

**ANKARA ÜNİVERSİTESİ**  
**MÜHENDİSLİK FAKÜLTESİ**  
**BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**



**BLM 4531-A PROJE RAPORU**

**Hastane Randevu Alma Sitesi**

**Erşan Çağan BOZKURT**

**19290220**

**Github Kullanıcı Adı: canbozkurt27**

**Proje Video Linki:**

**<https://drive.google.com/file/d/1ozKNrsczbuTvoj1jv6gWko8nAvr67wwG/view?usp=sharing>**

**Enver BAĞCI**

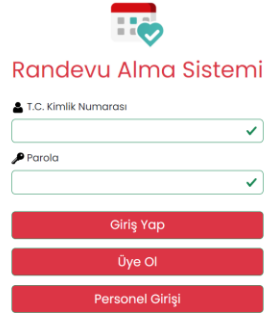
**15.01.2025**

## Projenin Tanıtımı

Projem hastanelerden randevu alma sitesidir. Kaydını yapan bir kullanıcı, ardından giriş yaptıktan sonra hastane doktoru, aşı ve aile hekimi randevuları alabilmektedir. Doktor randevusu için istediği şehir, ilçe, hastane ve klinik bilgisini seçtikten sonra filtrelenen doktorlardan randevu saati alabilmektedir. Ayrıca sitede personel girişi bulunmaktadır. Personel girişi yapan kişi admin yetkisinde olup hastane, klinik ve doktor ekleme,silme ve güncelleme yetkilerine sahiptir.

## Sitenin Görünümü

Siteye ilk giriş yapıldığında giriş(login) ekranı gelmektedir.



The login form for the 'Randevu Alma Sistemi' (Appointment System) is displayed. It features a red header with a calendar icon and a heart icon. Below the header, there are three input fields: 'T.C. Kimlik Numarası' (T.C. Identity Number), 'Parola' (Password), and 'Giriş Yap' (Login). The 'Giriş Yap' button is red. Below the login button, there are two more buttons: 'Üye Ol' (Sign Up) and 'Personel Girişi' (Staff Login), both in red.

Kullanıcının eğer bir hesabı bulunuyorsa doğrudan giriş yapabilir. Fakat hesabı yoksa önce üye olmalıdır.



The registration form for the 'Randevu Alma Sistemi' (Appointment System) is displayed. It features a red header with a calendar icon and a heart icon. Below the header, there is a red section titled 'Üye Ol' (Sign Up). The form includes several input fields: 'T.C. Kimlik Numarası' (T.C. Identity Number), 'Adınız' (Your Name), 'Soyadınız' (Your Surname), 'Doğum Tarihiniz' (Your Date of Birth) with a date picker, 'Şifrenizi Giriniz' (Enter your password), and 'Şifrenizi Tekrar Giriniz' (Repeat your password). A green 'Kaydol' (Register) button is at the bottom.

Buradan gerekli bilgileri girip üye olduktan sonra artık giriş yapabilir.

Kullanıcı giriş yaptıktan sonra karşısına şöyle bir sayfa çıkmaktadır:

Randevu Alma Sistemi

Hosgeldin, !!!!!!!

Aşı Randevusu Al

Hastane Randevusu Al

Aile Hekimi Randevusu Al

Hastanelerimiz

Istanbul	Fatih	Cerrahpaşa Tıp Fakültesi Hastanesi
Istanbul	Bahçelievler	Bahçelievler Devlet Hastanesi
Istanbul	Küçükçekmece	Istanbul Eğitim ve Araştırma Hastanesi
Ankara	Keçiören	Ankara Etilik Şehir Hastanesi
Ankara	Çankaya	Ankara Şehir Hastanesi
Ankara	Altındağ	Ankara Eğitim ve Araştırma Hastanesi
İzmir	Bornova	Ege Üniversitesi Hastanesi
İzmir	Konak	Tepecik Eğitim ve Araştırma Hastanesi
İzmir	Gaziemir	Gaziemir Devlet Hastanesi

Bu sayfada üst menü kısmında kullanıcının tc bilgisi, ona tıklandığında ise randevularım ve çıkış yap kısmı çıkmaktadır. Sayfanın üst kısmında aşı, hastane ve aile hekimi randevusu alma butonları bulunmaktadır. Altta ise randevu alınabilecek hastanelerin bilgisi bulunmaktadır. Aşı veya aile hekimi randevusu alma butonlarına basıldığında benzer sayfalar gelmektedir. Bunların farkı ise randevu alındığında birinin aşı randevusu diğerinin aile hekimi randevusu olarak gözükmesidir.

Randevu Alma Sistemi

Hosgeldin, !!!!!!!

< Aşı Randevusu Al

16 Ocak Perşembe

17 Ocak Cuma

18 Ocak Cumartesi

19 Ocak Pazar

20 Ocak Pazartesi

21 Ocak Salı

22 Ocak Çarşamba

9:00	
10:00	
11:00	
12:00	
13:00	
13:00	
13:20	
13:40	
14:00	
15:00	
16:00	

Randevu Al

Yukarıda bulunan sayfa aşı randevusu alma butonuna tıkladıktan sonra gelen sayfadır. Burada seçtiğim gün gri olmuştur ve seçtiğim saat butonu ise yeşil olmuştur. Saatlerin yanındaki aşağı inme butonlarına basarak o saatlere ait randevu saatleri görülebilir. Gün ve saati seçtikten sonra randevu al butonuna tıkladığında artık randevu alınmıştır ve kullanıcı bunu randevularım sayfasında görüntüleyebilir.



Üst kısımda da görüldüğü üzere kullanıcı tc numarasının üstüne basarak randevu bilgisi sayfası ve çıkış yap butonunu görebilir. Bir önceki görselde aşı randevusu alındıktan sonra ve randevu bilgisine tıklandıktan sonra gelen ekran şöyledir:

Randevularım				
Randevu Türü	Hastane Adı	Klinik Adı	Doktor Adı	Randevu Tarihi
Aşı Randevusu	Sağlık Ocağı	Aile Hekimi Polikliniği	Ekrem Koru	17 Ocak Cuma 13:20

<https://localhost:44308/Home/Index>

Randevu bilgisi sayfasında aldığım randevu gözükmemektedir. Burada aşı randevusu aldığım için randevunun türü aşıdır. Gün ve saatte doğru bir şekilde gözükmemektedir. Bu ekrandaki hastane, klinik ve doktor adı bilgisi ise kullanıcılara kayıt yaparken otomatik olarak atanan doktor bilgisinden gelmektedir.

Ana sayfaya döndüğümüzde aile hekimi randevusu alma butonuna tıklandığında önceden dediğim gibi aşı randevu sayfasına benzer bir sayfa çıkmaktadır. Fakat aile hekimi randevusu alındığında randevularım kısmında randevu türü aile hekimi randevusu olarak gözükmemektedir.

Ana sayfada hastane randevusu al butonuna tıklandığında ise şöyle bir sayfa gelmektedir:

Randevu Alma Sistemi

Hosgeldin, !!!!!!!

Şehirler

---Şehir Seçiniz---

İlçeler

---İlçe Seçiniz---

Hastaneler

---Hastane Seçiniz---

Klinikler

---Klinik Seçiniz---

Bu ekran doktor filtrelemek için kullanılır. Burada istediğiniz şehri, ilçeyi, hastaneyi ve kliniği seçerek ona bağlı olan doktorların listesini görüntülenmesini sağlarsınız. İlk başta şehir dışındaki diğer seçeneklerin kilitli olmasının sebebi birbirlerine bağlı olmaları ve bir üstteki seçenek seçildikçe açılmalıdır. İstenilen filtrelemeler yapıldıktan sonra ekran şöyle olur:

Randevu Alma Sistemi

Hosgeldin, !!!!!!!

Şehirler

Istanbul

İlçeler

Fatih

Hastaneler

Cerrahpaşa Tıp Fakültesi Hastanesi

Klinikler

Cildiye(Dermatoloji)

Doktor Ara

Doktor ara butonuna basıldığında bu kriterlere uygun doktor listesi gözükmetedir.

Randevu Alma Sistemi

Hosgeldin, !!!!!!!

< Hastane Listesi

Ömer Faruk Cengiz	Cerrahpaşa Tıp Fakültesi Hastanesi	Cildiye(Dermatoloji)	Randevu Al
Sedat Toprak	Cerrahpaşa Tıp Fakültesi Hastanesi	Cildiye(Dermatoloji)	Randevu Al
Berat Yılmaz	Cerrahpaşa Tıp Fakültesi Hastanesi	Cildiye(Dermatoloji)	Randevu Al

Listelenen doktorlardan biri seçilip randevu al butonuna basıldıktan sonra bizi artık o doktora ait randevu sayfasına yönlendirmektedir. Burada en üstteki doktor seçildikten sonra karşımıza şöyle bir ekran gelmektedir:

Randevu Alma Sistemi

Hosgeldin,!!!!!!!

< Doktor Randevusu Al

16 Ocak Perşembe17 Ocak Cuma18 Ocak Cumartesi19 Ocak Pazar20 Ocak Pazartesi21 Ocak Salı22 Ocak Çarşamba

9:00

10:00

10:00

10:20

10:40

11:00

12:00

13:00

14:00

15:00

16:00

Randevu Al

Burada gün ve saat tercihleri yapıldıktan ve randevu al butonuna basıldıktan sonra randevu alınmış olur. Randevu kontrolü için randevu bilgi sayfasına bakalım:

Randevu Alma Sistemi

Hosgeldin,!!!!!!!

Randevularım

Randevu Türü	Hastane Adı	Klinik Adı	Doktor Adı	Randevu Tarihi
Aşı Randevusu	Sağlık Ocağı	Aile Hekimi Polikliniği	Ekrem Kuru	17 Ocak Cuma 13:20
Doktor Randevusu	Cerrahpaşa Tıp Fakültesi Hastanesi	Cildiye(Dermatoloji)	Berat Yılmaz	21 Ocak Salı 10:20

Görüldüğü üzere seçtiğim hastane, klinik, doktor, gün ve saat bilgilerine göre randevum gözükmemektedir. Doktor randevusu olduğu için randevu türüm doktor randevusudur. Diğer randevum ise bir önceki yaptığım işlemde kalan randevudur. Son olarak seçtiğim randevu tarihleri başka kullanıcılar tarafından seçilmesin diye o gün ve saate ait buton disabled(seçilemez) olur.

Kullanıcı sayfası dışında admin paneli de bulunur. Admin işlemleri yapmak için giriş ekranında personel girişin seçilmesi gerekir ve sistemde tanımlanan kullanıcı adı ve şifre bilgisiyle giriş yapılır. Admin panelinde hastane, klinik, doktor ekleme, silme ve

güncelleme işlemleri yapılabilir.

Admin Paneli

Hastaneler

Klinikler

Doktorlar

Çıkış Yap

+ Yeni Hastane Ekle

Hastane ID	İl Adı	İlçe Adı	Hastane Adı	Sil	Güncelle
1	İstanbul	Fatih	Cerrahpaşa Tıp Fakültesi Hastanesi	Sil	Güncelle
2	İstanbul	Bahçelievler	Bahçelievler Devlet Hastanesi	Sil	Güncelle
3	İstanbul	Küçükçekmece	İstanbul Eğitim ve Araştırma Hastanesi	Sil	Güncelle
4	Ankara	Keçiören	Ankara Etilik Şehir Hastanesi	Sil	Güncelle
5	Ankara	Çankaya	Ankara Şehir Hastanesi	Sil	Güncelle
6	Ankara	Altındağ	Ankara Eğitim ve Araştırma Hastanesi	Sil	Güncelle
7	İzmir	Bornova	Ege Üniversitesi Hastanesi	Sil	Güncelle
8	İzmir	Konak	Tepecik Eğitim ve Araştırma Hastanesi	Sil	Güncelle
9	İzmir	Gaziemir	Gaziemir Devlet Hastanesi	Sil	Güncelle
1012	Ankara	Çankaya	Sağlık Ocağı	Sil	Güncelle

Hastane listesi ve düzenleme ekranı

Admin Paneli

Hastaneler

Klinikler

Doktorlar

Çıkış Yap

+ Yeni Klinik Ekle

Klinik ID	Hastane Adı	Klinik Adı	Sil	Güncelle
1	Cerrahpaşa Tıp Fakültesi Hastanesi	Cildiye(Dermatoloji)	Sil	Güncelle
2	Cerrahpaşa Tıp Fakültesi Hastanesi	Genel Cerrahi	Sil	Güncelle
3	Cerrahpaşa Tıp Fakültesi Hastanesi	Çocuk Sağlığı ve Hastalıkları	Sil	Güncelle
4	Cerrahpaşa Tıp Fakültesi Hastanesi	İç Hastalıkları(Dahiliye)	Sil	Güncelle
5	Cerrahpaşa Tıp Fakültesi Hastanesi	Kulak-Burun-Böğaz(KBB)	Sil	Güncelle
6	Cerrahpaşa Tıp Fakültesi Hastanesi	Gastroenteroloji	Sil	Güncelle
7	Cerrahpaşa Tıp Fakültesi Hastanesi	Kardiyoloji	Sil	Güncelle
8	Cerrahpaşa Tıp Fakültesi Hastanesi	Nöroloji	Sil	Güncelle
1010	Bahçelievler Devlet Hastanesi	Cildiye(Dermatoloji)	Sil	Güncelle
1011	Bahçelievler Devlet Hastanesi	Genel Cerrahi	Sil	Güncelle

Klinik listesi ve düzenleme ekranı

Admin Paneli

Hastaneler

Klinikler

Doktorlar

Çıkış Yap

+ Yeni Doktor Ekle

Aranacak doktor adını giriniz

Ara

Doktor ID	Hastane Adı	Klinik Adı	Doktor Adı	Sil	Güncelle
1	Cerrahpaşa Tıp Fakültesi Hastanesi	Cildiye(Dermatoloji)	Ömer Faruk Cengiz	Sil	Güncelle
2	Cerrahpaşa Tıp Fakültesi Hastanesi	Genel Cerrahi	Berkay Akparlar	Sil	Güncelle
3	Cerrahpaşa Tıp Fakültesi Hastanesi	Çocuk Sağlığı ve Hastalıkları	Eren Kıranatlı	Sil	Güncelle
4	Cerrahpaşa Tıp Fakültesi Hastanesi	İç Hastalıkları(Dahiliye)	Ahmet Çakıl	Sil	Güncelle
5	Cerrahpaşa Tıp Fakültesi Hastanesi	Kulak-Burun-Böğaz(KBB)	Zeynep Gürler	Sil	Güncelle
6	Cerrahpaşa Tıp Fakültesi Hastanesi	Gastroenteroloji	Ebru Gün	Sil	Güncelle
7	Cerrahpaşa Tıp Fakültesi Hastanesi	Kardiyoloji	Hakan İlkey	Sil	Güncelle
8	Cerrahpaşa Tıp Fakültesi Hastanesi	Nöroloji	Buket Dağdeviren	Sil	Güncelle
9	Bahçelievler Devlet Hastanesi	Cildiye(Dermatoloji)	Soner Kabak	Sil	Güncelle
10	Bahçelievler Devlet Hastanesi	Genel Cerrahi	Duygu Özlü	Sil	Güncelle

Doktor listesi ve düzenleme ekranı(Doktor arama ve sayfalama seçeneği mevcuttur)

Doktor sayfasından örnek vererek tüm sayfalar için yapılabilecek işlemleri anlatacağım. Yeni doktor ekle tuşuna basıldığında şöyle bir ekran gelmektedir:

Admin Paneli

Hastaneler

Klinikler

Doktorlar

Çıkış Yap

Hastaneler

---Hastane Seçiniz---

Klinikler

---Klinik Seçiniz---

Doktor Adı

Doktor Ekle

Burada seçeneklere tıklandığında gelen dropdownlistten istenilen bilgiler seçilerek o bilgilere ait yani o hastane adı ve kliniğine ait doktor adı da girildikten sonra doktor eklemesi yapılabilir.

Güncelleme butonuna basıldığında aşağıdaki gibi bir ekran gelmektedir. Bu ekranda değiştirilmek istenen bilgiler değiştirilip onaylanırsa o doktora ait bilgiler güncellenir.

Admin Paneli

Hastaneler

Klinikler

Doktorlar

Çıkış Yap

Hastane Adı

Cerrahpaşa Tıp Fakültesi Hastanesi

Klinik Adı

Kardiyoloji

Doktor Adı

Hakan İlkay

Doktor Güncelle

Sil tuşuna basıldığında ise pop-up olarak silme onayı sormaktadır, işlem onaylanınca silme gerçekleşir.

localhost:44308 web sitesinin mesajı

Gerçekten Silmek İstiyor musunuz?

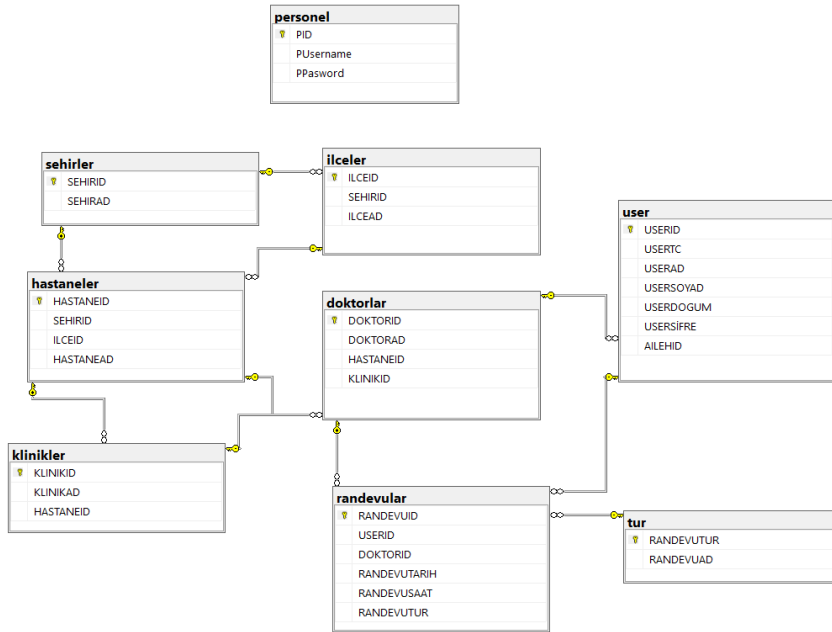
Tamam

İptal



## Database Kısmı

Veri tabanımı diyagramım üzerinden açıklayacağım.



Diyagramımda tüm tablolarım ve aralarındaki ilişkiler bulunmaktadır. user

tablosu: Kayıt yapan kullanıcının bilgileri bulunmaktadır.

personel tablosu: Admin giriş bilgisini tutmaktadır.

sehirler tablosu: Hastanelerimin bulunduğu şehirleri içermektedir.

ilceler tablosu: Bu şehirlere bağlı olan ilçeleri içermektedir.

hastaneler tablosu: Şehire ve ilçeye bağlı olan hastaneler bulunmaktadır.

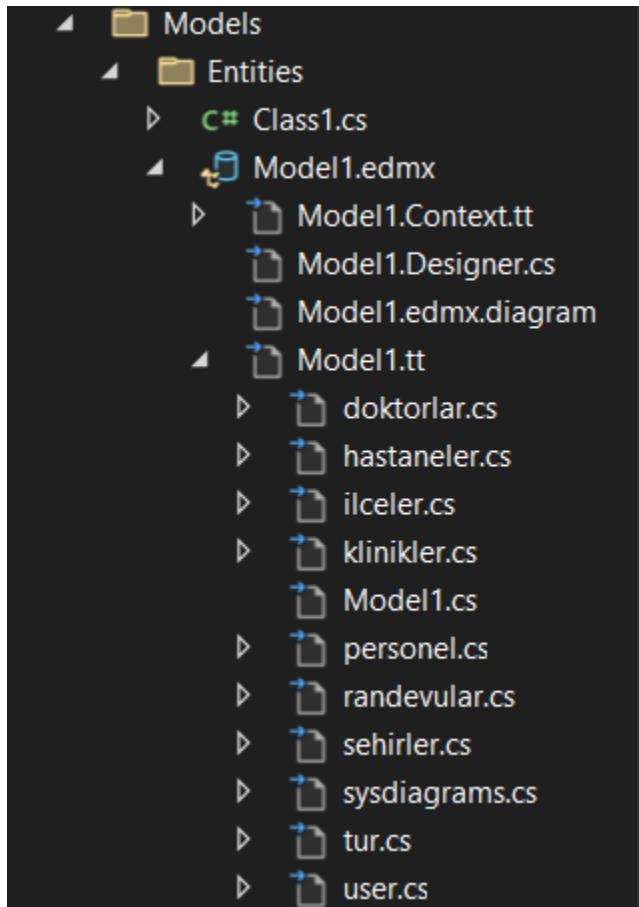
kliniker tablosu: Hastaneye bağlı olan polikliniklerin bilgisi bulunmaktadır.

doktorlar tablosu: Hastane ve ona bağlı olan kliniklerdeki doktorların bilgisi bulunmaktadır.

randevular tablosu: Alınan randevuların tür, gün, saat ve doktor bilgisini içermektedir.

tur tablosu: Randevunun tür bilgisini bulundurmaktadır.

## Model Kısmı



## Birkaç örnek:

```
namespace hastanerandevu.Models.Entities
{
    using System;
    using System.Collections.Generic;

    15 references
    public partial class doktorlar
    {
        [System.Diagnostics.CodeAnalysis.SuppressMessage("Microsoft.Usage", "CA2214:DoNotCallOverridableMethodsInConstructors")]
        1 reference
        public doktorlar()
        {
            {
                this.randevular = new HashSet<randevular>();
                this.user = new HashSet<user>();
            }
        }

        3 references
        public int DOKTORID { get; set; }
        5 references
        public string DOKTORAD { get; set; }
        3 references
        public int HASTANEID { get; set; }
        6 references
        public int KLINIRID { get; set; }

        1 reference
        public virtual hastaneler hastaneler { get; set; }
        1 reference
        public virtual klinikler klinikler { get; set; }
        [System.Diagnostics.CodeAnalysis.SuppressMessage("Microsoft.Usage", "CA2227:CollectionPropertiesShouldBeReadOnly")]
        1 reference
        public virtual ICollection<randevular> randevular { get; set; }
        [System.Diagnostics.CodeAnalysis.SuppressMessage("Microsoft.Usage", "CA2227:CollectionPropertiesShouldBeReadOnly")]
        1 reference
        public virtual ICollection<user> user { get; set; }
    }
}
```

Yukarıda doktorlar tabloma ait doktorlar.cs dosyası ve içindeki değişken, ilişkiler gibi bilgiler bulunmaktadır.

```
namespace hastanerandevu.Models.Entities
{
    using System;
    using System.Collections.Generic;
    using System.ComponentModel;
    using System.ComponentModel.DataAnnotations;
    7 references
    public partial class user
    {
        [System.Diagnostics.CodeAnalysis.SuppressMessage("Microsoft.Usage", "CA2214:DoNotCallOverridableMethodsInConstructors")]
        0 references
        public user()
        {
            this.randevular = new HashSet<randevular>();
        }
        4 references
        public int USERID { get; set; }
        [Required(ErrorMessage = "Lütfen TC Kimlik No Giriniz", AllowEmptyStrings = false)]
        10 references
        public string USERTC { get; set; }
        [Required(ErrorMessage = "Lütfen Adınızı Giriniz", AllowEmptyStrings = false)]
        1 reference
        public string USERAD { get; set; }
        [Required(ErrorMessage = "Lütfen Soyadınızı Giriniz", AllowEmptyStrings = false)]
        0 references
        public string USERSOYAD { get; set; }

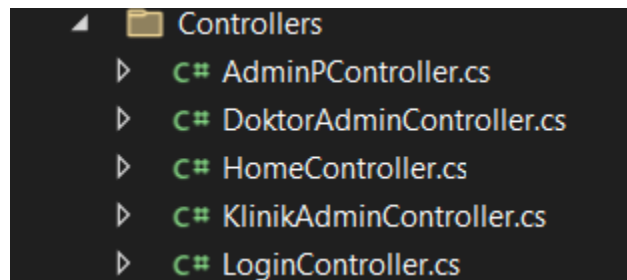
        [DataType(DataType.Date)]
        [DisplayFormat(ApplyFormatInEditMode = true, DataFormatString = "{0:MM/dd/yyyy}")]
        0 references
        public System.DateTime USERDOGUM { get; set; }
        [Required(ErrorMessage = "Lütfen Şifrenizi Giriniz", AllowEmptyStrings = false)]
        [DataType(System.ComponentModel.DataAnnotations.DataType.Password)]
        [StringLength(50, MinimumLength = 6, ErrorMessage = "Şifreniz en az 6 hane olmalıdır")]
        2 references
        public string USERSİFRE { get; set; }
        [Compare("USERSİFRE", ErrorMessage = "Şifre eşleşmiyor")]
        [DataType(System.ComponentModel.DataAnnotations.DataType.Password)]
        0 references
        public string USERRESİFRE { get; set; }
        0 references
        public int AILEHID { get; set; }

        [System.Diagnostics.CodeAnalysis.SuppressMessage("Microsoft.Usage", "CA2227:CollectionPropertiesShouldBeReadOnly")]
        1 reference
        public virtual ICollection<randevular> randevular { get; set; }
        0 references
        public virtual doktorlar doktorlar { get; set; }
    }
}
```

Yukarıda user tabloma ait user.cs dosyası bulunmaktadır. Değişkenler dışında giriş yapıldığında veya kayıt olduğunda gerekli mesajlar, data type tanımlamaları ve display bilgileri içermektedir.

## Backend Kısmı

Backend kısmında controllerlarım bulunmaktadır. Bu controllerlardaki kodlar veri tabanında gelen bilgilerle işlemler yaparak view kısmında görüntülenecek bilgilerin arka plan kodlarını bulundurur. Controller'larım;



### Login Controller:

```

1  using hastanerandevu.Models.Entities;
2  using System;
3  using System.Collections.Generic;
4  using System.EnterpriseServices.CompensatingResourceManager;
5  using System.Linq;
6  using System.Threading.Tasks;
7  using System.Web;
8  using System.Web.Mvc;
9  using System.Web.Security;
10
11 namespace hastanerandevu.Controllers
12 {
13     public class LoginController : Controller
14     {
15         // GET: Login
16         hastaneEntities db = new hastaneEntities();
17         public ActionResult Register()
18         {
19             return View();
20         }
21         [HttpPost]
22         [ValidateAntiForgeryToken]
23         public ActionResult Register(user U)
24         {
25             if (ModelState.IsValid)
26             {
27                 using (hastaneEntities dc = new hastaneEntities())
28                 {
29                     if (dc.user.Any(x => x.USERTC == U.USERTC))
30                     {
31                         ViewBag.DuplicateMessage = "Kullanıcı zaten mevcut";
32                         return View(U);
33                     }
34                     dc.user.Add(U);
35                     dc.SaveChanges();
36                     ModelState.Clear();
37                     U = null;
38                     ViewBag.Message = "Kaydınız başarıyla tamamlanmıştır";
39                 }
40             }
41             return View(U);
42         }
43         [HttpGet]
44         public ActionResult Login()
45         {
46             return View();
47         }
48         [HttpPost]
49         [ValidateAntiForgeryToken]

```

Yukardaki controller kısmında yeni kayıt yapacak bir kullanıcının girdiği bilgilerin veritabanına eklenmesini sağlayan fonksiyonum bulunmaktadır. Ayrıca işlemlerin doğru veya yanlış yapılması halinde gönderilen mesaj bilgilerini içermektedir.

```

    }
    [HttpPost]
    [ValidateAntiForgeryToken]
    0 references
    public ActionResult Login(user US)
    {
        var checkLogin = db.user.Where(x => x.USERTC.Equals(US.USERTC) && x.USERSİFRE.Equals(US.USERSİFRE)).FirstOrDefault();
        if (checkLogin != null)
        {
            Session["USERCSS"] = US.USERTC.ToString();
            TempData["UserName"] = checkLogin.USERAD;
            FormsAuthentication.SetAuthCookie(US.USERTC, false);
            return RedirectToAction("Index", "Home");
        }
        else
        {
            ViewBag.Notification = "Yanlış TC veya şifre";
        }
        return View();
    }
    [HttpGet]
    0 references
    public ActionResult PersonelG()
    {
        return View();
    }
    [HttpPost]
    [ValidateAntiForgeryToken]
    0 references
    public ActionResult PersonelG(personel PS)
    {
        var checkLogin = db.personel.Where(x => x.PUsername.Equals(PS.PUsername) && x.PPassword.Equals(PS.PPassword)).FirstOrDefault();
        if (checkLogin != null)
        {
            Session["PUsernameSS"] = PS.PUsername.ToString();
            return RedirectToAction("Index", "AdminP");
        }
        else
        {
            ViewBag.Notification = "Yanlış kullanıcı adı veya şifre";
        }
        return View();
    }
    0 references
    public ActionResult Logout()
    {
        FormsAuthentication.SignOut();
        return RedirectToAction("Login", "Login");
    }
}

```

Yukardaki controller kısmında ise sisteme giriş yapacak bir kullanıcının girdiği bilgilerin veri tabanındaki bilgilerle karşılaştırılması ve bu bilgilerin doğru veya yanlış olmasına göre yapılacak işlemlerin fonksiyonu bulunmaktadır. Ayrıca giriş yapan bir üyenin çıkmasını sağlayan logout işlemi bulunmaktadır.

### Home Controller:

Bu controllerda kullanıcının giriş yapmasının ardından seçebileceği işlemlerin arka plan kodları bulunmaktadır.

```

1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Net.Sockets;
5 using System.Security.Policy;
6 using System.Web;
7 using System.Web.Mvc;
8 using System.Web.Security;
9 using Newtonsoft.Json;
10 using hastaneraandevu.Models.Entities;
11 using System.Security.Cryptography.Xml;
12
13 namespace hastaneraandevu.Controllers
14 {
15     [0 references]
16     public class HomeController : Controller
17     {
18         // GET: Home
19         hastaneEntities db=new hastaneEntities();
20         [0 references]
21         public ActionResult Index()
22         {
23             var degerler=db.hastaneler.ToList();
24             return View(degerler);
25         }
26         Class1 cs = new Class1();
27
28         [0 references]
29         public ActionResult Cascading()
30         {
31             ViewBag.SehirList = new SelectList(ilgetir(), "SEHIRID", "SEHIRAD");
32             return View();
33         }
34         [1 reference]
35         public List<sehirler> ilgetir()
36         {
37             List<sehirler> sehirler=db.sehirler.ToList();
38             return sehirler;
39         }
40         [0 references]
41         public ActionResult ilcegetir(int SEHIRID)
42         {
43             List<ilceler> selectlist=db.ilceler.Where(x=>x.SEHIRID==SEHIRID).ToList();
44             ViewBag.ilcelist= new SelectList(selectlist,"ILCEID", "ILCEAD");
45             return PartialView("ilcegoster");
46         }
47         [0 references]
48         public ActionResult hastanegetir(int ILCEID)
49         {
50             List<hastaneler> selectlist = db.hastaneler.Where(x => x.ILCEID == ILCEID).ToList();
51             ViewBag.Hastanelist = new SelectList(selectlist, "HASTANEID", "HASTANEAD");
52             return PartialView("hastanegoster");
53         }
54         [0 references]
55         public ActionResult klinikgetir(int HASTANEID)
56         {
57             List<klinikler> selectlist = db.klinikler.Where(x => x.HASTANEID == HASTANEID).ToList();
58             ViewBag.Kliniklist = new SelectList(selectlist, "KLINIKID", "KLINIKAD");
59             return PartialView("klinikgoster");
60         }
61         [0 references]
62         public ActionResult doktorgetir(int KLINIKID)
63         {
64             List<doktorlar> selectlist = db.doktorlar.Where(x => x.KLINIKID == KLINIKID).ToList();
65             ViewBag.Doktorlist = new SelectList(selectlist, "DOKTORID", "DOKTORAD");
66             return PartialView("doktorgoster");
67         }
68     }
69 }

```

Yukardaki kısımda giriş yaptıktan sonra gelen ana sayfanın fonksiyonu bulunmaktadır(Index) ve listeleme için değer döndürülmüştür. Geri kalan kısımda ise hastaneden randevu alma kısmına tıklandığında doktor aramak için ekrana çıkan filtreleme sayfasının fonksiyonlarıdır. Bu sayfa bir üstteki seçeneğe bağlı olacak şekilde cascading dropdownlist kullandığı için arama kriterlerine ait fonksiyonlar yazılmıştır ve sistemin çalışması için bunlar üzerinden partialviewlar oluşturulmuştur. Birinin üzerinden örnek verecek olursam ilcegetir fonksiyonu bir üstte seçilen şehire ait ilçelerin bilgisini getirmek için yazılan arkaplan kodunu içermektedir.

```

0 references
public ActionResult AsiRandevu()
{
    return View();
}

[HttpPost]
0 references
public ActionResult AsiRandevuAl(string data)
{
    randevular rande = new randevular();
    var dataObject = JsonConvert.DeserializeObject<randevular>(data);
    var kullanicadi = User.Identity.Name;
    var kullanıcı = db.user.FirstOrDefault(x => x.USERTC == kullanicadi);
    rande.USERID = kullanıcı.USERID;
    rande.DOKTORID = dataObject.DOKTORID;
    rande.RANDEVUTARIH = dataObject.RANDEVUTARIH;
    rande.RANDEVUSAAT = dataObject.RANDEVUSAAT;
    rande.RANDEVUTUR = dataObject.RANDEVUTUR;

    db.randevular.Add(rande);
    db.SaveChanges();
    return Json(true);
}

0 references
public ActionResult AileHekimi()
{
    return View();
}

[HttpPost]
0 references
public ActionResult AilehekimRandevuAl(string data)
{
    randevular rande = new randevular();
    var dataObject = JsonConvert.DeserializeObject<randevular>(data);
    var kullanicadi = User.Identity.Name;
    var kullanıcı = db.user.FirstOrDefault(x => x.USERTC == kullanicadi);
    rande.USERID = kullanıcı.USERID;
    rande.DOKTORID = dataObject.DOKTORID;
    rande.RANDEVUTARIH = dataObject.RANDEVUTARIH;
    rande.RANDEVUSAAT = dataObject.RANDEVUSAAT;
    rande.RANDEVUTUR = dataObject.RANDEVUTUR;
    db.randevular.Add(rande);
    db.SaveChanges();
    return Json(true);
}

```

Controllerların bu kısmında ise aşı ve aile hekimi randevusu alma işlemlerinin fonksiyonları bulunmaktadır. Bu fonksiyonlar viewdan gelen bilgilerin veri tabanına eklenmesini sağlar. Viewdan gelen bilgiler, kullanıcının ekranda seçtiği gün ve saat butonları gibi bilgilerdir. Bu bilgiler de veri tabanına eklenerek kullanıcıya ait randevuyu oluşturur.



```

105
106
107 0 references
108 public ActionResult Arama()
109 {
110     return View();
111 }
112
113 [HttpPost]
114 0 references
115 public ActionResult Arama(Class1 id)
116 {
117     var model=db.doktorlar.Where(x=>x.KLINIKID==id.KLINIKID).ToList();
118     var firma = db.doktorlar.Where(x => x.KLINIKID == id.KLINIKID).Select(x => x.DOKTORID).FirstOrDefault();
119     ViewBag.viewfirma = firma;
120     return View("Arama",model);
121 }
122
123 0 references
124 public ActionResult Doktor()
125 {
126     return View();
127 }
128
129 [HttpPost]
130 0 references
131 public ActionResult DoktorRandevuAl(string data,doktorlar p1)
132 {
133     var u = db.doktorlar.Find(p1.DOKTORID);
134     randevular rande = new randevular();
135     var dataObject = JsonConvert.DeserializeObject<randevular>(data);
136     var kullaniciadi = User.Identity.Name;
137     var kullanici = db.user.FirstOrDefault(x => x.USERTC == kullaniciadi);
138     rande.USERID = kullanici.USERID;
139     rande.DOKTORID = dataObject.DOKTORID;
140     rande.RANDEVUTARIH = dataObject.RANDEVUTARIH;
141     rande.RANDEVUSAAT = dataObject.RANDEVUSAAT;
142     rande.RANDEVUTUR = dataObject.RANDEVUTUR;
143     db.randevular.Add(rande);
144     db.SaveChanges();
145     return Json(true);
146 }
147
148 0 references
149 public ActionResult Goster()
150 {
151     var kullaniciadi = User.Identity.Name;
152     var kullanici = db.user.FirstOrDefault(x => x.USERTC == kullaniciadi);
153     var model = db.randevular.Where(x => x.USERID == kullanici.USERID).ToList();
154     return View(model);
155 }
156
157 }

```

Bu kısımda ise arama fonksiyonu,iki önceki kısımda bahsettiğim filtreleme işine yarayan fonksiyon sonucunda belli kriterlere göre şehir, ilçe, hastane ve klinik araması yapıldıktan sonra o kriterleri sağlayan doktorların bilgisini döndürülmesini sağlar. Doktorrandevual bölümü ise randevu alma ekranında kullanıcıların gününü ve saatini seçtikten sonra aldığı randevunun veri tabanına eklenmesini sağlar. Son olarak goster fonksiyonu ise kullanıcıların randevusunu görüntülemek için veritabanında karşılaştırma yaparak o kullanıcıya ait randevu bilgisini döndürür.

### AdminP Controller:

Bu controller ve DoktorAdmin, KlinikAdmin controllerları aynı işlevi görmektedir. Farkları AdminP controllerı admin panelindeki hastane sayfası için, DoktorAdmin controllerı admin panelindeki doktor sayfası için, KlinikAdmin controllerı ise admin panelindeki klinik sayfası için yapılmıştır. Ama içerdiği fonksiyonlar bakımından hepsi farklı tablolar için CRUD işlemlerini gerçekleştirir. O yüzden bu 3 controllerı sadece Adminp controllerı üzerinden analatacağım.



```

1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Web;
5  using System.Web.Helpers;
6  using System.Web.Mvc;
7  using hastanerandevu.Models.Entities;
8
9  namespace hastanerandevu.Controllers
10 {
11     public class AdminPController : Controller
12     {
13         // GET: AdminP
14         hastaneEntities db = new hastaneEntities();
15         public ActionResult Index()
16         {
17             var degerler = db.hastaneler.ToList();
18             return View(degerler);
19         }
20         Class1 cs = new Class1();
21         [HttpGet]
22
23         public ActionResult Ekleme()
24         {
25             List<SelectListItem> firmas = (from x in db.sehirler.ToList()
26                                         select new SelectListItem
27                                         {
28                                             Text = x.SEHIRAD,
29                                             Value = x.SEHIRID.ToString(),
30                                         }).ToList();
31             ViewBag.firms = firmas;
32
33             List<SelectListItem> degerler = (from i in db.ilceler.ToList()
34                                         select new SelectListItem
35                                         {
36                                             Text = i.ILCEAD,
37                                             Value = i.ILCEID.ToString(),
38                                         }).ToList();
39             ViewBag.dgr = degerler;
40             ViewBag.SehirList = new SelectList(ilgetir(), "SEHIRID", "SEHIRAD");
41             return View();
42         }
43         [HttpPost]
44         public ActionResult Ekleme(hastaneler p1)
45         {
46             hastaneler ab = new hastaneler();
47             ab.HASTANEAD = p1.HASTANEAD;
48             ab.SEHIRID = p1.SEHIRID;
49             ab.ILCEID = p1.ILCEID;
50             db.hastaneler.Add(ab);
51             db.SaveChanges();
52             return RedirectToAction("Index");

```

Yukardaki kısımda admin panelinde hastanelerin listelenmesi için değer döndüren bir fonksiyon bulunuyor. Ardından ise adından da anlaşılacağı üzere ekleme işlemini sağlayan ve burdaki dropdownlistlerin id olarak değil string olarak dönmesini sağlayan fonksiyonlar bulunmaktadır.

```

54 public ActionResult Sil(int id)
55 {
56     var hstnlr = db.hastaneler.Find(id);
57     db.hastaneler.Remove(hstnlr);
58     db.SaveChanges();
59     return RedirectToAction("Index");
60 }
61 0 references
62 public ActionResult Güncelle(int id)
63 {
64     var ur = db.hastaneler.Find(id);
65     List<SelectListItem> ils = (from x in db.sehirler.ToList()
66                               select new SelectListItem
67                               {
68                                   Text = x.SEHIRAD,
69                                   Value = x.SEHIRID.ToString(),
70                               }).ToList();
71     ViewBag.shr = ils;
72     List<SelectListItem> ilces = (from i in db.ilceler.ToList()
73                                  select new SelectListItem
74                                  {
75                                      Text = i.ILCEAD,
76                                      Value = i.ILCEID.ToString(),
77                                  }).ToList();
78     ViewBag.ilc = ilces;
79     return View("Güncelle", ur);
80 }
81 0 references
82 public ActionResult Guncel(hastaneler p1)
83 {
84     var u = db.hastaneler.Find(p1.HASTANEID);
85     u.HASTANEAD = p1.HASTANEAD;
86     var fr = db.sehirler.Where(m => m.SEHIRID == p1.sehirler.SEHIRID).FirstOrDefault();
87     u.SEHIRID = fr.SEHIRID;
88     var mrk = db.ilceler.Where(m => m.ILCEID == p1.ilceler.ILCEID).FirstOrDefault();
89     u.ILCEID = mrk.ILCEID;
90     db.SaveChanges();
91     return RedirectToAction("Index");
92 }
93 1 reference
94 public List<sehirler> ilgetir()
95 {
96     List<sehirler> sehirler = db.sehirler.ToList();
97     return sehirler;
98 }
99 0 references
100 public ActionResult ilcegetir(int SEHIRID)
101 {
102     List<ilceler> selectlist = db.ilceler.Where(x => x.SEHIRID == SEHIRID).ToList();
103     ViewBag.ILcelist = new SelectList(selectlist, "ILCEID", "ILCEAD");
104     return PartialView("ilcegoster");
105 }

```

Bu kısımda ise silme işlemini sağlayan bir fonksiyon ve hastanelerin bilgisini güncellemek için arka plan işlemlerini sağlayan kodlar bulunmaktadır. En alttaki kısım ise ekleme ve güncelleme ekranındaki dropdownlistlerin cascading olarak görüntülenmesi için yazılan fonksiyonlardır.

## Frontend Kısım

Frontend kısmında yazdığım kodlar view dosyasında bulunmaktadır.



```

76 <body>
77 @using (Html.BeginForm())
78 {
79     @Html.AntiForgeryToken()
80     @Html.ValidationSummary(true)
81
82     <div class="giris">
83
84         
85         <h1 class="text-danger">Randevu Alma Sistemi</h1>
86
87         <form class="needs-validation">
88
89             <div class="abc was-validated">
90                 
91                 <label class="form-label" for="USERTC">T.C. Kimlik Numarası</label>
92                 <br>
93                 @Html.EditorFor(model => model.USERTC, new
94                 {
95                     htmlAttributes = new { @class = "form-control" }
96                 })
97                 <div class="text-danger">
98                     @Html.ValidationMessageFor(model => model.USERTC, "", new { @class = "text-danger" })
99                 </div>
100             </div>
101
102             <div class="abc was-validated">
103                 
104                 <label class="form-label" for="USERSİFRE">Parola</label>
105                 <br>
106                 @Html.EditorFor(model => model.USERSİFRE, new { htmlAttributes = new { @class = "form-control" } })
107                 <div class="text-danger">
108                     @Html.ValidationMessageFor(model => model.USERSİFRE, "", new { @class = "text-danger" })
109                 </div>
110             </div>
111
112             <input class="btn btn-success w-100 text-bg-danger" type="submit" value="Giriş Yap">
113             <br>
114             <a href="/Login/Register">
115                 <input class="btn btn-success w-100 text-bg-danger" type="button" value="Üye Ol">
116             </a>
117             <a href="/Login/PersonelG">
118                 <input class="btn btn-success w-100 text-bg-danger" type="button" value="Personel Girişi">
119             </a>
120             @if (ViewBag.Notification != null)
121             {
122                 <div class="bg-light text-danger">
123                     @ViewBag.Notification
124                 </div>
125             }
126         </form>
127     </div>
128
129 </body>
130 </html>
131

```

İlk bölümde bootstrap ve özel font kullandığım için onlara ait linkler bulunmaktadır. Ardından sayfamın görünüş olarak tasarımı için style tagı altında classlarımın css bilgileri bulunmaktadır. Body kısmında ise giriş sayfasının html kısmı bulunuyor. Bu sayfadaki çoğu divde formların daha güzel gözükmesi için bootstrap classları kullanılmıştır.

Cascading.cshtml: Bu sayfa hastane randevu al butonuna tıklandıktan sonra yönlendirilen sayfadır. Burada filtreleme yapılır.



Arama.cshtml: Bu sayfada yukardaki yapılan filtreleme işleminin ardından o filtreye uygun doktorların listelenmesi sağlanır.

Doktor.cshtml: Bu sayfa randevu alma sayfasıdır. Aynı zamanda AsiRandevu.cshtml ve AileHekimiRandevu.cshtml dosyalarıyla benzerdir.

Bu kısımda bu sayfaya ait cdnler ve css kısmı bulunmaktadır.



```

40 </style>
41 <body>
42 <a href="/Home/Index" class="fa-solid fa-chevron-left" style="margin-top: 50px;text-decoration:none; color:black;" /><a href="/Home/Index" style="text-decoration:none; font-size:24px;font-w
43
44 <div id="gunler" class="tab" style="margin-top: 50px">
45 <button id="DateLine Now.AddDays(1).ToString("dd MMM dddd")" class="tablinks DateLine Now.AddDays(1).ToString("dd MMM dddd")" onclick="openCity(event, 'Butonlar(1)')">DateLine Now Ad
46 <button id="DateLine Now.AddDays(2).ToString("dd MMM dddd")" class="tablinks DateLine Now.AddDays(2).ToString("dd MMM dddd")" onclick="openCity(event, 'Butonlar(1)')">DateLine Now Ad
47 <button id="DateLine Now.AddDays(3).ToString("dd MMM dddd")" class="tablinks DateLine Now.AddDays(3).ToString("dd MMM dddd")" onclick="openCity(event, 'Butonlar(2)')">DateLine Now Ad
48 <button id="DateLine Now.AddDays(4).ToString("dd MMM dddd")" class="tablinks DateLine Now.AddDays(4).ToString("dd MMM dddd")" onclick="openCity(event, 'Butonlar(2)')">DateLine Now Ad
49 <button id="DateLine Now.AddDays(5).ToString("dd MMM dddd")" class="tablinks DateLine Now.AddDays(5).ToString("dd MMM dddd")" onclick="openCity(event, 'Butonlar(3)')">DateLine Now Ad
50 <button id="DateLine Now.AddDays(6).ToString("dd MMM dddd")" class="tablinks DateLine Now.AddDays(6).ToString("dd MMM dddd")" onclick="openCity(event, 'Butonlar(3)')">DateLine Now Ad
51 </div>
52
53 <div id="Allbuttons" class="Allbuttons">
54 <for (int i = 0; i < 7; i++)>
55
56 <div id="Butonlar(1)" class="Butonlar(1)" tabcontent" style="border-style:solid; border-width:1px;border-radius:5px;margin-bottom:10px;width:auto;">
57 <div>
58 <div style="margin-left: 10px; margin-top: 10px; font-size: 10px;float:left;">9:00</div>
59 <button style="float: right; margin-top: 10px; font-size: 10px; margin-right: 10px;" onclick="slide(1).ToString();"><i class="fa-sharp fa-solid fa-chevron-down" /></button>
60 </div>
61
62 <div class="box at-5" style="border-bottom: 1px solid; ">
63 <div id="box(1).ToString()" class="box-3" style="display: none;">
64 <button type="button" id="9:00" style="width: 100px; background-color: blue;" class="btn btn-primary btn-m mx-5 buttons 9:00" 9:00</button>
65 <button type="button" id="9:20" style="width: 100px; background-color: blue;" class="btn btn-primary btn-m mx-5 buttons 9:20" 9:20</button>
66 <button type="button" id="9:40" style="width: 100px; background-color: blue;" class="btn btn-primary btn-m mx-5 buttons 9:40" 9:40</button>
67 </div>
68
69 <div>
70 <div style="margin-left: 10px; margin-top: 10px; font-size: 10px; float: left; font-weight: 500;">10:00</div>
71 <button style="float: right; margin-top: 10px; font-size: 10px;margin-right:10px;" onclick="sd(1).ToString();"><i class="fa-sharp fa-solid fa-chevron-down" /></button>
72 </div>
73
74 <div class="box at-5" style="border-bottom: 1px solid;">
75 <div id="aa(1).ToString()" class="aa-3" style="display: none;">
76 <button type="button" id="10:00" style="width: 100px; background-color: blue;" class="btn btn-primary btn-m mx-5 buttons 10:00" 10:00</button>
77 <button type="button" id="10:20" style="width: 100px; background-color: blue;" class="btn btn-primary btn-m mx-5 buttons 10:20" 10:20</button>
78 <button type="button" id="10:40" style="width: 100px; background-color: blue;" class="btn btn-primary btn-m mx-5 buttons 10:40" 10:40</button>
79 </div>
80
81 <div>
82 <div style="margin-left: 10px; margin-top: 10px; font-size: 10px; float: left;">11:00</div>
83 <button style="float: right; margin-top: 10px; font-size: 10px;margin-right:10px;" onclick="sa(1).ToString();"><i class="fa-sharp fa-solid fa-chevron-down" /></button>
84 </div>
85
86 <div class="box at-5" style="border-bottom: 1px solid;">
87 <div id="bb(1).ToString()" class="bb-3" style="display: none;">
88 <button type="button" id="11:00" style="width: 100px; background-color: blue;" class="btn btn-primary btn-m mx-5 buttons 11:00" 11:00</button>
89 </div>
90
91 </div>
92
93 <div class="box at-5" style="border-bottom: 1px solid;">
94 <div id="cc(1).ToString()" class="cc-3" style="display: none;">
95 <button type="button" id="11:00" style="width: 100px; background-color: blue;" class="btn btn-primary btn-m mx-5 buttons 11:00" 11:00</button>
96 </div>
97
98 </div>
99
100

```

Bu kısımda ise randevu alma ekranındaki günler ve saatler bulunmaktadır. Sayfanın devamında html kısmında geri kalan kısım saat butonları olduğu için doğrudan sayfa sonundaki script kısmına geçeceğim.

```

211
212 function openCity(evt, cityName) {
213     var i, tabcontent, tablinks;
214     tabcontent = document.getElementsByClassName("tabcontent");
215     for (i = 0; i < tabcontent.length; i++) {
216         tabcontent[i].style.display = "none";
217     }
218     tablinks = document.getElementsByClassName("tablinks");
219     for (i = 0; i < tablinks.length; i++) {
220         tablinks[i].className = tablinks[i].className.replace(" active", "");
221     }
222     document.getElementById(cityName).style.display = "block";
223     evt.currentTarget.className += " active";
224 }
225
226 function slide(i) {
227     console.log(i);
228     var x = document.getElementById("box(" + i + ")");
229     if (x.style.display === "none") {
230         x.style.display = "block";
231     } else {
232         x.style.display = "none";
233     }
234 }
235
236 function sd(i) {
237     var x = document.getElementById("aa(" + i + ")");
238     if (x.style.display === "none") {
239         x.style.display = "block";
240     } else {
241         x.style.display = "none";
242     }
243 }
244
245 function sa(i) {
246     var x = document.getElementById("bb(" + i + ")");
247     if (x.style.display === "none") {
248         x.style.display = "block";
249     } else {
250         x.style.display = "none";
251     }
252 }
253
254 function my(i) {
255     var x = document.getElementById("cc(" + i + ")");
256     if (x.style.display === "none") {
257         x.style.display = "block";
258     } else {
259         x.style.display = "none";
260     }
261 }
262
263

```

Bu kısımdaki ilk fonksiyonum tab panel içindir. Yani randevu alma sayfamda günlerin üstüne tıklanınca alt tarafına tüm saat butonlarının bulunduğu kısmın gelmesini sağlar. Altteki fonksiyonlar ise saatin yanında bulunan slide up/down butonunun işlevli hale gelmesini sağlıyor. Yani aşağı kayarak o saate ait 3 tane randevu dakikasının gösterilmesini sağlıyor.

```
290 }
291 let selected = "";
292 document.getElementById('gunler').addEventListener('click', function (event) {
293
294     const collection = document.getElementsByClassName('tablinks');
295
296     for (let i = 0; i < collection.length; i++) {
297
298         if (event.target.classList[i] == collection[i].classList[1]) {
299             selected = event.target.classList[1] + " " + event.target.classList[2] + " " + event.target.classList[3];
300         }
301         else {
302
303         }
304     }
305     console.log(selected);
306 });
307
308 let selectedhour = "";
309 document.getElementById('Allbuttons').addEventListener('click', function (event) {
310
311     const collection = document.getElementsByClassName('butons');
312     for (let i = 0; i < collection.length; i++) {
313         if (event.target.classList[0] != "fa-sharp") {
314             collection[i].style.backgroundColor = "blue";
315         }
316     }
317     console.log(event.target.classList[0]);
318     for (let i = 0; i < collection.length; i++) {
319
320         if (event.target.classList[5] == collection[i].classList[5]) {
321
322             if (collection[i].style.backgroundColor != "rgb(0, 171, 182)") {
323                 if (collection[i].style.backgroundColor == "blue") {
324                     collection[i].style.backgroundColor = "green";
325                     selectedhour = collection[i].classList[5];
326                     console.log(selectedhour);
327                 }
328                 else {
329                     collection[i].style.backgroundColor = "blue";
330                 }
331             }
332         }
333         else {
334
335         }
336     }
337 }
338 }
339 }
340 }
```

Bu iki fonksiyon ise gün ve saat butonlarına basıldığında onları dinlememi(okumamı) sağlıyor. Dinlediğim bilgileri ise bir değişkene atarak orda tutuyorum. Ayrıca altteki fonksiyon saat butonlarına tıklandığında renginin değişmesini sağlamaktadır.



```

181 </script>
182
183 $("#buyButton").click(function (event) {
184     console.log("aaa");
185     var url = window.location.pathname;
186     var sid = url.substring(url.lastIndexOf('/') + 1);
187     var ticket = {
188         "DOKTORID": sid,
189         "RANDEVUTUR": 3,
190         "RANDEVUTARIH": selected,
191         "RANDEVUSAAT": selectedhour,
192     }
193     event.preventDefault();
194
195     $.ajax({
196         type: "POST",
197         dataType: "json",
198         url: "/Home/DoktorRandevuAl",
199         data: {
200             data: JSON.stringify(
201                 ticket)
202         },
203         success: function (response) {
204             window.location.href = "/Home/Index";
205         }
206     });
207
208 });
209
210

```

Burda ajax kullanarak kullanıcının randevu alma ekranında tıkladığı gün ve saat bilgilerinin randevu alma controllerına atılması sağlanıyor.

DoktorAdmin-Index.cshtml: Bu sayfada admin panelinde bulunan doktor bilgisi ekranının görüntülenmesi sağlanır. Adminnp-Index ve KlinikAdmin-Index sayfaları da bu sayfayla benzerdir.

[illegible]

Bu sayfada doktorların adıyla arama yapabilmek için arama kutusu ve doktor sayısı fazla olduğu için sayfalama(pagedlist) kullanılmıştır.