya yapısı

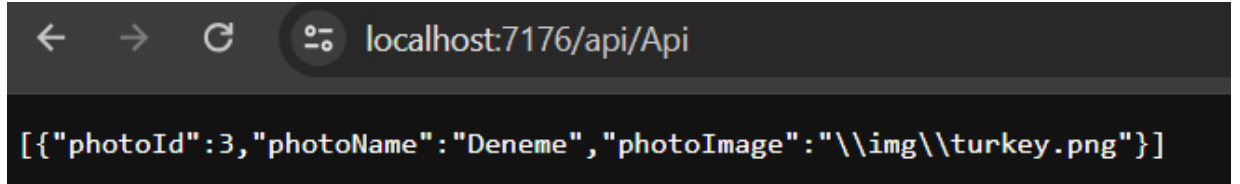


Proje dört katmandan oluşmaktadır. EntityLayer içerisinde kullandığımız sınıflar mevcuttur. DataAccessLayer katmanında database bağlantısı yapılmıştır. BusinessLayer katmanı validation işlemleri için kullanılır. HospitalAutomation proje dosyasıdır. Asp:Net Core projesi burada gerçekleştirilmiş diğer katmanlar library olarak kullanılmıştır.

1.2.5 Identity

Kullanıcılara rol verilmesi, proje içerisinde Authorization ve Authentication işlemlerinin doğru bir şekilde yerine getirilmesi Asp.net.core.Identity kütüphanesiyle gerçekleştirilmiştir.

1.2.6 API



EntitLayer katmanında veritabanı için bir Gallery sınıfı oluşturulmuştur. Sonrasında HospitalAutomation projesinin Controller dosyasında ApiController ile Gallery tablosundan veri çekebilen, silabilen ve güncelleyebilen servis hazırlanmıştır. Bu Api hizmeti yine aynı dosya dizininde bulunan CallApiController içerisinde çağırılmış ve ilgili hizmet projenin çalışır durumunda navbardaki Gallery sekmesinde verilmiştir.

2. SONUÇ

Bu projede Identity, Localization, Api, Linq gibi temel anahtar kelimeler kullanılmış doktorlar için belirlenen mesai saatlerine göre çalışan Asp Net Core MVC randevu sistemi oluşturulmuştur.