|  |  |  |
| --- | --- | --- |
|  | **T.C.**  **KASTAMONU ÜNİVERSİTESİ**  **MÜHENDİSLİK VE MİMARLIK FAKÜLTESİ**  **BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ** |  |

**BİTİRME PROJESİ I**

**KONU**

Hastane Otomasyonu

**DANIŞMAN**

Doç. Dr. Kemal AKYOL

Nisan- 2022

**KASTAMONU**

**T.C.**

**KASTAMONU ÜNİVERSİTESİ**

**MÜHENDİSLİK VE MİMARLIK FAKÜLTESİ**

**BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**

**BİTİRME PROJESİ I**

**KONU**

Hastane Otomasyonu

**HAZIRLAYAN**

194410006 – Eren YARDIM

**DANIŞMAN**

Doç. Dr. Kemal AKYOL

Nisan- 2022

**KASTAMONU**

**23/04/ 2022**

**ETİK BEYAN**

Kastamonu Üniversitesi, Mühendislik ve Mimarlık Fakültesi, Bilgisayar Mühendisliği Bölümü, Mühendislik Tamamlama Programı, Tez Hazırlama Kılavuzu’nda yer alan kurallara uygun olarak hazırladığım bu çalışmada; proje içinde sunduğum verileri, bilgileri ve dokümanları akademik ve etik kurallar çerçevesinde elde ettiğimi, tüm bilgi, belge, değerlendirme ve sonuçları bilimsel etik ve ahlak kurallarına uygun olarak sunduğumu, proje çalışmasında yararlandığım eserlerin tümüne uygun atıfta bulunarak kaynak gösterdiğimi, kullanılan verilerde herhangi bir değişiklik yapmadığımı, bu projede sunduğum çalışmanın özgün olduğunu, bildirir; aksi bir durumda aleyhime doğabilecek tüm hak kayıplarını kabullendiğimi beyan ederim.

**Öğrenci Numarası :** 194410006 **İmza**

**Adı Soyadı :** Eren YARDIM

**ÖZET**

**Bitime Projesi**

**HASTANE OTOMASYONU**

**Eren YARDIM**

**Kastamonu Üniversitesi**

**Bilgisayar Mühendisliği Bölümü**

**Bitirme Projesi Danışmanı:**

**Doç. Dr. Kemal AKYOL**

**Nisan 2022, 27 sayfa**

Bu bitirme projesinde ilk olarak programlama dilleri hakkında bilgiler verilmiştir. Daha sonra nesneye yönelik programlama dillerinden olan C# dili üzerinde araştırmalar yapılmıştır. C# programlama dili derleyicisi olarak kullanılan Visual Studio 2019 programı incelenmiş ve Visual Studio 2019 ilk defa kullanacaklara yönelik programın arayüzüme aşina olabilmeleri için arayüz bileşenleri verilmiştir. Bunlarla birlikte bitirme projemin konusunu oluşturan otomasyon sistemi için kullanılmak üzere veri tabanları incelenmiştir. Projemizde kullanılan MICROSOFT SQL Server ve MY SQL hakkında bilgiler işlenmiştir. Projem herhangi bir hastane servisinin gereksinim duyacağı, hastanenin ihtiyaçlarına cevap verebilecek bir otomasyon sistemi ve bu sistemle Web Service üzerinden haberleşen Eczane Otomasyonu içermektedir. Yazılım geliştirme süreç aşamaları işlenmiş ve proje görselleriyle birlikte tanıtılmıştır.

**Anahtar Sözcükler :** C#, MICROSOFT SQL Server, Visual Studio 2019, MVC, Microsoft SQL Server Management Studio, MY SQL, WEB Service

**ABSTRACT**

**Senior Project**

**HOSPİTAL AUTOMATION**

**Eren YARDIM**

**Kastamonu University**

**Faculty of Engineering and Architecture**

**Department of Computer Engineering**

**Project Advisor:**

**Assoc. Dr. Kemal AKYOL**

**April 2022, 27 pages**

In this graduation project, firstly, information about programming languages is given. Later, researches were made on C# language, which is one of the object-oriented programming languages. The Visual Studio 2019 program, which is used as a C# programming language compiler, has been examined and the interface components have been given for those who will use Visual Studio 2019 for the first time so that they can be familiar with the interface of the program. Along with these, databases were examined to be used for the automation system, which is the subject of my graduation project. Information about MICROSOFT SQL Server and MY SQL used in our project has been processed. My project includes an automation system that can meet the needs of any hospital service and Pharmacy Automation that communicates with this system via Web Service. The software development process stages were processed and introduced with the project visuals.

**Key Words :** C#, MS SQL, Visual Studio 2019, MVC, MICROSOFT SQL Server Management Studio, MY SQL, WEB Service

**İÇİNDEKİLER SAYFA**

[**BİTİRME PROJESİ I** i](#_Toc105664591)

[**DANIŞMAN** i](#_Toc105664592)

[**BİTİRME PROJESİ I** ii](#_Toc105664593)

[**HAZIRLAYAN** ii](#_Toc105664594)

[**DANIŞMAN** ii](#_Toc105664595)

[**ÖZET** iv](#_Toc105664596)

[**ABSTRACT** v](#_Toc105664597)

[**Eren YARDIM** v](#_Toc105664598)

[**Kastamonu University** v](#_Toc105664599)

[**Project Advisor:** v](#_Toc105664600)

[1. GİRİŞ ve TANITIM 8](#_Toc105664601)

[2. PROJE KONUSU HAKKINDA GENEL BİLGİLER 8](#_Toc105664602)

[3. PROJE ÇALIŞMASINDA KULLANILAN MATERYALLER 9](#_Toc105664603)

[3.1. C# Programlama Dili 9](#_Toc105664604)

[3.2. MVC 10](#_Toc105664605)

[**3.2.1.** **Model Nedir?** 11](#_Toc105664606)

[**3.2.2. View Nedir?** 12](#_Toc105664607)

[**3.2.3. Controller Nedir?** 12](#_Toc105664608)

[**3.2.4. MVC’nin Yaşam Döngüsü?** 12](#_Toc105664609)

[3.2. Visual Studio 2019 13](#_Toc105664610)

[3.3 MICROSOFT SQL Server 14](#_Toc105664611)

[3.4 MY SQL 15](#_Toc105664612)

[**3.4.1 MySQL Nerede Kullanılır** 16](#_Toc105664613)

[**3.4.2MySQL WorkBench** 16](#_Toc105664614)

[3.5. Web Service 17](#_Toc105664615)

[4. Hastane Otomasyonu Proje Görünüm ve Açıklamaları 18](#_Toc105664616)

[**4.1. Hastane Veri Tabanı** 18](#_Toc105664617)

[**4.2. Anasayfa Ekranı** 21](#_Toc105664618)

[**4.3. Login Ekranı** 21](#_Toc105664619)

[**4.4. Randevu Sistemi** 24](#_Toc105664620)

[**4.5. Hasta Muayene İşlemi** 27](#_Toc105664621)

[**4.6. Hasta Muayene Geçmişi** 30](#_Toc105664622)

[**4.7. Doktor İzin Sistemi** 31](#_Toc105664623)

[**4.8. Doktorlar Arası Mesajlaşma** 32](#_Toc105664624)

[**4.9. Personel Ekleme ve Listeleme** 34](#_Toc105664625)

[**4.10. Doktor İzin Onay** 36](#_Toc105664626)

[**4.11. Doktorlar ve Personeller** 38](#_Toc105664627)

[**4.12. Personel Ekle** 39](#_Toc105664628)

[**4.13. Hastane CRUD işlemleri** 41](#_Toc105664629)

[**4.14. Klinik CRUD İşlemleri** 45](#_Toc105664630)

[**4.15. İlaç CRUD İşlemleri** 48](#_Toc105664631)

[5. Eczane Otomasyonu Görünüm ve Açıklamaları 50](#_Toc105664632)

[**5.1. Eczane Veri Tabanı** 50](#_Toc105664633)

[**5.2. Reçeteli İlaç Ver** 53](#_Toc105664634)

[**5.3. Faturalar** 56](#_Toc105664635)

[**5.4. Personel CRUD İşlemleri** 57](#_Toc105664636)

[**5.6. İlaç CRUD İşlemleri** 61](#_Toc105664637)

[6. Projeyi Somee Sunucusuna Yükleme 64](#_Toc105664638)

[7. SONUÇLAR VE ÖNERİLER 68](#_Toc105664639)

[**KAYNAKLAR** 69](#_Toc105664640)

[**ÖZGEÇMİŞ** 70](#_Toc105664641)

[**ADRES BİLGİLERİ** 70](#_Toc105664642)

# 1. GİRİŞ ve TANITIM

Hastane otomasyon programları, sağlık merkezinize başvuran hastanın detaylı kişisel iletişim ve medikal geçmiş bilgilerini kayıt altına alıp takip etmekle kalmaz aynı zamanda cari hesap oluşturmanızı, ön muhasebe, maliyet yönetimi ve [personel yönetimi](https://www.mikro.com.tr/yazilim-programlari/personel-mesai-takip-programi) yapmanızı, kolayca randevu oluşturmanızı, hasta kabul, yatış ve çıkışlarını düzenlemenizi, anlaşmalı kurum takibi yapmanızı, doktor hak edişleri hesaplamanızı ve sağlık merkezinizin sağlıklı şekilde çalışmasını sağlayan daha pek çok işlemi gerçekleştirmenizi sağlar.

Bütünleşik bir yapıda çalışan hastane otomasyon yazılımları, bir birime yapılan veri girişini aynı anda diğer birimlere de yansıtır. Böylece hem bilgilerin tekrar tekrar girilmesine gerek kalmaz hem de hata oranı sıfırlanır. Pratik ve kolay kullanım sunan bu programlar, dilediğiniz ayrıntıda raporlar sunarak kapsamlı maliyet ve gelir analizi yapmanıza olanak tanır. Böylece merkezinizin geleceğini ilgilendiren önemli kararları alırken elinizde daha sağlam veriler olur.

Eczane otomasyonu, muayene edilen hastaya verilen reçete kodu ile reçetede bulunan ilaç barkodlarıyla örtüşen ilaçların belirlenmesi ve satılması işlemlerini yürütür. İlaç stok takibi ve Personel CRUD işlemleri gerçekleştirmektedir.

# 2. PROJE KONUSU HAKKINDA GENEL BİLGİLER

Hastane otomasyonu, genel olarak sistem üzerinden hasta randevu alma işlemi ve doktor muayene işlemi gibi faaliyetler gerçekleştirmeyi hedefler. Sistem üzerinde hastane yönetimi tarafından personel ekleme, silme ve izin kontrolü faaliyetleri gerçekleştirilecek. Doktor muayenesi sonucunda hastaneden hastaya verilmiş olan ilaçların temin edilmesi için Eczane otomasyonu da yapılacak. Hastane ve eczane otomasyonu arasında veri alışverişi Web Service üzerinden gerçekleştirilecek.

# 3. PROJE ÇALIŞMASINDA KULLANILAN MATERYALLER

## 3.1. C# Programlama Dili

C# Programlama Dili Microsoft'un geliştirmiş olduğu yeni nesil dilidir. Yine Microsoft tarafından geliştirilmiş .NET Teknolojisi için geliştirilmiş dillerden biridir.

C programlama dilinde bir tamsayı değişkeni 1 attırmak için ++ soneki kullanılır. C++ dili adını, C diliyle Nesneye Yönelimli Programlama yapabilmek için eklentiler (C With Classes) almıştır. Benzer şekilde C++ diline yeni eklentiler yapılarak ((C++)++) bir adım daha ileriye götürülmüş ve tamamen nesneye yönelik tasarlanmış C# dilinin isimlendirilmesinde, + karakterlerinin birbirlerine yakınlaşmış hali ve bir melodi anahtarı olan C# Major kullanılmıştır.

Bu dilin tasarlanmasına Pascal, Delphi derleyicileri ve J++ programlama dilinin tasarımlarıyla bilinen Anders Hejlsberg liderlik etmiştir. Birçok alanda Java'yı kendisine örnek alır ve C# da java gibi C ve C++ kod sözdizimine benzer bir kod yapısındadır. .NET kütüphanelerini kullanmak amacıyla yazılan programların çalıştığı bilgisayarlarda uyumlu bir kütüphanenin ve yorumlayıcının bulunması gereklidir. Bu, Microsoft'un .Net Framewok'u olabileceği gibi ECMA standartlarına uygun herhangi bir kütüphane ve yorumlayıcı da olabilir.

Özellikle nesne yönelimli programlama kavramının gelişmesine katkıda bulunan en aktif programlama dillerinden biridir .NET platformunun anadili olduğu bazı kesimler tarafından kabul görse de bazıları bunun doğru olmadığını savunur. C#, .NET orta seviyeli programlama dillerindendir. Yani hem makine diline hem de insan algısına eşit seviyededir. Buradaki orta ifadesi dilin gücünü değil makine dili ile günlük konuşma diline olan mesafesini göstermektedir. Örneğin; Visual Basic .NET (VB.NET) yüksek seviyeli bir dildir. Dersek bu dilin insanların günlük yaşantılarında konuşma biçimine yakın şekilde yazıldığını ifade etmektedir.

##### **3.1.1. Razor Yapısı**

Razor View Engine; View’lar da HTML, CSS, JavaScript ile beraber C#’ın da kullanılabilmesini sağlamaktadır. View’lara yazılan HTML, CSS ve JavaScript kodları istemci tarafında çalışmaktadır. C# kodları ise sunucu tarafında çalışır ve sonuç istemciye gönderilir.

View’de C# kodları yazabilmek için “***@{ }***” ile Razor kod bloğu oluşturmamız gerekmektedir. Örneğimiz de Razor kod bloğu içerisinde 3 adet değişken tanımladık. Razor kod bloğu içerisinde tanımladığımız değişkenlere, kod bloğu dışında erişebilmek için başlarına “***@***” işareti koymamız gerekmektedir.

@{

string Ad = “Eren”;

string Soyadı = “Yardım”;

string Mail = “erenym3@gmail.com”;

}

<b>Adı:</b>@Ad <br/>

<b>Soyadı:</b>@Soyadı <br/>

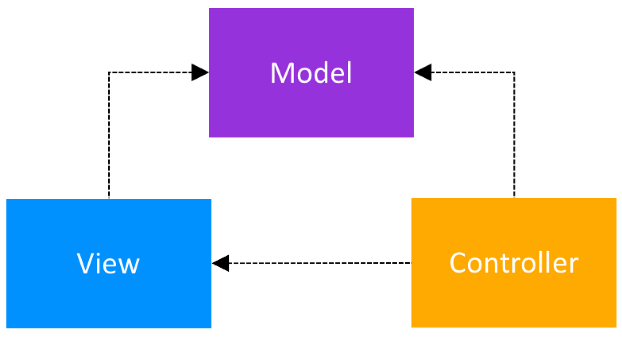
<b>Mail Adresi:</b>@Mail <br/>s

## 3.2. MVC

MVC, Yazılım Mühendisliği’nde önemli bir yere sahip architectural patterns (yazılım mimari desenleri)’ın bir parçasıdır. Model, View ve Controller kelimelerinin baş harflerinden oluşan MVC (Model-View-Controller), 1979 yılında Tygve Reeskaug tarafından oluşturulmuş ve yazılım gelişmede birçok projede kullanılmıştır. Son dönemlerde Microsoft’un MVC desenini Asp.Net teknolojisi ile birleştirmesi ile popülaritesi daha da artmıştır.

MVC ile ilgili en yanlış bilgi, MVC’nin Microsoft tarafından çıkartıldığı düşüncesidir. Yukarıdaki paragrafta da bahsedildiği üzere, MVC’nin Asp.Net’e entegre edilmesinden önce bu deseni birçok (.Net) yazılım geliştiricisi bilmemekteydi, bilse de kullanmıyorlardı. Asp.Net MVC’nin gelişiminin ardından MVC’ye ilgi oldukça artmıştır.

MVC deseni, 3 katmandan oluşmaktadır ve katmanları birbirinden bağımsız (birbirini etkilemeden) olarak çalışmaktadır. Bu sebeple çoğunlukla büyük çaplı projelerde projelerin yönetiminin ve kontrolünün daha rahat sağlanabilmesi için tercih edilmektedir. MVC ile geliştirilen projelerde projenin detaylarına göre birçok kişi eş zamanlı olarak kolaylıkla çalışabilmektedir.



Şekil 3.1. MVC

### **3.2.1.** **Model Nedir?**

Model, MVC’de projenin iş mantığının (business logic) oluşturulduğu bölümdür. İş mantığıyla beraber doğrulama (validation) ve veri erişim (data access) işlemleri de bu bölümde gerçekleştirilmektedir.

Model tek katmandan oluşabileceği gibi kendi içinde birden fazla katmandan da oluşabilir. İç yapılandırma projenin büyüklüğü ile yazılım geliştiricinin planlamasına kalmış bir durumdur. Eğer proje büyük çaplı ise modeli birden çok katmana ayırmak projenin yönetimi açısından faydalı olacaktır.

### **3.2.2. View Nedir?**

View, MVC’de projenin arayüzlerinin oluşturulduğu bölümdür. Bu bölümde projenin kullanıcılara sunulacak olan HTML dosyaları yer almaktadır. Projenin geliştirildiği yazılım dillerine göre dosya uzantıları da değişebilmektedir. Projelerin büyüklüğüne göre dikkat edilmesi gereken bir nokta ise, klasörlemedir.

Eğer bir web projesi geliştiriyorsanız, projenin View’larının yer aldığı klasörlerinin hiyerarşisi, ilerleyen dönemlerde karmaşıklığa sebep olmaması için dikkatli yapılmalıdır. Kimi yazılım geliştiriciler web projelerinde HTML dosyaları ile Javascript, CSS ve resim dosyalarını aynı klasör içinde barındırmaktadır. Ufak bir ayrıntı gibi görünse de projenin ilerleyen dönemlerinde ciddi problemler oluşturmaktadır. View’ın bir görevi de kullanıcılardan alınan istekleri controller’a iletmektir.

### **3.2.3. Controller Nedir?**

Controller, MVC’de projenin iç süreçlerini kontrol eden bölümdür. Bu bölümde View ile Model arasındaki bağlantı kurulur. Kullanıcılardan gelen istekler (request) Controller’larda değerlendirilir, isteğin detayına göre hangi işlemlerin yapılacağı ve kullanıcıya hangi View’ın döneceği (response) belirtilir.

### **3.2.4. MVC’nin Yaşam Döngüsü?**

MVC’nin parçaları olan Model, View ve Controller’ın ne olduğu yukarıdaki bölümde anlatıldı. Şimdi bu bilgileri toparlayıp MVC’nin yaşam döngüsünü (çalışma prensibini) detaylıca inceleyelim.



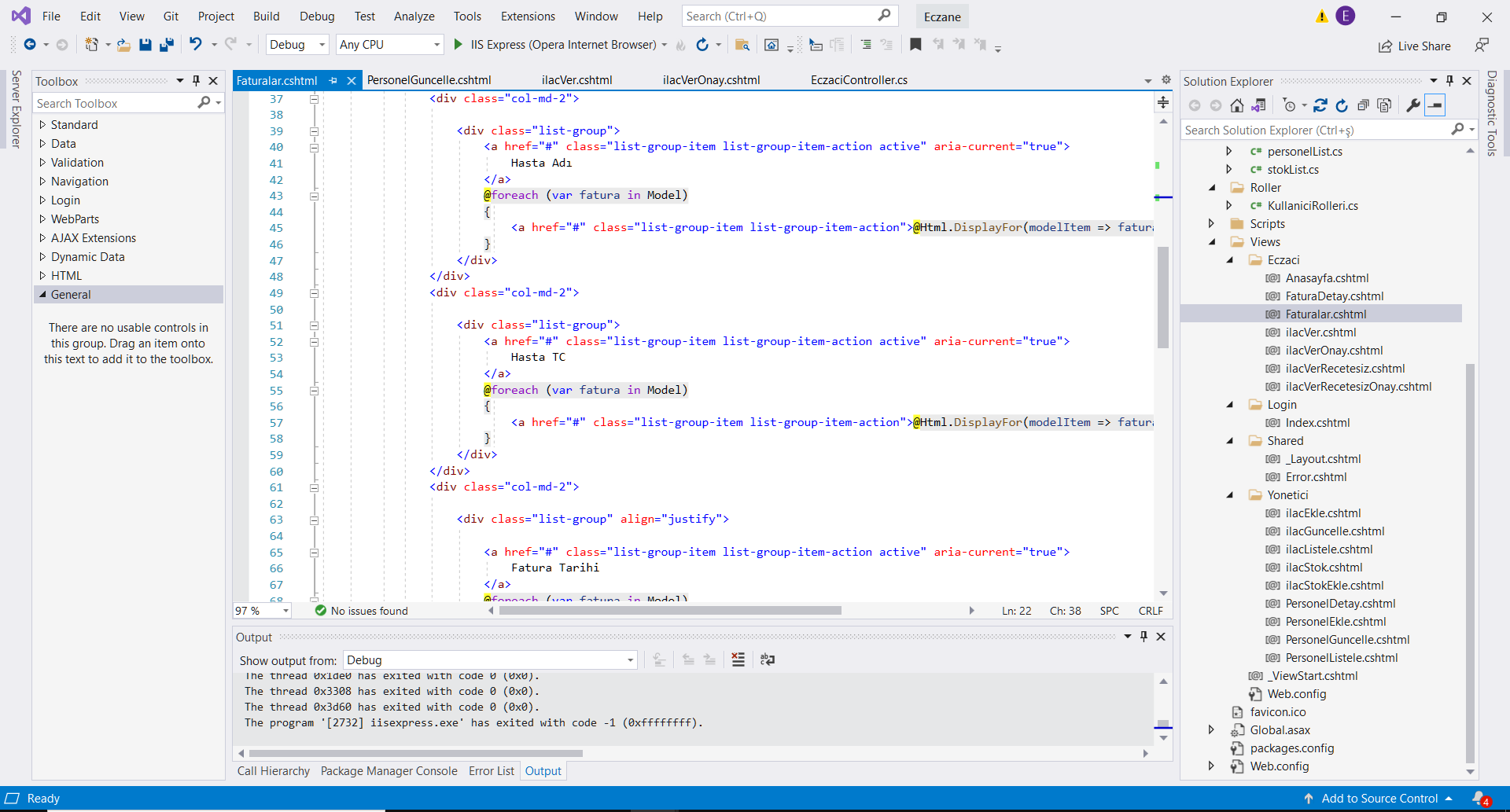
Şekil 3.2. MVC Yaşam Döngüsü

**1.** **HTTP Request:** Sizin her ASP.NET MVC uygulamasını görüntülemek istemeniz bir request(istek) tir. Bu istediğinizi HTTP üzerinden IIS tarafından alınır. Her yaptığınız istek Server tarafından bir yanıtla  
son bulması gerekir.  
**2. Routing:**ASP.NET MVC uygulamasını her istek yaptığınızda, yaptığınız yanıt UrlRoutingModule HTTP Module tarafından durdurulur. UrlRoutingModule bir isteği durdurduğu zaman, gelen istek RouteTable’dan hangi Controller tarafından üstleneceğine karar verilir.  
**3. Controller:** RouteTable’dan gelen route bilgisine göre Controller hangi Action’ı çalıştıracaksa o View çalıştırılır. View, Controller tarafından render edilmez. Controller tarafından geriye ViewResult döndürülür.  
**4. ViewResult:**ViewResult, View’i render etmek için aktif View Engine’i çağırır.  
**5. ViewEngine :**Bir CSHTML dosyayı oluşturduğunuzda içerisindeki script ve markuplar, Razor View Engin tarafından bazı ASP.NET API’lerini sayfalarınızı HTML’e çevirmek için kullanır.  
**6. View:**View Engine tarafından HTML’e çevirilen kodlar kullanıcıya sunulur.  
**7. Response:**HTTP üzerinden View kullanıcıya gösterilir.

## 3.2. Visual Studio 2019

Microsoft Visual Studio, Windows için IDE yani entegre bir geliştirme ortamıdır. Visual Studio, Microsoft Windows, Windows Mobil, Windows CE, .NET Framework, .NET Compact Framework ve Microsoft Silverlight tarafından desteklenen tüm platformlar için yönetilen kod ile birlikte yerel kod ve web siteleri, web uygulamaları ve web hizmetleriyle ilgili kodlar yazmak için kullanışlı bir programdır.

Bu programı bu kadar özel kılan ve kullanıcılar için tercih sebebi olma nedeniyse kullanıcılarına kod düzenleyici, hata ayıklayıcı, GUI tasarım aracı ve veri tabanı şeması tasarımı seçeneği, revizyon kontrol sistemi gibi çeşitli özellikler sunmasından kaynaklıdır.



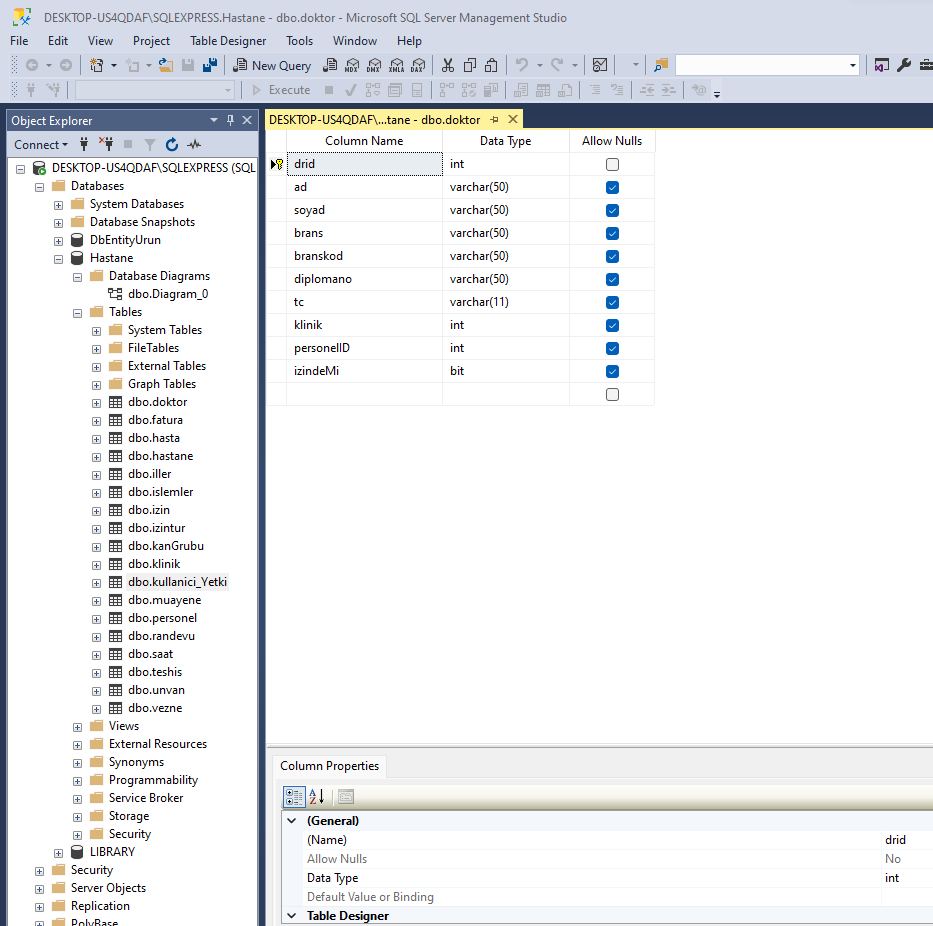
Şekil 3.3. Visual Studio 2019

## 3.3 MICROSOFT SQL Server

Microsoft SQL Server, verilerin güvenle ve bütünlük içerisinde depolanmasını ve aynı anda birden fazla kullanıcı tarafından erişilmesini sağlayan kurumsal çaplı bir ilişkisel veri tabanı yönetim sistemidir (RDBMS). Birbiriyle ilişkili verilerin sistematik bir şekilde kaydedilmesini ve bu verilerden beslenen uygulamalar tarafından ihtiyaç anında kullanılmasına olanak sağlayarak işletmelerin BT altyapılarındaki veri aktarımını sorunsuz bir şekilde yönetmelerini sağlar.

SQL Server, işletmeler için kritik önem taşıyan verileri şifreleme, verilere erişim sağlayan kişileri gözlemleme ve erişim kısıtlamaları tanımlama gibi güvenlik özellikleri sayesinde kullanıcılara kapsamlı bir denetim kapasitesi sunar. Yüksek Kullanılabilirlik (High Availability) ve olağanüstü durum kurtarma özelliklerini tek bir çözümde birleştiren SQL Server, hızlı yük devretme ve yük dengeleme desteği ile aksaklık süresini en aza indirir ve güvenli veri akışının optimize edilmesini sağlar.

Bilişim, sağlık ve finans gibi çeşitli sektörlerde önde gelen şirketler tarafından tercih edilen Microsoft SQL Server, her tip kullanıcının ihtiyaçlarına uyum sağlayabilen ölçeklenebilen bir yapıya sahiptir.



Şekil 3.4. MS SQL Server Management Studio

## 3.4 MY SQL

**MySQL**, modern büyük veri ekosistemindeki en tanınmış teknolojilerden biridir. MySQL ile, ilişkisel sistemlerde yeni olanlar bile hızlı, güçlü ve güvenli veri depolama sistemleri oluşturabilir. MySQL ’in programatik sözdizimi ve ara yüzleri, diğer popüler sorgu dillerinin ve yapılandırılmış veri depolarının geniş dünyasına açılan mükemmel bir ağ geçitidir.

MySQL, Oracle tarafından geliştirilmiş, yapılandırılmış sorgu diline (SQL) dayalı bir ilişkisel veri tabanı yönetim sistemidir (RDBMS).

Açık kaynak yapısı, istikrarı ve Oracle’nin sürekli geliştirme desteğiyle zengin bir özellik setine sahiptir. Facebook, Flickr, Twitter, Wikipedia ve YouTube gibi internet açısından kritik kuruluşların hepsi MySQL kullanır.

Verileri bir veri tabanında depolamak için kullanabileceğiniz farklı yaklaşımlar vardır. MySQL, ilişkisel veri tabanı adı verilen bir yaklaşımı tercih eder.

İlişkisel bir veri tabanıyla, verileriniz, her şeyi tek bir büyük depolama biriminde bir araya getirmek yerine, tablo adı verilen çok sayıda ayrı depolama alanına böler.

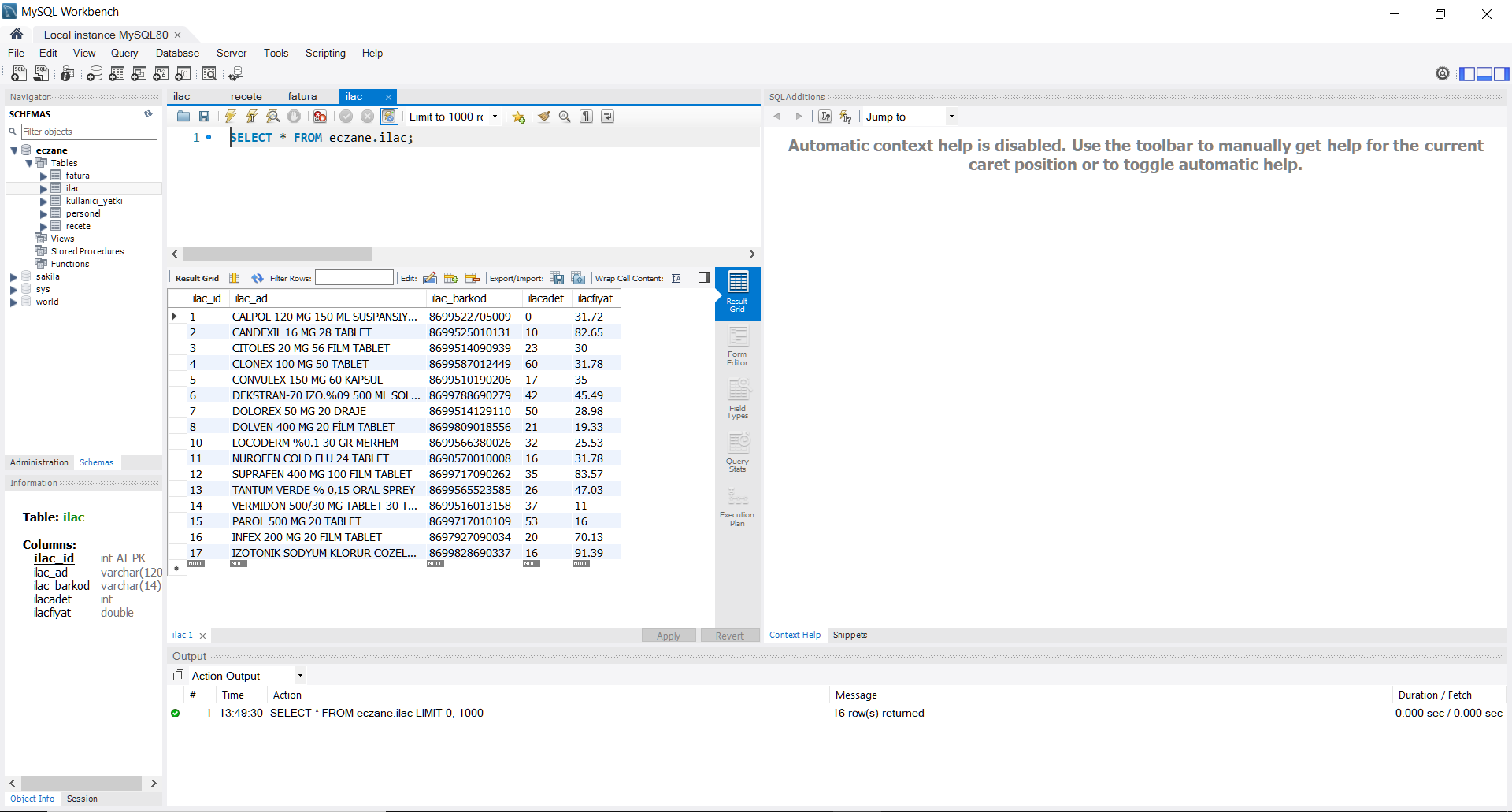
### **3.4.1 MySQL Nerede Kullanılır**

**MySQL** için en yaygın kullanım, bir web veri tabanı içindir. Tek bir bilgi kaydından çevrimiçi mağaza için mevcut ürünlerin tüm envanterine kadar her şeyi depolamak için kullanılabilir.

PHP veya Perl gibi bir betik diliyle ilişkili olarak, kategorize edilmiş ve aranabilir bilgileri bir web sitesi kullanıcısına hızlı bir şekilde göstermek için bir mySQL veri tabanı ile gerçek zamanlı olarak etkileşime girecek web siteleri oluşturmak mümkündür.

### **3.4.2MySQL WorkBench**

MySQL Workbench, veri tabanı mimarları, geliştiricileri ve DBA’lar için birleşik bir görsel araçtır. Veri tabanlarını tasarlamak, geliştirmek ve yönetmek için çeşitli işlevler sağlar. 3 sürümü mevcuttur: Community, Standard ve Enterprise. Community, temel bir dizi özellik sunan ücretsiz bir açık kaynak sürümüdür, geri kalan ikisi genişletilmiş işlevselliğe sahip ticari sürümlerdir.



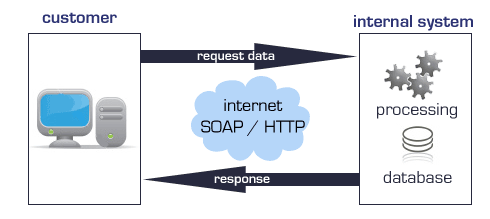
Şekil 3.5. My SQL WorkBench

## 3.5. Web Service

Genel olarak web servisler SOAP (Simple Object Access Protocol/Basit Nesne Erişim Protokolü) XML yapısını kullanarak çalışmaktadır. Bu yapının bizlere sağlamış olduğu avantaj platformdan bağımsız olarak istenilen yerden erişilmesini sağlamaktır.

Örneğin; .Net ile yazılan bir web servisi Javascript veya JSON ile ve birçok yazılım diliyle kullanabilmekteyiz. Bunun sebebi web servisten istediğimiz sonucun bizlere XML olarak sonuçları getirmesidir.

Web servisler birçok alanda kullanılmaktadır. Örneğin; TCMB döviz kurlarını almak istersek kurumun vermiş olduğu web servisten istediğimiz veriyi alıp işlemlerimizi yapabiliriz. Web servis kullanırken dikkat etmemiz gereken bir husus ise web servis yapısına uygun değerleri göndermemiz olmalıdır.



Şekil 3.6. Web Service

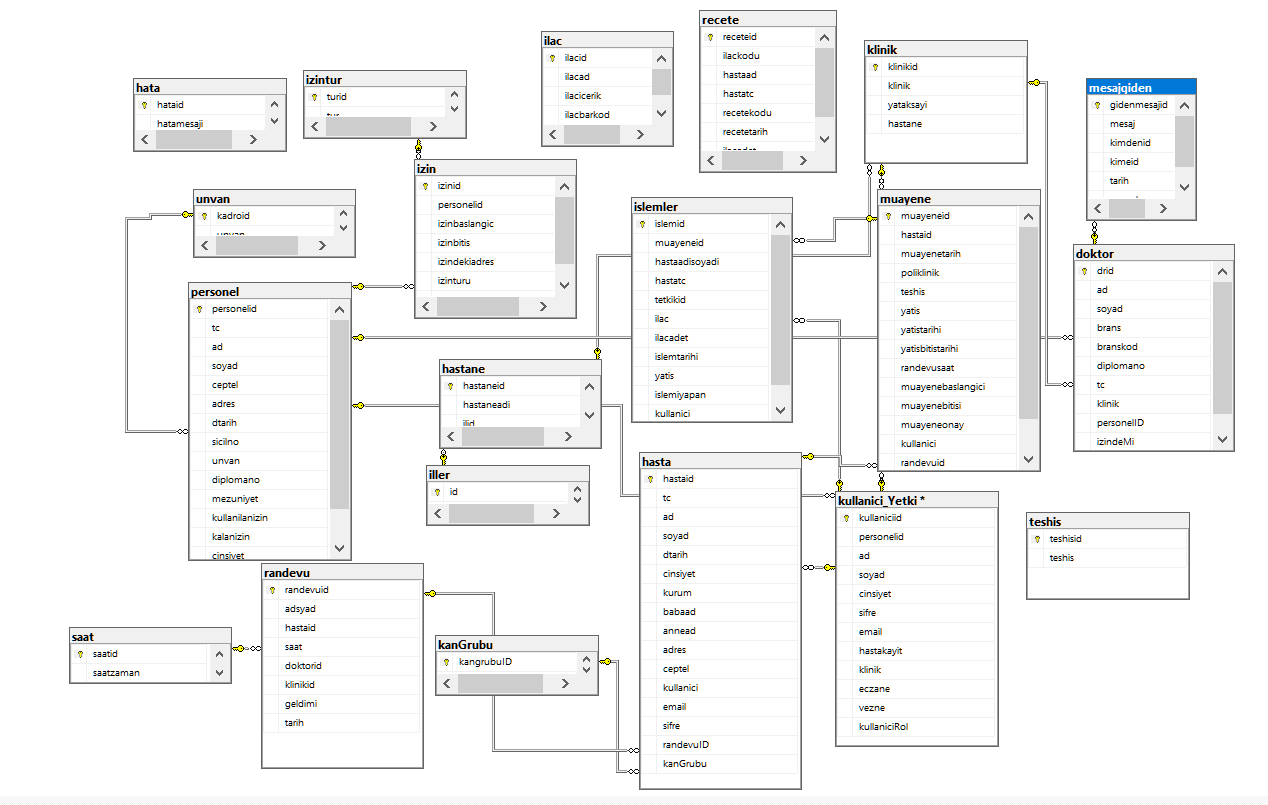
# 4. Hastane Otomasyonu Proje Görünüm ve Açıklamaları

### **4.1. Hastane Veri Tabanı**

MS SQL’de oluşturduğum veri tabanını projeye ADO.NET Entity Data Model üzerinden EF Designer from database yaklaşımı ile dahil ettim. Modele dahil edilecek veritabanı bağlantısını, model için ayarları ve veritabanı nesnelerini seçtikten sonra uygulamamızın etkileşimde bulunacağı sınıflar modelden oluşturulur.



Şekil 4.1. MS SQL Bağlantısı.



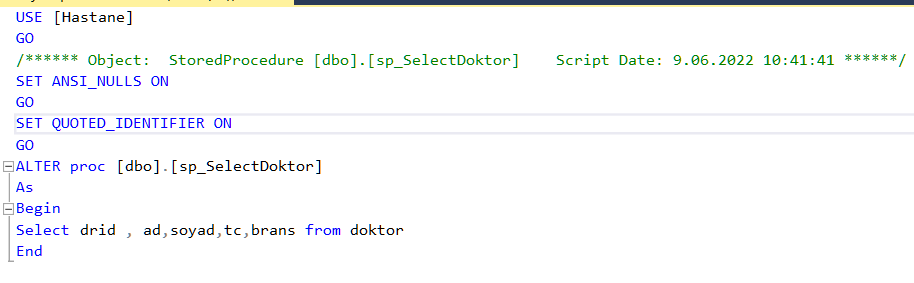
Şekil 4.2. MS SQL Tablolar Arası İlişki

**Tablolar arası ilişkiler:**

**Bağlantılı Tablolar Primary Key Foreing Key**

1. randevu-saat saat(saatid) randevu(saat)
2. unvan-personel unvan(kadroid) personel(kadroid)
3. izintur-izin izintur(turid) izin(izinturu)
4. izin-personel personel(personelid) izin(personelid)
5. iller-hastane iller(id) hastane(ilid)
6. hastane-klinik hastane(hastaneid) klinik(hastane)
7. klinik-doktor klinik(klinikid) doktor(klinik)
8. klinik-muayene klinik(klinikid) muayene(poliklinik)
9. doktor-mesajgiden doktor(drid) mesajgiden(kimeid)
10. randevu-hasta randevu(randevuid) hasta(randevuID)
11. kanGrubu-hasta kanGrubu(kangrubuID) hasta(kanGrubu)
12. personel-doktor personel(personelid) doktor(personelID)
13. personel-kullanici\_Yetki personel(personelid) kullanici\_Yetki(personelid)
14. kullanici\_Yetki-hasta kullanici\_Yetki(kullaniciid) hasta(kullanici)
15. kullanici\_Yetki-islemler kullanici\_Yetki(kullaniciid) islemler(kullanici)
16. kullanici\_Yetki-muayene kullanici\_Yetki(kullaniciid) muayene(kullanici)
17. hasta-muayene hasta(hastaid) muayene(hastaid)
18. muayene-islemler muayene(muayeneid) işlemler(muayeneid)

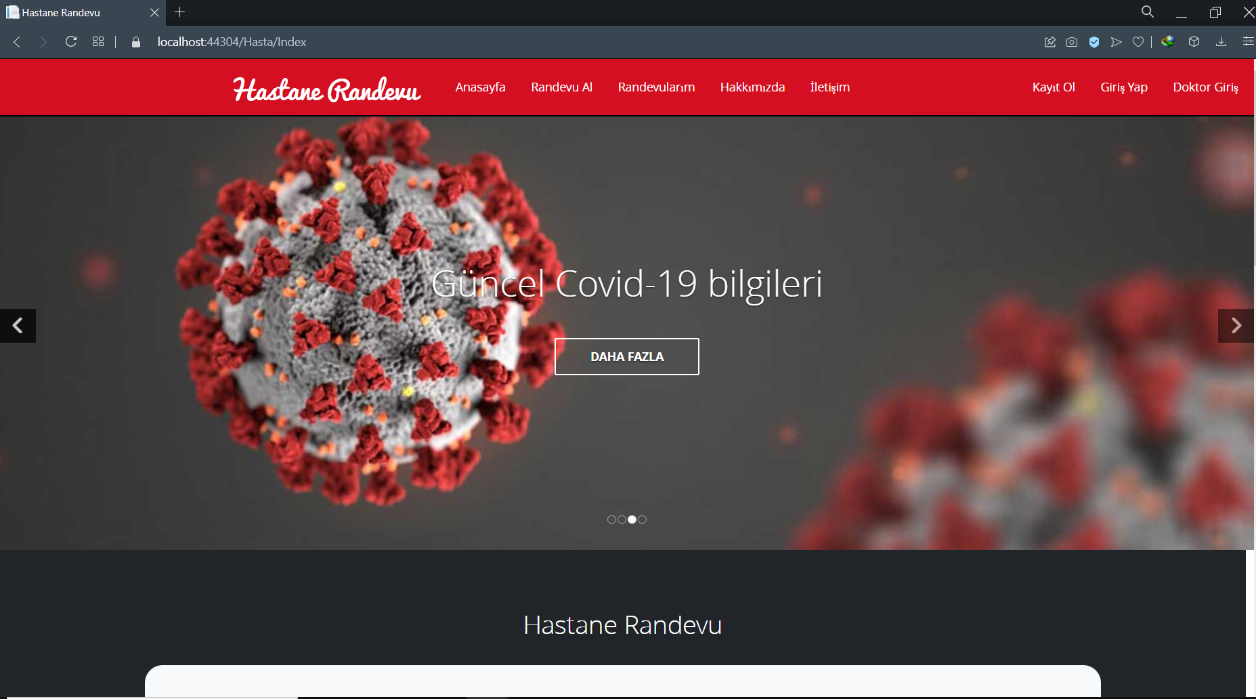
Stored Procedure’ün normal Query kodundan farkı query kodları çalıştırıldığında hata kontrolü yapılır ancak Stored Procedurler’de buna gerek yoktur. Bu nedenle Stored Procedure’ün çalıştırılması daha hızlıdır.



Şekil 4.3. Stored Procedure

### **4.2. Anasayfa Ekranı**

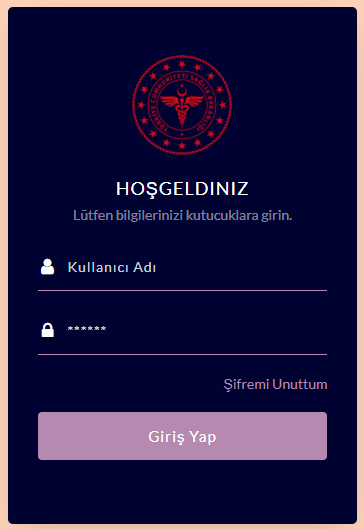
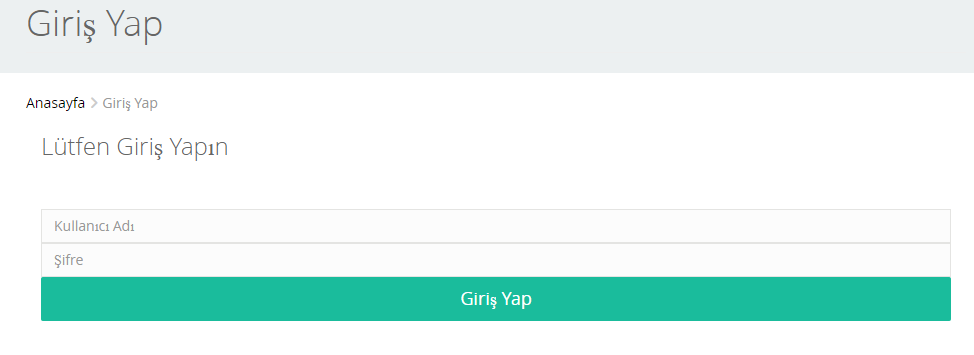
Anasayfa ekranında Bootstrap elemanlarından Carousel kullandım.



Şekil 4.5. Anasayfa Sayfası

### **4.3. Login Ekranı**

Projemde hasta ve doktor olmak üzere iki farklı login ekranı bulunmakta. Site üzerinde giriş yapılmadığı taktirde kullanıcının erişmesini istemediğimiz sekmeleri Authorize ve RoleProvider Class yapısı sayesinde engelliyorum.



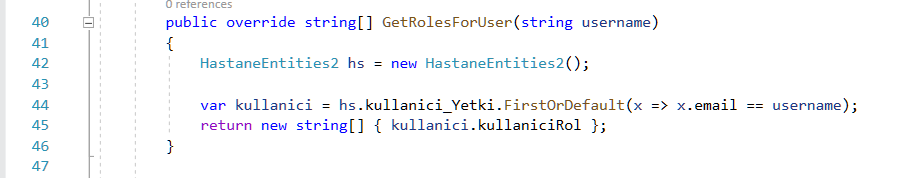
Şekil 4.1. Hasta Giriş Ekranı Şekil 4.2. Doktor Giriş Ekranı



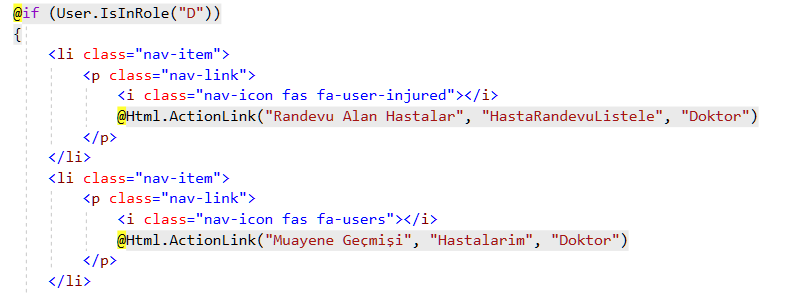
Şekil 4.6. Hasta Giriş Kod



Şekil 4.7. Doktor Giriş Kod



Şekil 4.8. Role Provider Class Yapısı



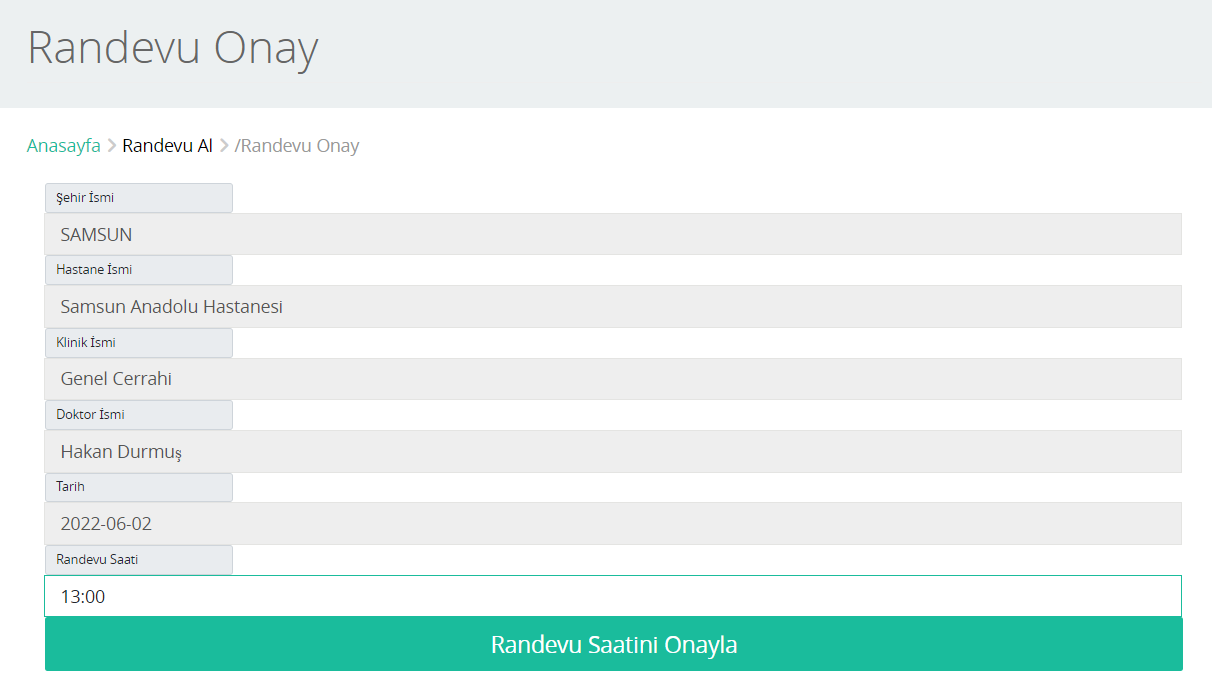
Şekil 4.9. Role Provider Rol Kontrol



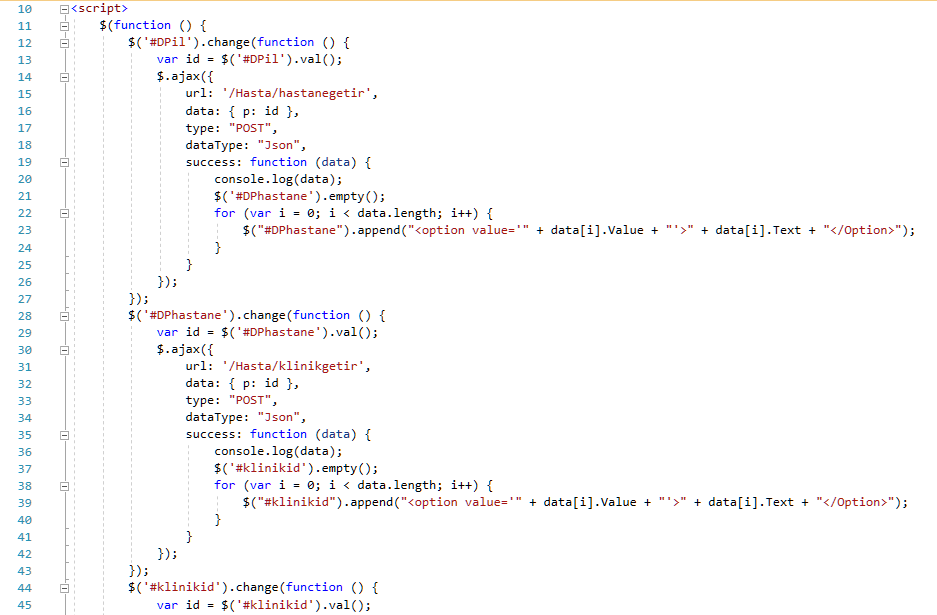
Şekil 4.10. Login View Tasarım Kodu

### **4.4. Randevu Sistemi**

Randevu sisteminde hasta seçtiği ildeki randevu almak istediği hastanenin ilgili polikliniğinde bulunan doktorlardan belirlediği tarihte ve saatte randevu almasına imkân tanıyor. Hasta aynı doktordan aynı gün randevu alma durumu engellenmiştir. Hasta randevularım sekmesinde geçmiş tarihli randevularını görebilir, gelecek tarihteki randevularından iptal etmek istediği randevuyu iptal edebilir.



Şekil 4.11. Randevu Al

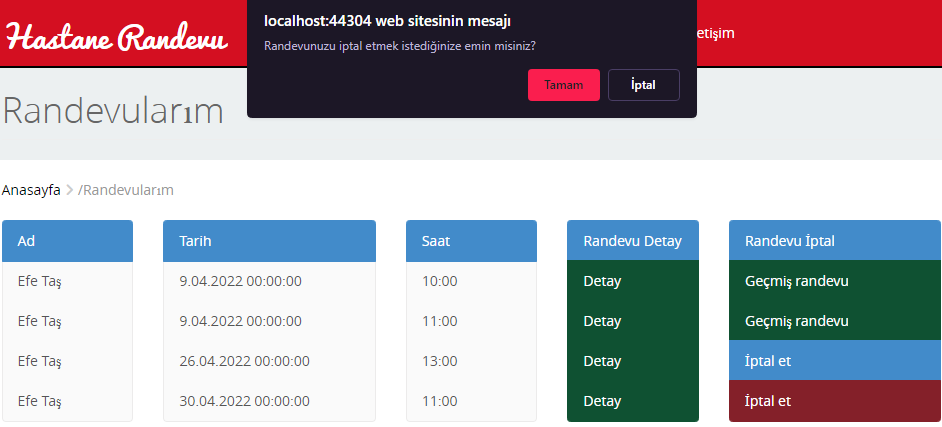


Şekil 4.12. Randevu Al Script Komutları





Şekil 4.13. Randevu Al Kodları



Şekil 4.14. Randevularım ve Randevu İptal



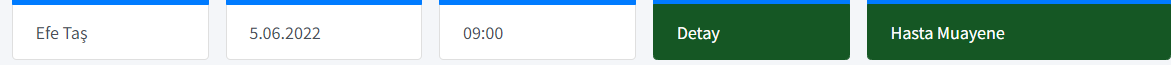
Şekil 3.15. Randevularım ve Randevu İptal Kodları

### **4.5. Hasta Muayene İşlemi**

Öncelik olarak randevu alan hastalar



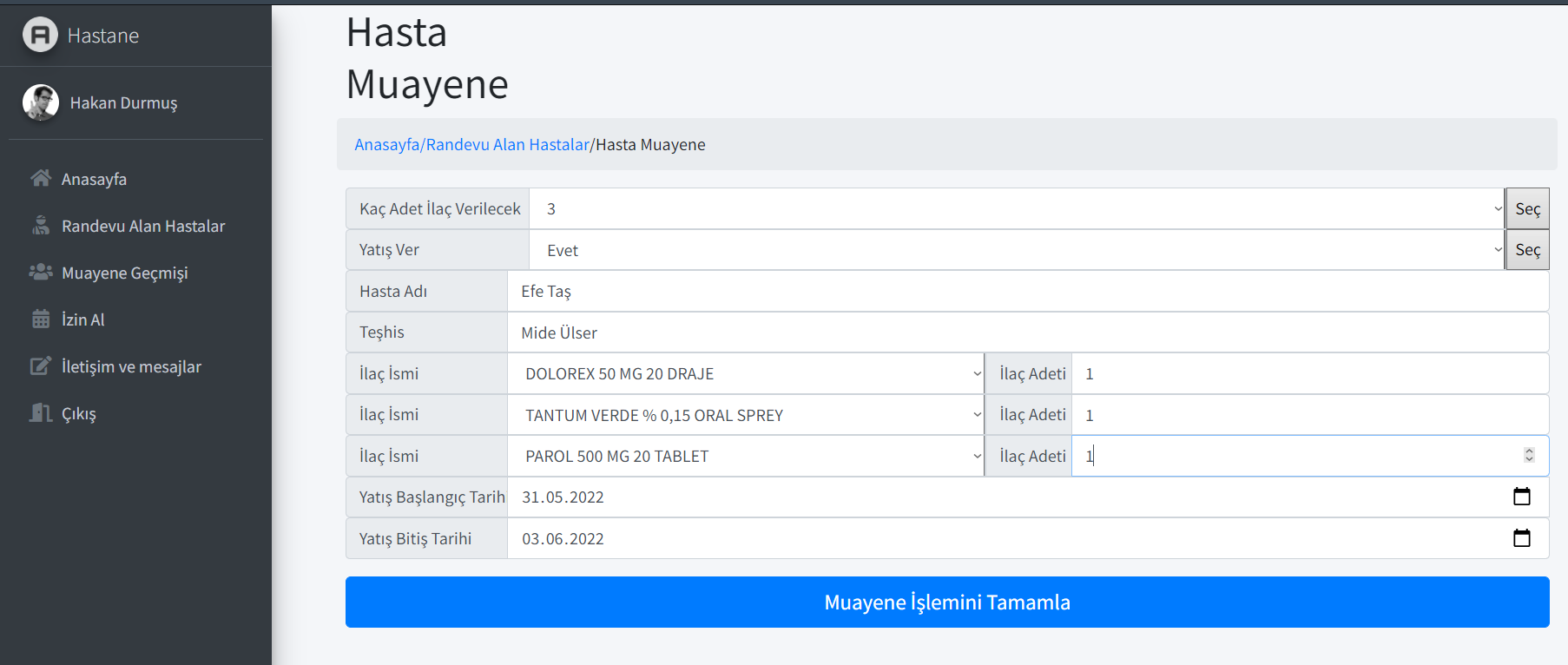
Kodu ile sadece o gün ki hasta randevuları listeleniyor.



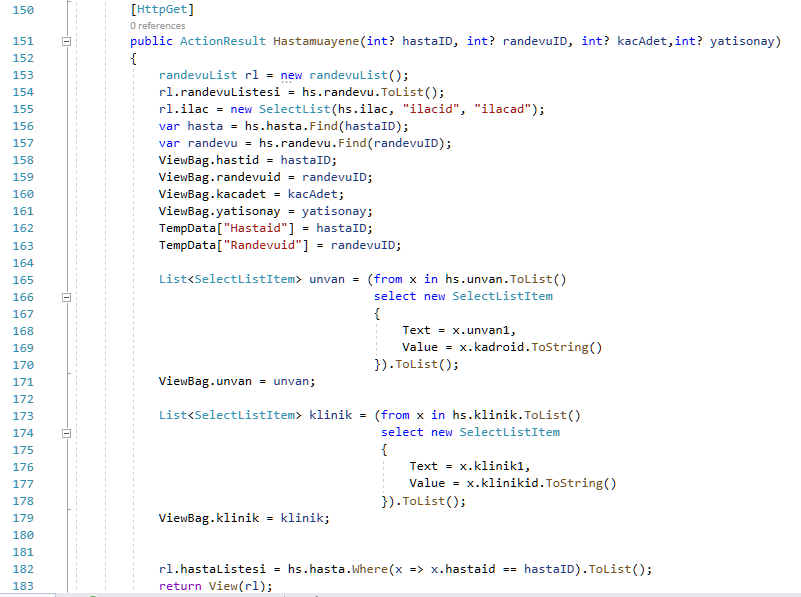
Hasta muayene butonuna tıklandığında muayene işlemi başlamış oluyor.

Öncelik olarak hastaya kaç adet ilaç verileceği ve yatış işlemi olup olmayacağı belirleniyor. Ardından verilecek ilaçlar dropdownlist de seçilip karşısındaki textboxa ilaç adeti giriliyor. Yatış verilecekse yatış başlangıç ve bitiş tarihleri seçiliyor. İşin Controller tarafında:

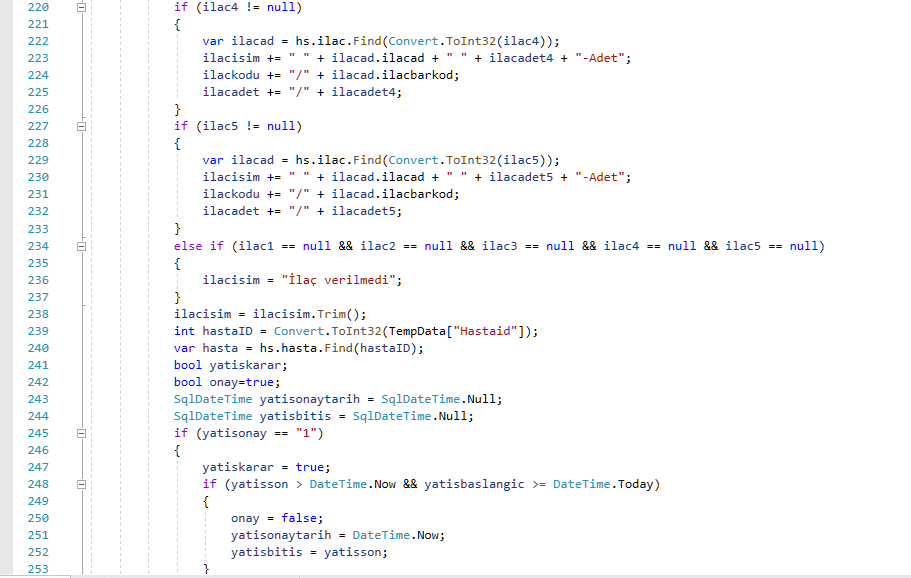
 Kodu ile hastaya özel reçete kodu oluşturuluyor. Bu kod ve hastanın gerekli bilgileri Web Service üzerinde ki “GetRecete” fonksiyonu ile MySQL deki eczane veri tabanına aktarılıyor.



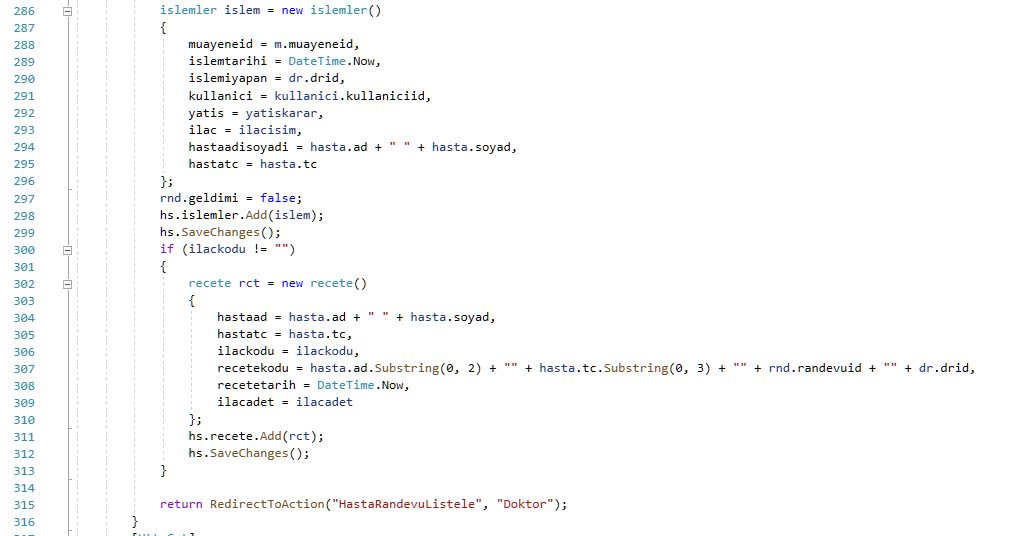
Şekil 4.16. Hasta Muayene Ekranı





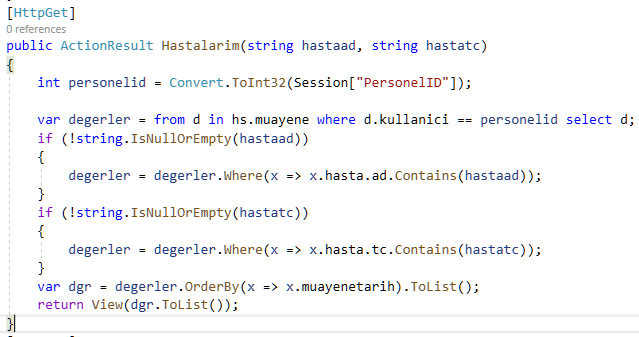






Şekil 4.17. Hasta Muayene Kod

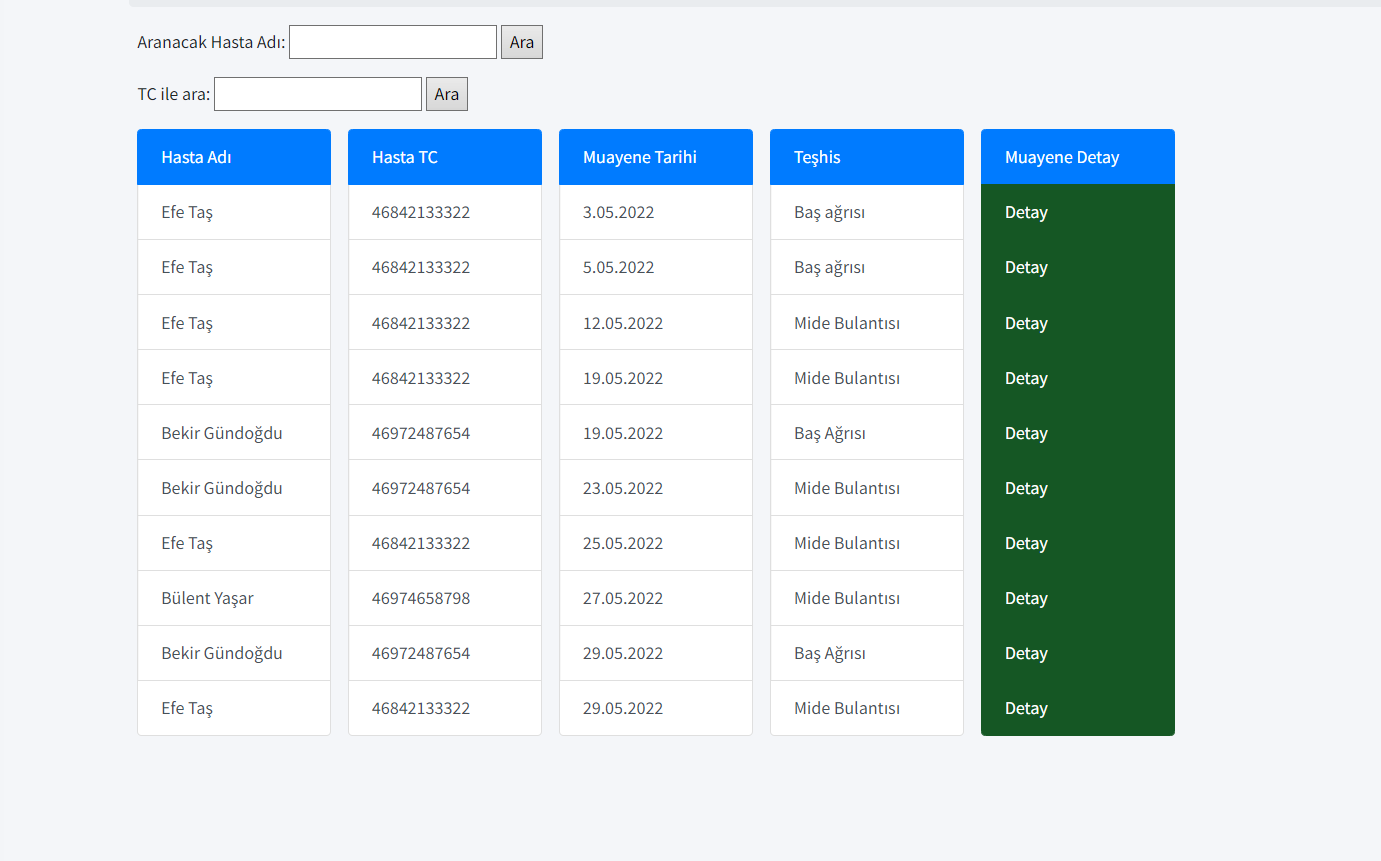
### **4.6. Hasta Muayene Geçmişi**



Sisteme giriş yapmış olan doktorun id si Session[“PersonelID”] de tutuluyor. Bu id ye göre geçmiş hasta muayeneleri listeleniyor. Sayfada hastanın adı ve TC kimlik numarasına göre arama işlemi gerçekleştirilebiliyor.



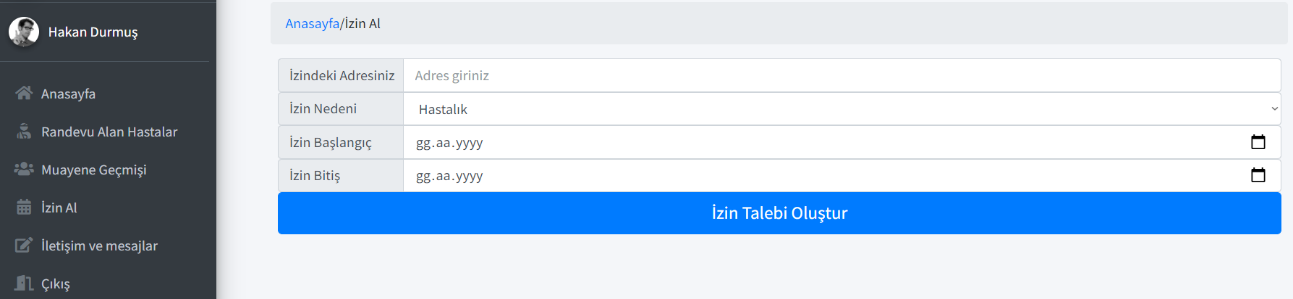
Yukarıda belirtilen kod sayesinde muayene listesi muayene tarihine göre sıralanıyor.



Şekil 4.18. Muayene Geçmişi Sayfası

### **4.7. Doktor İzin Sistemi**

Doktor sistem üzerinden belirli tarih aralıklarında izin talebi oluşturabilir. Oluşturulan izin talebi yönetici tarafından onaylanması gereklidir.



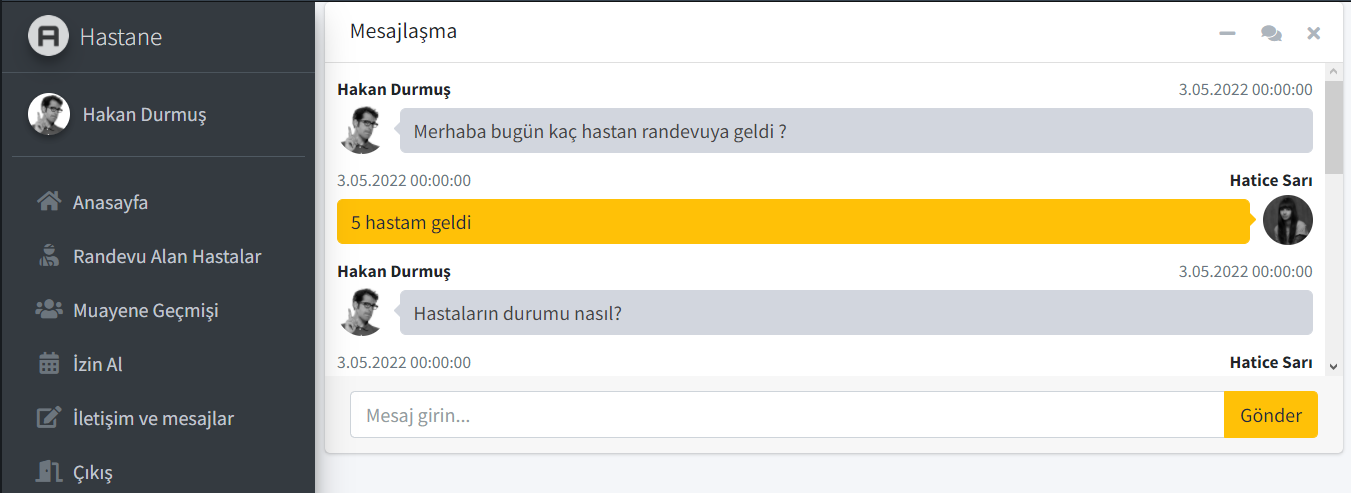
Şekil 4.19. İzin Talebi Sayfası



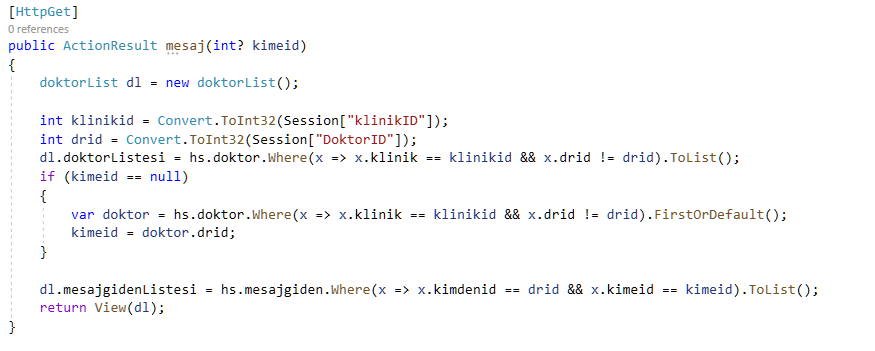
Şekil 4.20. İzin Talebi Kodları

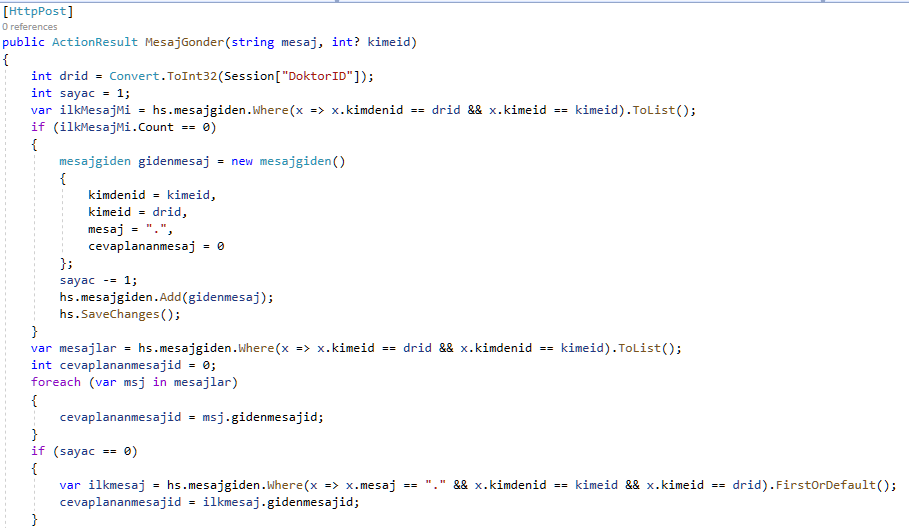
### **4.8. Doktorlar Arası Mesajlaşma**

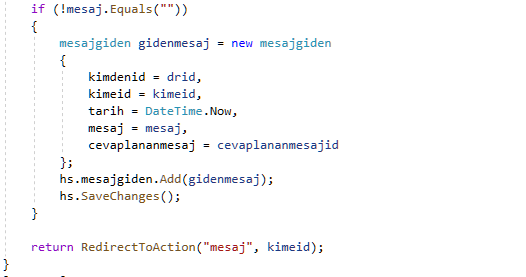
Doktor, kayıtlı olduğu klinikte bulunan diğer doktorlar ile mesajlaşabiliyor.



Şekil 4.21. Mesajlaşma Ekranı

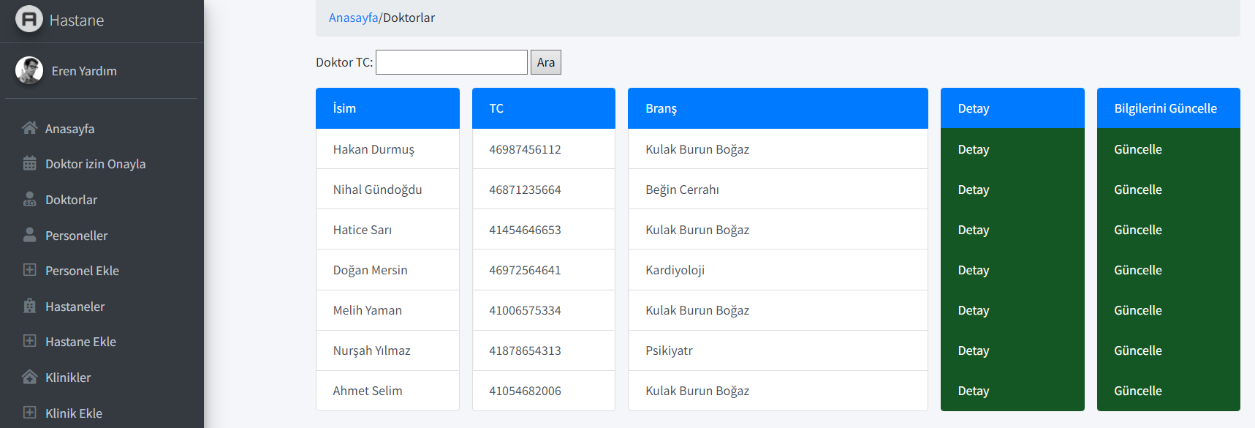




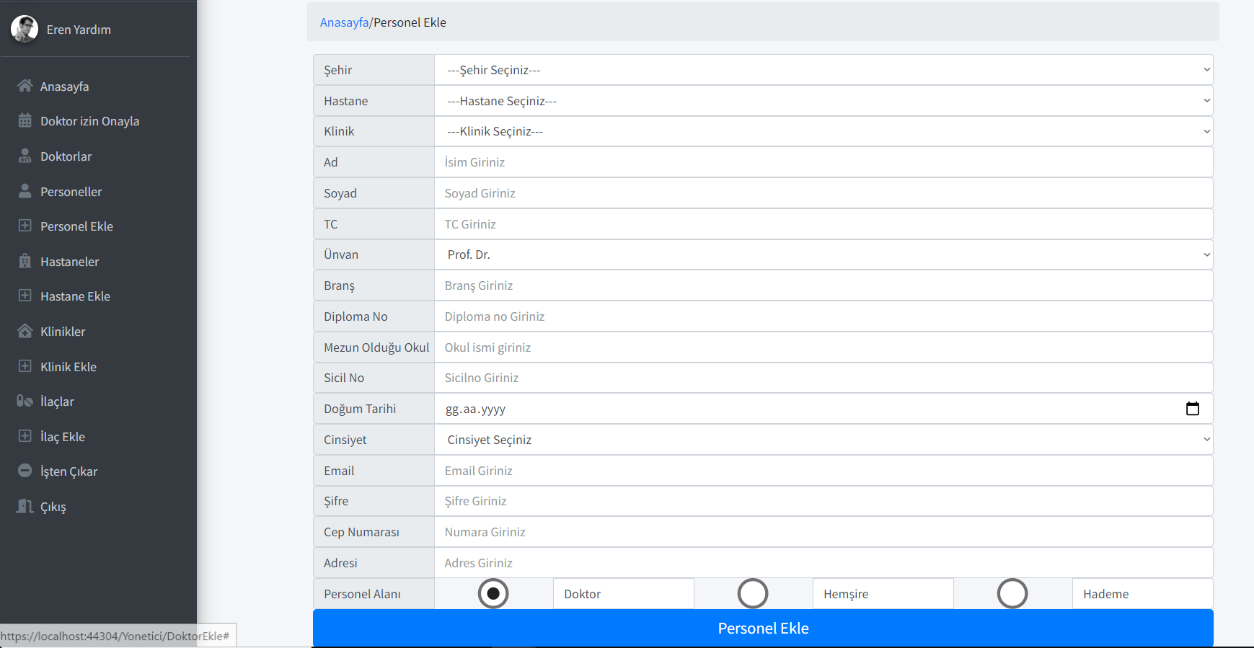


### **4.9. Personel Ekleme ve Listeleme**

Yönetici tarafından kontrol edilen alanlar.

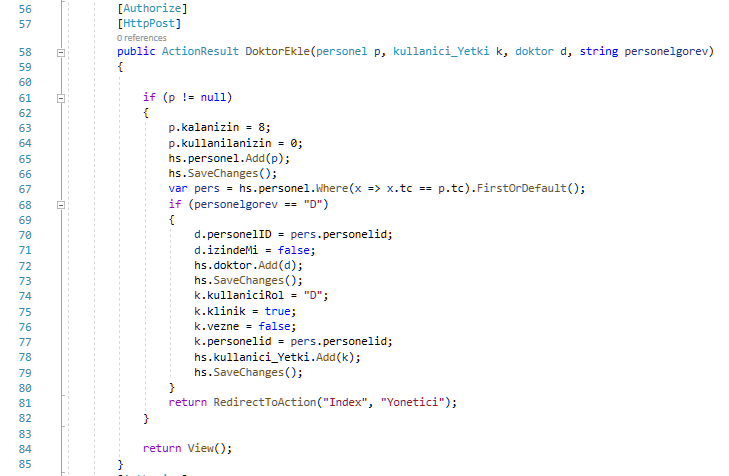


Şekil 4.22. Doktor Listele Sayfası



Şekil 4.23. Personel Ekle Sayfası

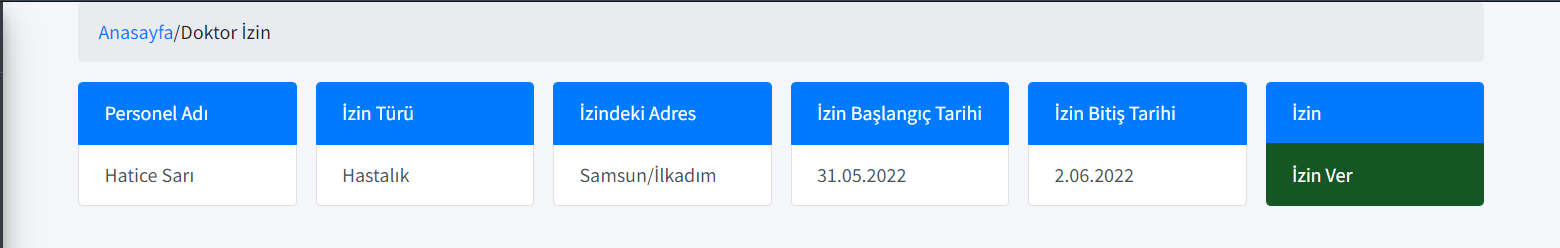




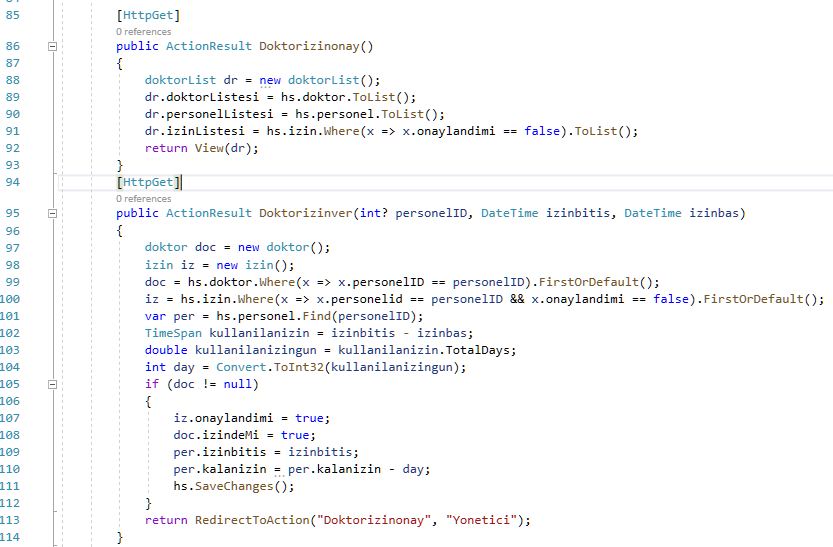
Şekil 4.24. Personel Ekle ve Listele Kod

### **4.10. Doktor İzin Onay**

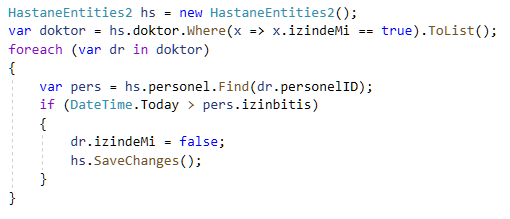
Doktorların oluşturduğu izin talepleri liste biçiminde ekrana geliyor ve “izin ver” butonuna tıklayarak izin verme işlemi gerçekleştiriliyor.



Şekil 4.25. Doktor İzin Onay Ekranı



Şekil 4.26. Doktor İzin Onay Kod

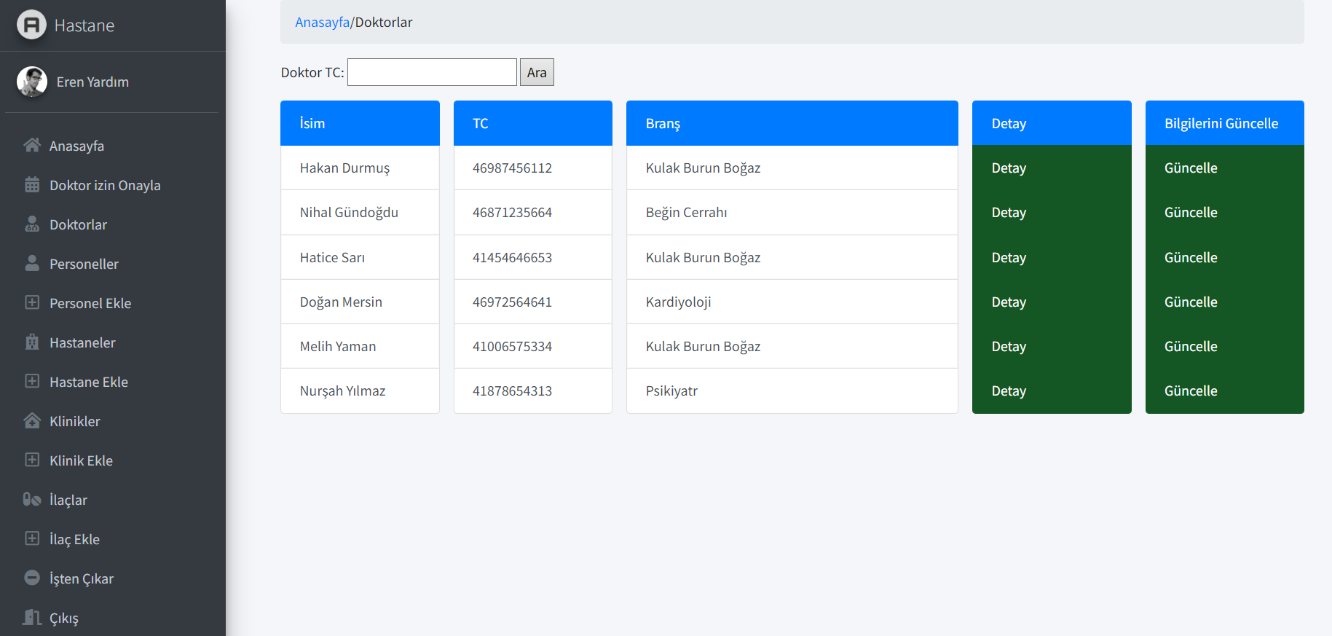


Globalde tanımlanmış yukarıdaki kod ile izin tarihi bitmiş olan doktorların tespiti gerçekleştiriliyor.

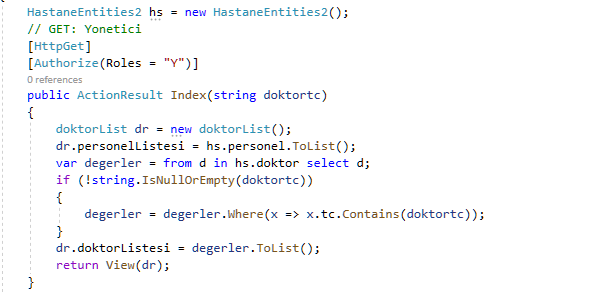
### **4.11. Doktorlar ve Personeller**

Doktorlar ve personeller sekmelerinin ara yüzü ve kodları oldukça benzer.

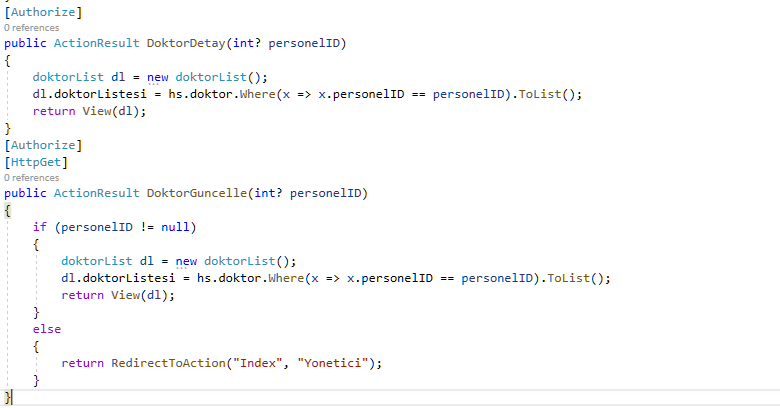
Aranan doktorun TC si ile ekrana o doktora ait bilgileri getirebiliyoruz. Doktora ait bilgilerin detayını görebilir ve doktora ait bilgileri güncelleyebiliriz.



Şekil 4.27. Doktorlar Ekranı



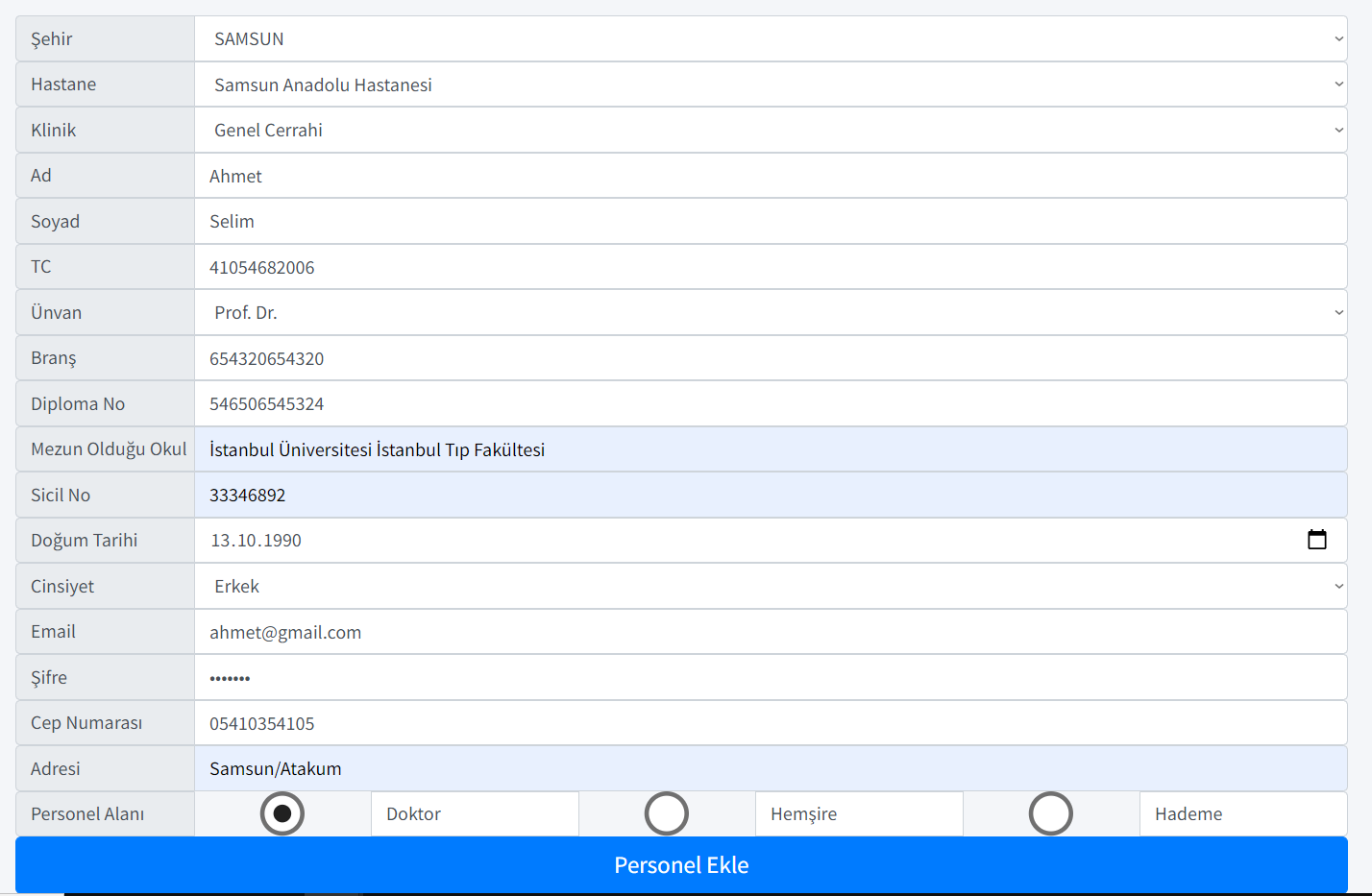
Şekil 4.28. Doktorlar Kod



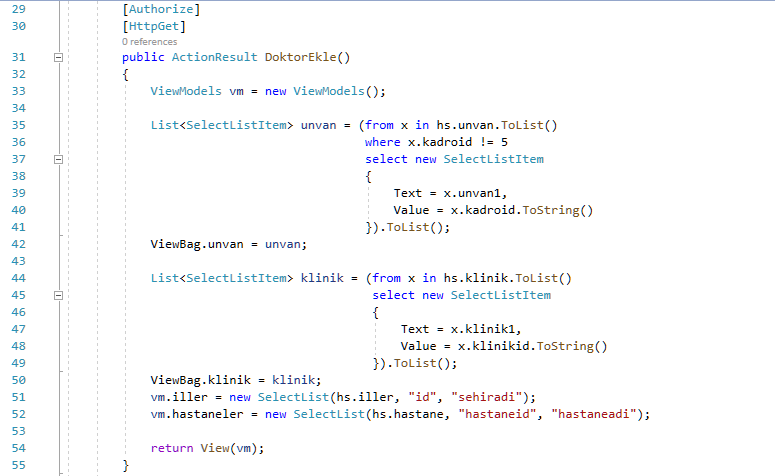
Şekil 4.29. Doktorlar Detay ve Güncelle Kod

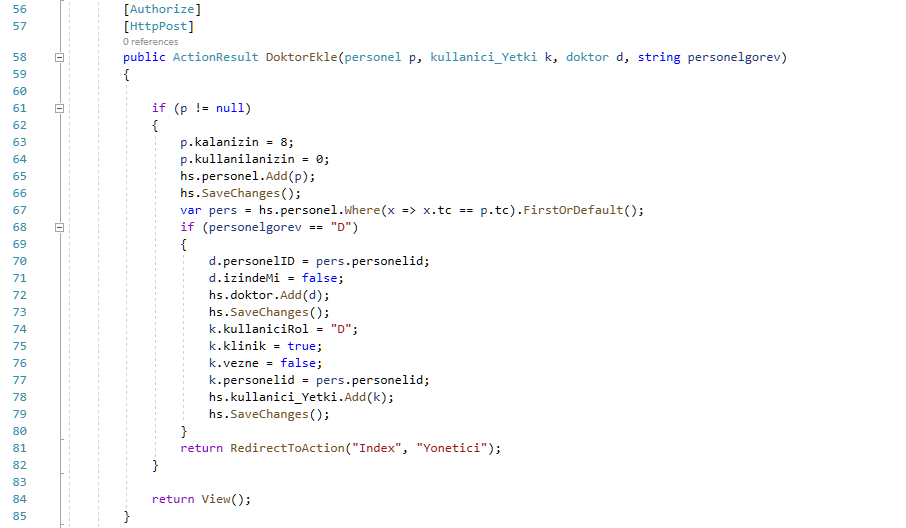
### **4.12. Personel Ekle**

Seçilen şehirde belirtilen hastanenin kliniğine radio button ile seçilen alanda personel ekleme işlemi gerçekleştirilir.



Şekil 4.30. Personel Ekleme Ekranı

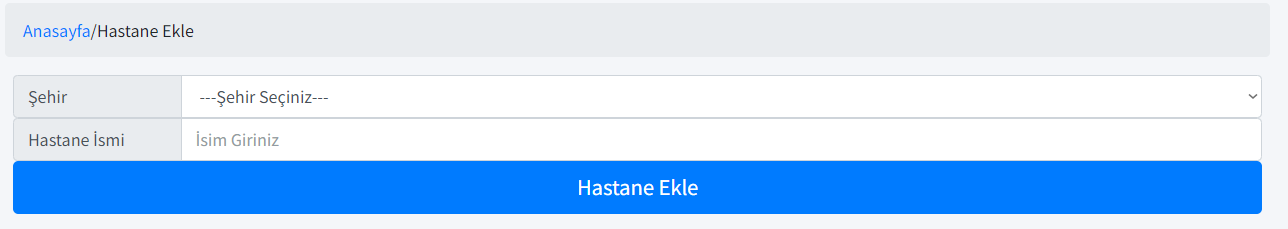




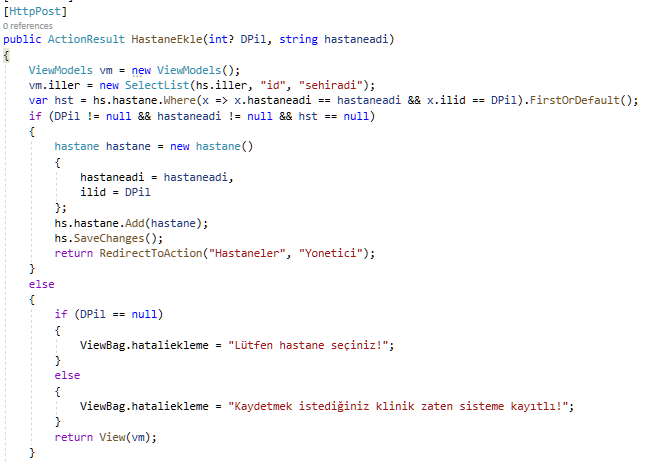
Şekil 4.31. Personel Ekleme Kod

### **4.13. Hastane CRUD işlemleri**

Dropdownlistde hastanenin ekleneceği ili seçtikten sonra hastane ismini yazarak ekliyoruz.

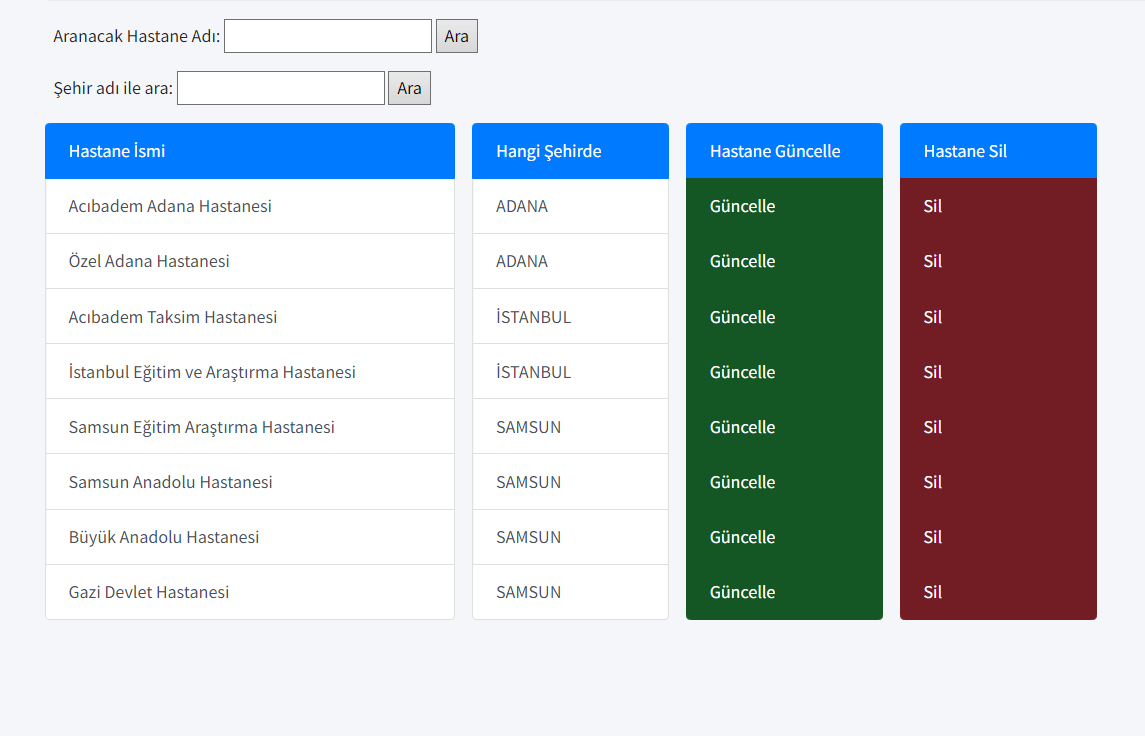


Şekil 4.32. Hastane Ekle Sayfası

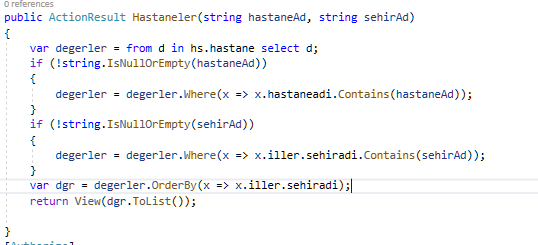


Şekil 4.33. Hastane Ekle Kod

Hastanenin adı ve bulunduğu şehir ile hastane araması gerçekleştirilebiliyor. Listelenme şehir ismine göre A’dan Z’ye olacak şekilde gerçekleşiyor.

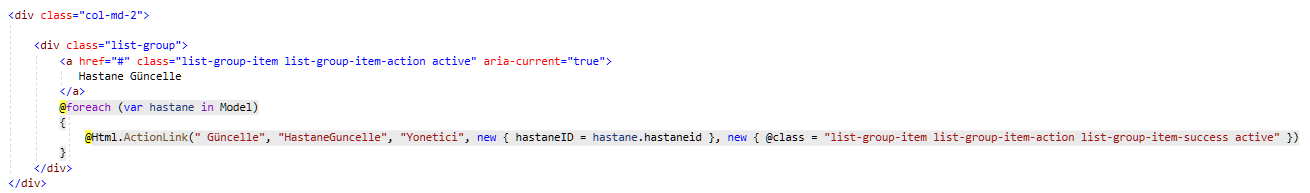


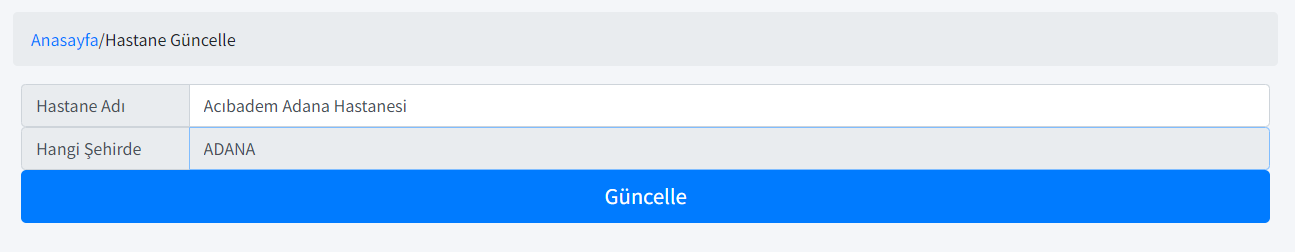
Şekil 4.34. Hastane Listele Ekranı



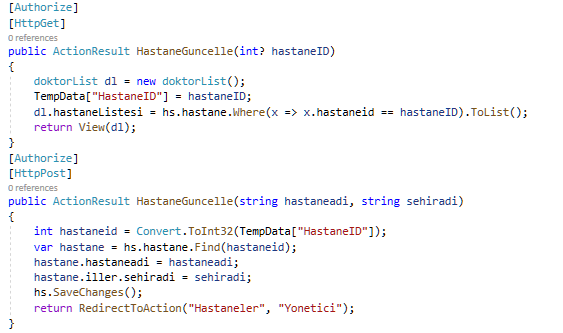
Şekil 4.35. Hastaneler Kod

Hastane güncelleştirilmek istendiğinde “Hastaneler” sekmesindeki “Güncelle” butonuna tıklayarak güncelleştirilmek istenen hastanenin id sini Html.Actionlink ile “HastaneGuncelle” controllerına gönderir.



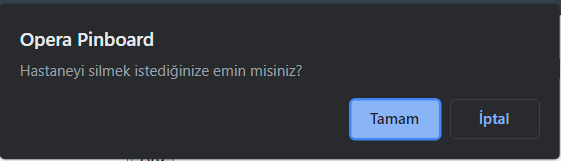


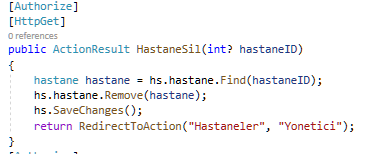
Şekil 4.36. Hastane Güncelleştirme Ekranı



Şekil 4.37. Hastane Güncelleştirme Kod

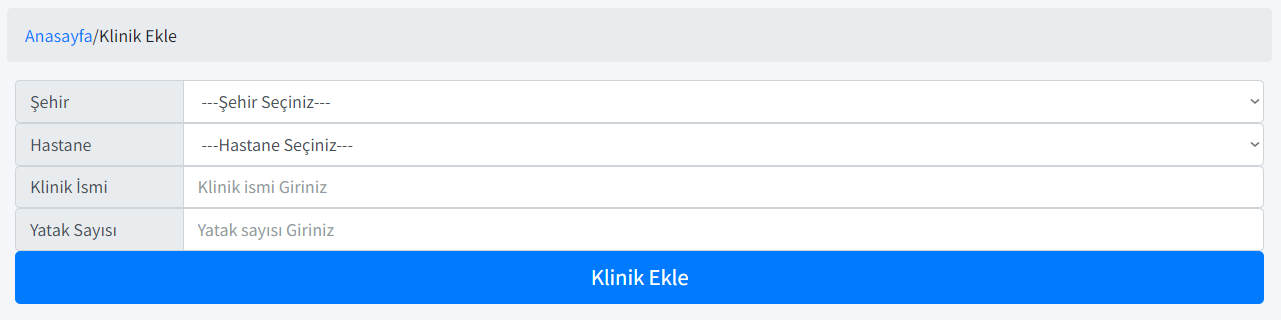
Hastane silinmek istendiğinde “Hastaneler” sekmesindeki “Sil” butonuna tıklandığında gelen onay ekranında tamama tıklandığında hastanenin id sini Html.Actionlink ile “HastaneSil” controllerına gönderir.



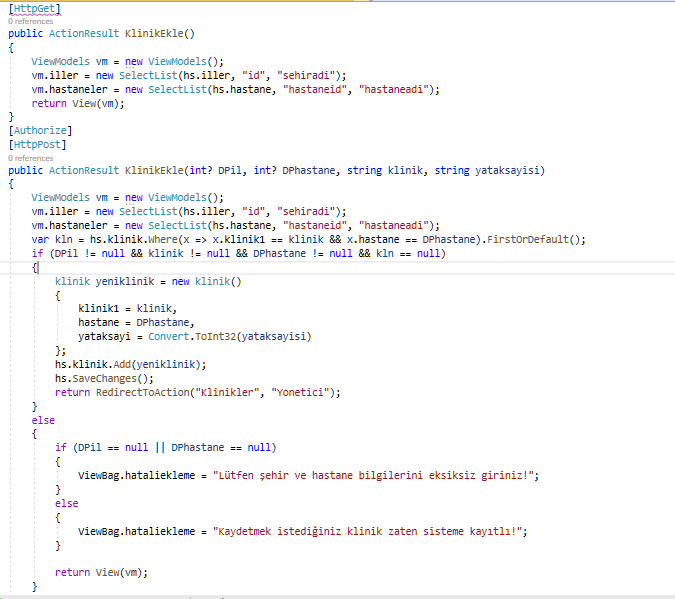


### **4.14. Klinik CRUD İşlemleri**

Dropdownlistde kliniğin ekleneceği ilde ki hastaneyi seçtikten sonra klinik ismini ve yatak sayısını yazarak klinik ekleme işlemini gerçekleştiriyoruz.

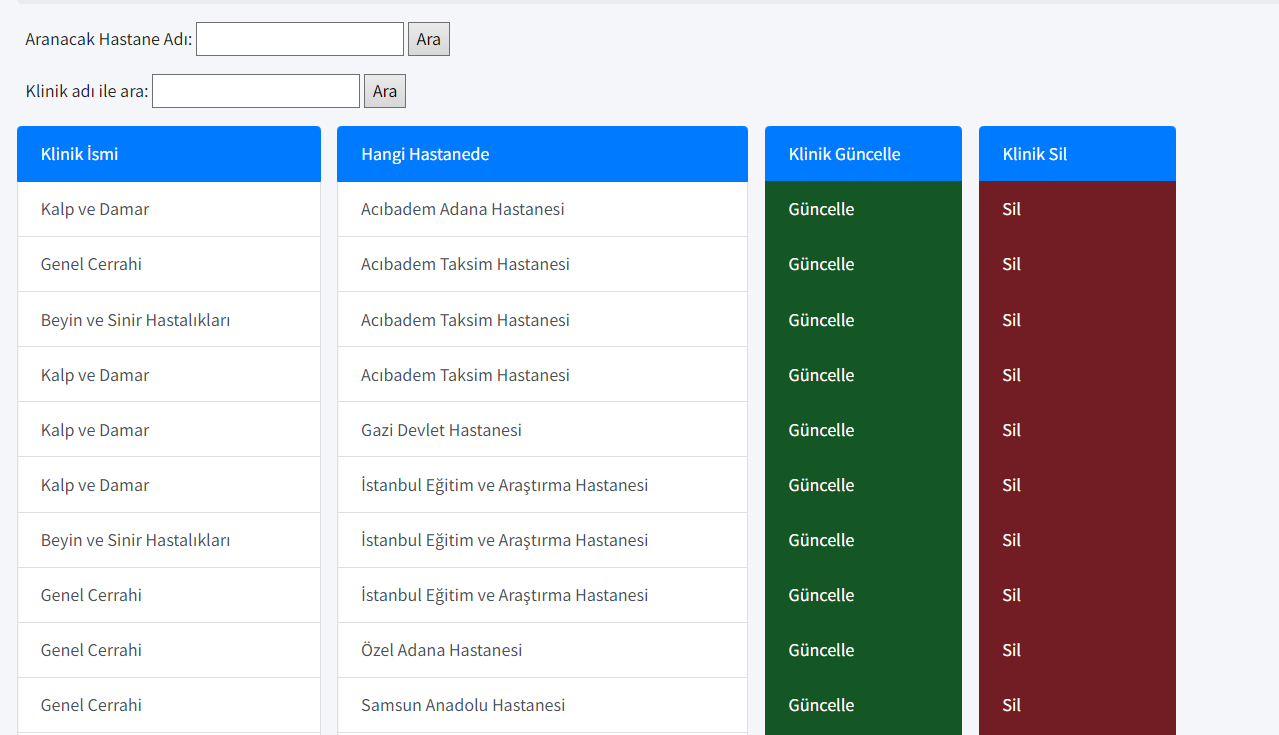


Şekil 4.38. Klinik Ekle Sayfası

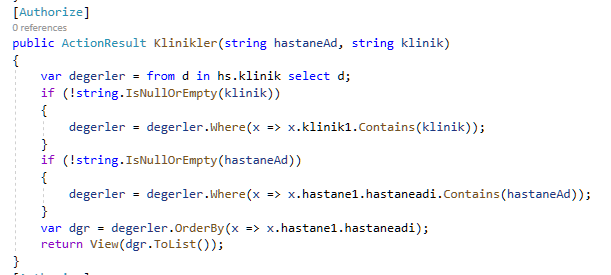


Şekil 4.39. Klinik Ekle Kod

Kliniğin adı ve bulunduğu hastane ile hastane araması gerçekleştirilebiliyor. Listelenme hastane ismine göre A’dan Z’ye olacak şekilde gerçekleşiyor.

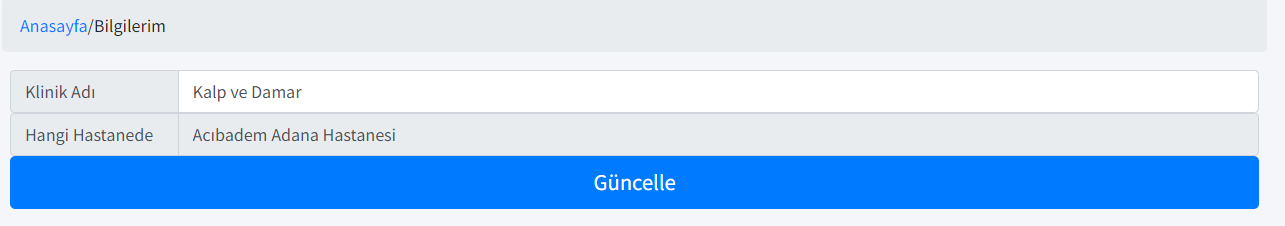


Şekil 4.40. Klinik Listele Ekranı

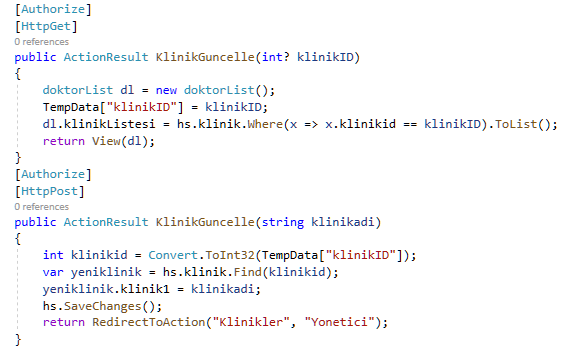


Şekil 4.41. Klinik Listele Kodu

Klinik güncelleştirilmek istendiğinde “Klinikler” sekmesindeki “Güncelle” butonuna tıklayarak güncelleştirilmek istenen kliniğin id sini Html.Actionlink ile “KlinikGuncelle” controllerına gönderir.

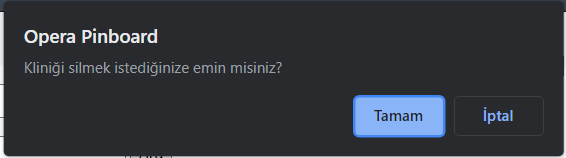


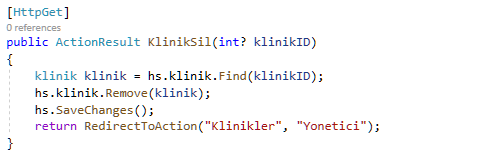
Şekil 4.42. Klinik Güncelle Sayfası



Şekil 4.43. Klinik Güncelle Kod

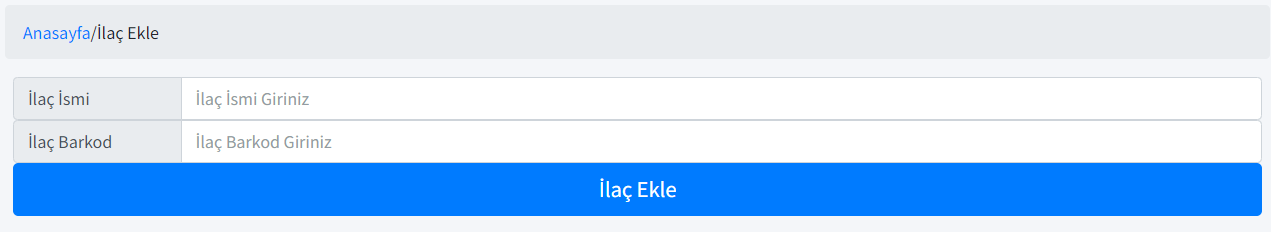
Klinik silinmek istendiğinde “Klinikler” sekmesindeki “Sil” butonuna tıklandığında gelen onay ekranında tamama tıklandığında kliniğin id sini Html.Actionlink ile “KlinikSil” controllerına gönderir.



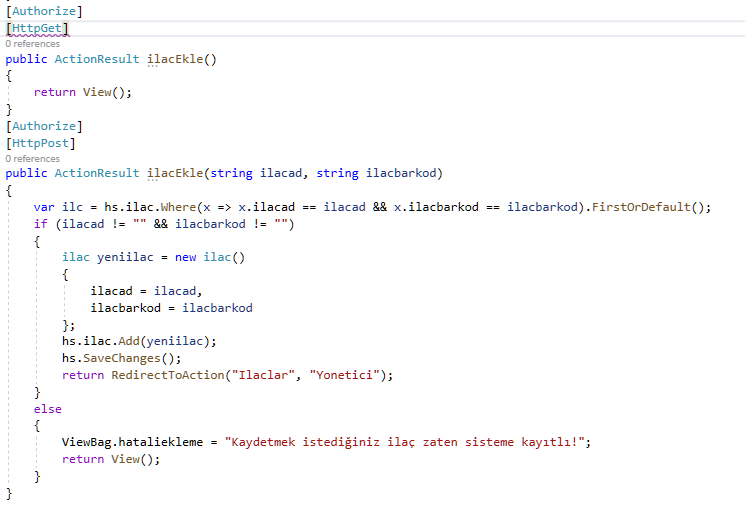


### **4.15. İlaç CRUD İşlemleri**

Eklenmek istenen ilacın ismini ve barkod numarasını girerek ekleme işlemi yapılır.



Şekil 4.44. İlaç Ekle Sayfası

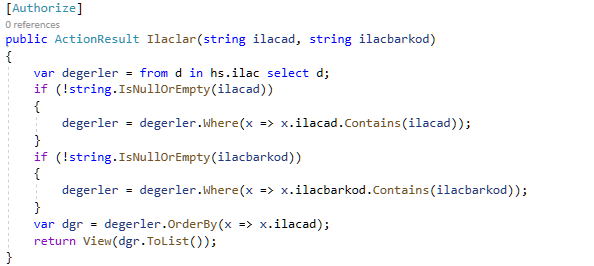


Şekil 4.45. İlaç Ekle Kod

İlacın ismi ve barkod numarası ile arama işlemi gerçekleştirilebiliyor. Listelenme ilaç ismine göre A’dan Z’ye olacak şekilde gerçekleşiyor.



Şekil 4.46. İlaçlar Sayfası



Şekil 4.47. İlaç Listele Kodu

# 5. Eczane Otomasyonu Görünüm ve Açıklamaları

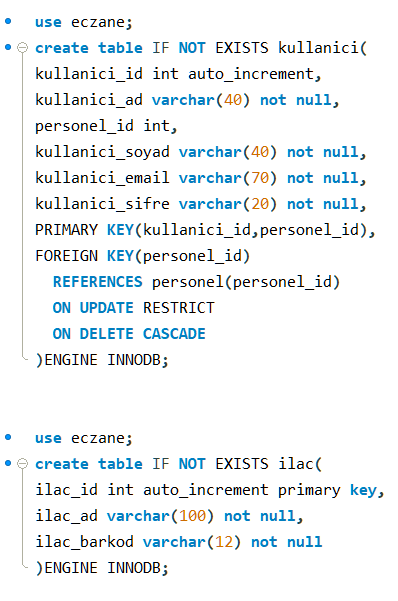
### **5.1. Eczane Veri Tabanı**

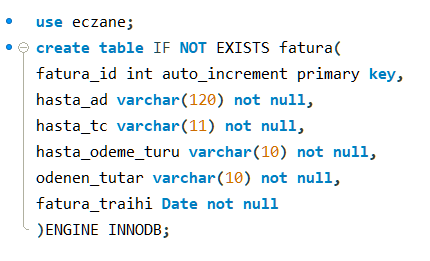
MySQL’de oluşturduğum veri tabanını projeye ADO.NET Entity Data Model üzerinden EF Designer from database yaklaşımı ile dahil ettim. Modele dahil edilecek veritabanı bağlantısını, model için ayarları ve veritabanı nesnelerini seçtikten sonra uygulamamızın etkileşimde bulunacağı sınıflar modelden oluşturulur.

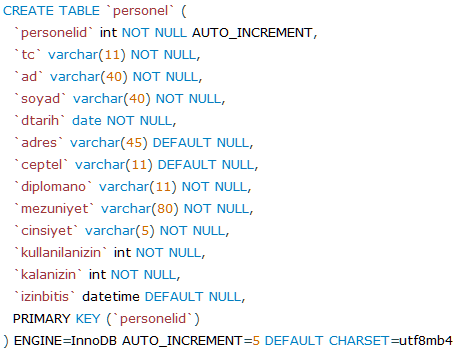


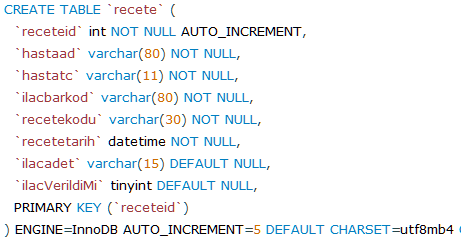
Şekil 5.1. MySQL Bağlantısı.

Eczane otomasyonu için MySQL’de “eczane” adlı veri tabanı oluşturdum. Eczane veri tabanına ait tabloları oluşturduğum Query kodları:



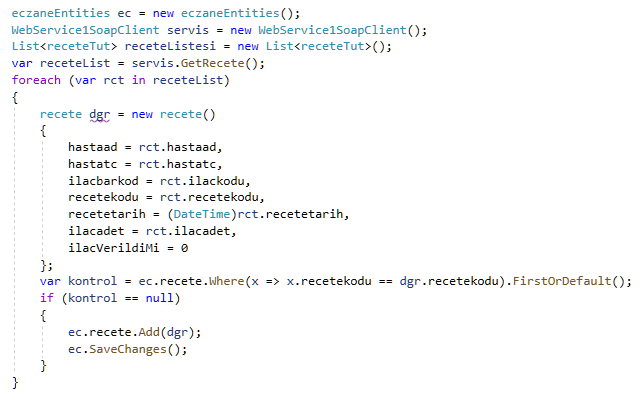


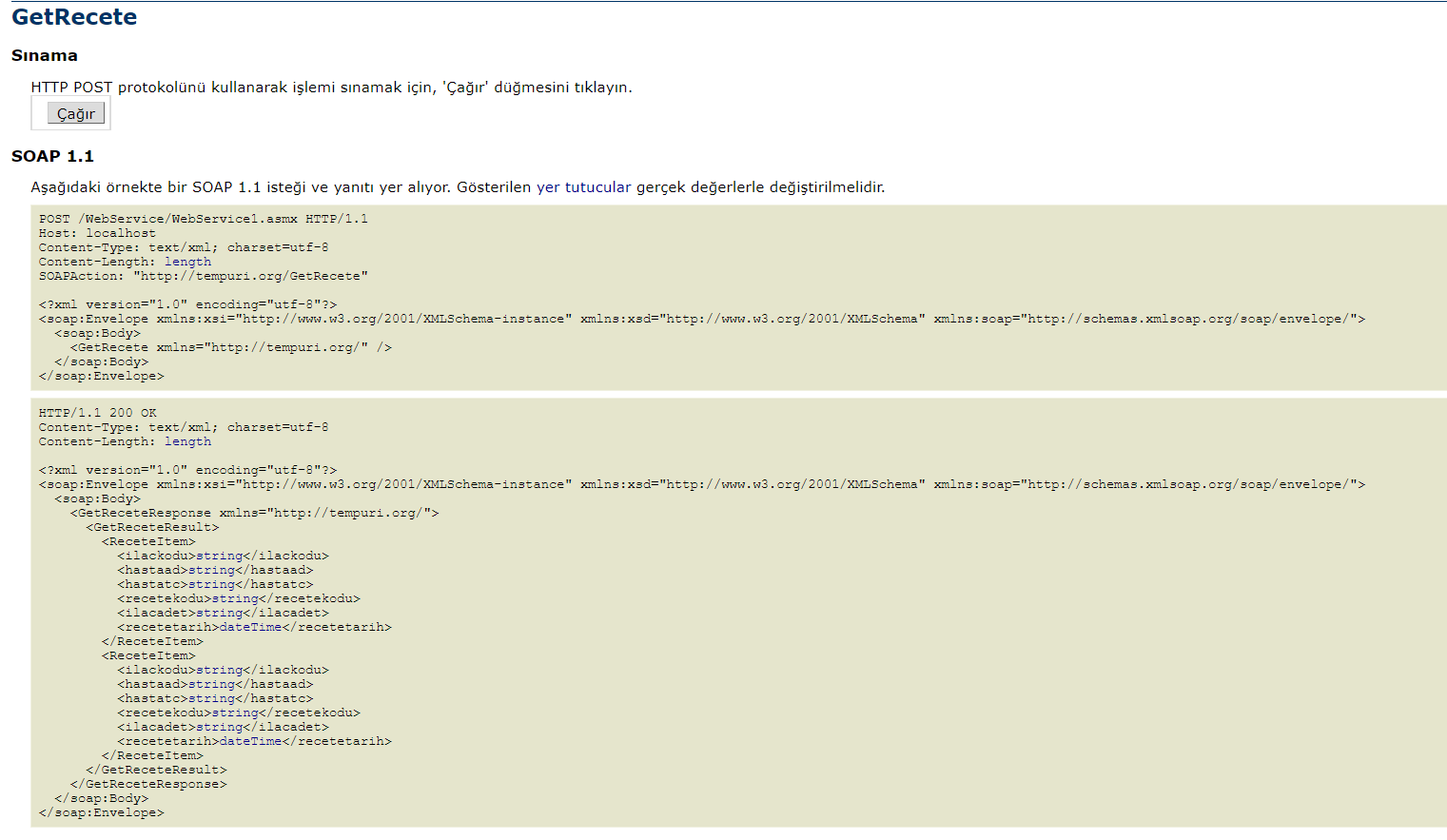




### **5.2. Reçeteli İlaç Ver**

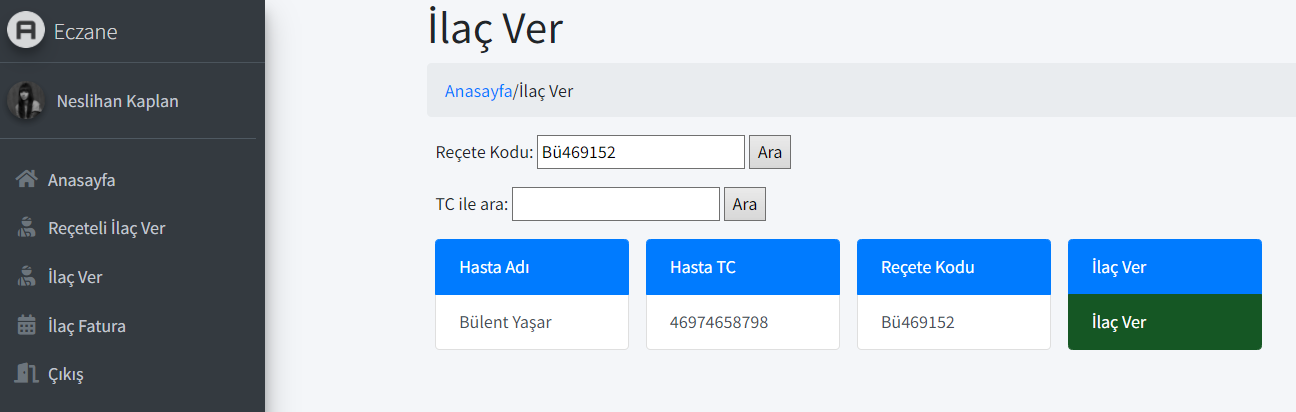
İlaç alacak olan hastanın bilgileri ve reçete kodu projenin global sayfasında tanımlanmış olan aşağıdaki kod ile hastane otomasyonunda yapılmış muayeneye ait sonucu Web Service üzerinden eczane veri tabanındaki recete adlı tabloya aktarılıyor.



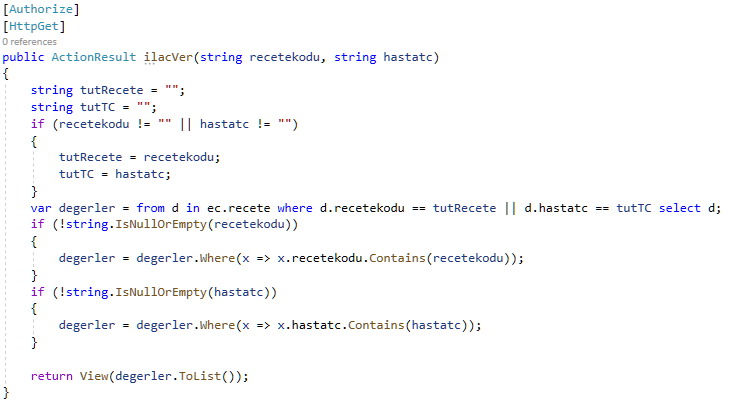


Şekil 5.2. Web Service GetRecete Fonksiyonu

İlaç almak isteyen kişinin hastanede ki muayenesi sonunda aldığı reçete kodu ile veya reçetesini kaybettiği durumu göz önünde bulundurarak hastanın TC’si ile reçete aratma işlemi gerçekleştiriliyor.

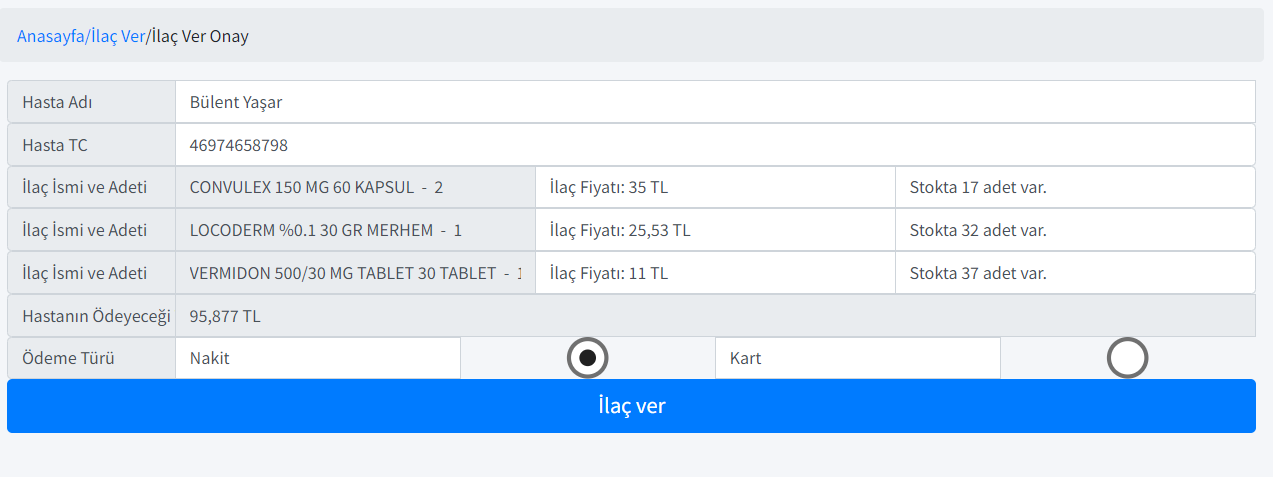


Şekil 5.3. İlaç Ver Sayfası



Şekil 5.4. İlaç Ver Kod

“İlaç Ver” Butonuna tıklandığında “İlaç Ver Onay” sayfasına yönlendiriliyoruz. Bu sayfada hastaya hangi ilaçtan kaç adet verileceği ve hastanın ödeyeceği toplam tutar ekrana geliyor. “İlaç Ver” butonuna tıklayarak ilaç verme işlemini onaylamış oluyoruz. İlaç verme işlemi gerçekleştirildikten sonra verilen ilaçlar stoktan düşürülüyor.



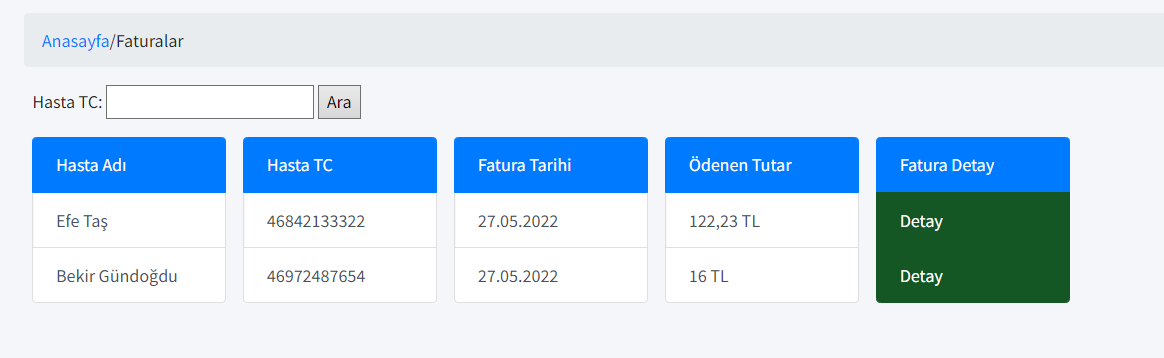
Şekil 5.5. İlaç Ver Onay Sayfası



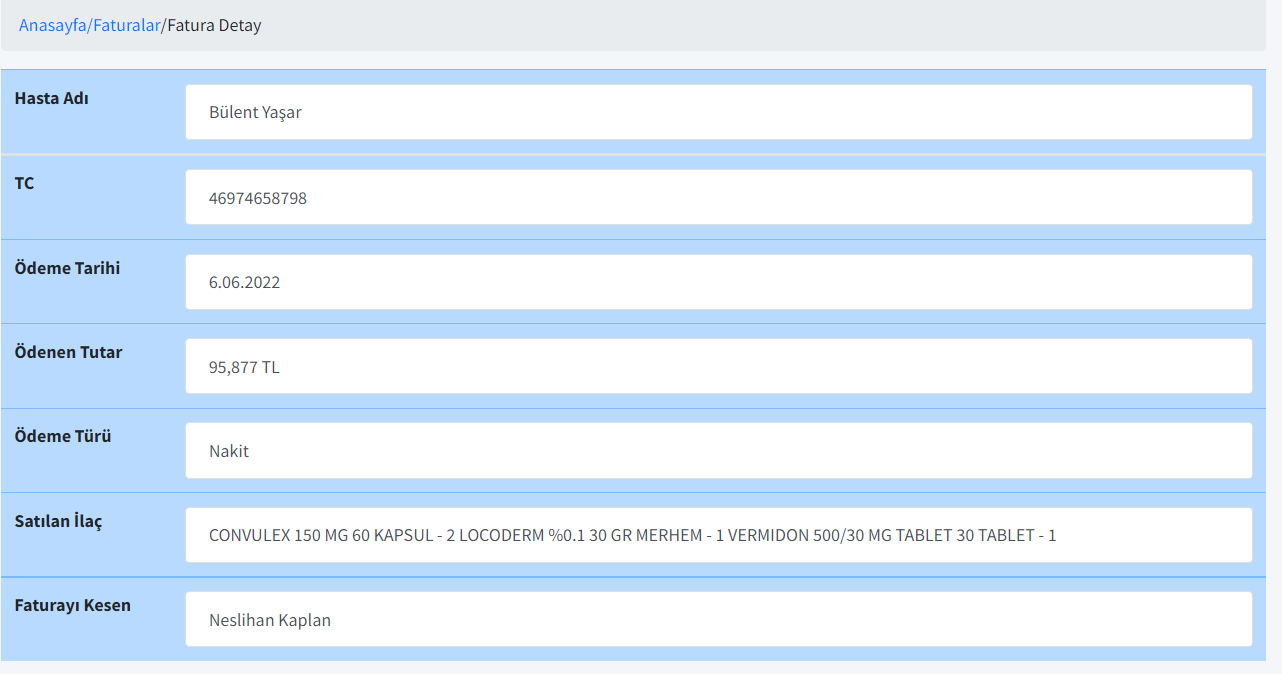
Şekil 5.6. İlaç Ver Kod

### **5.3. Faturalar**

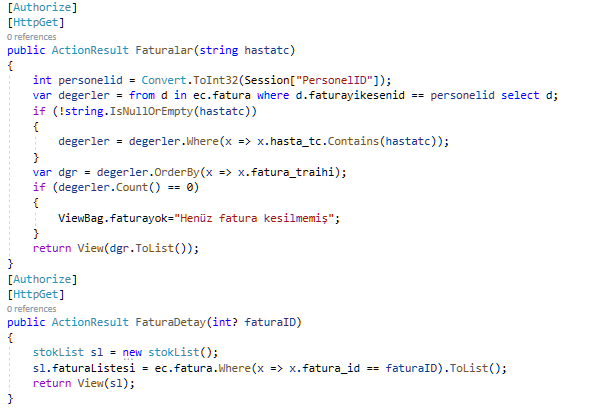
Giriş yapmış olan personelin yapmış olduğu fatura kesimleri listeleniyor. Listeleme işlemi eski fatura tarihinden yeniye doğru olacak şekilde yapılıyor. Hastanın TC sine göre fatura aranan fatura bulunabiliyor.



Şekil 5.7. Faturalar Sayfası



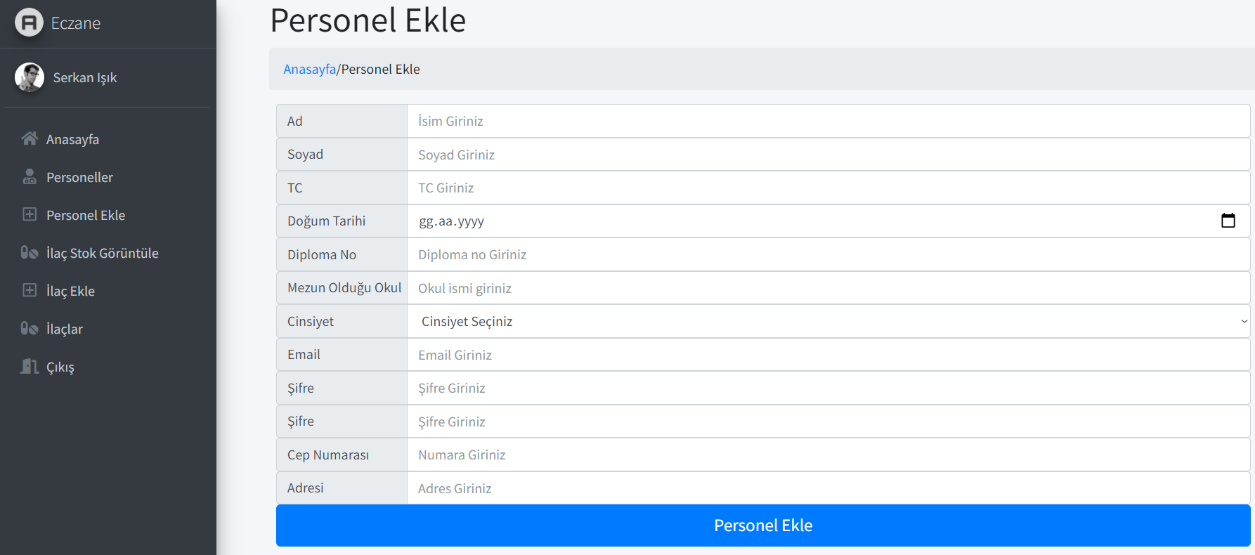
Şekil 5.8. Faturalar Detay Sayfası



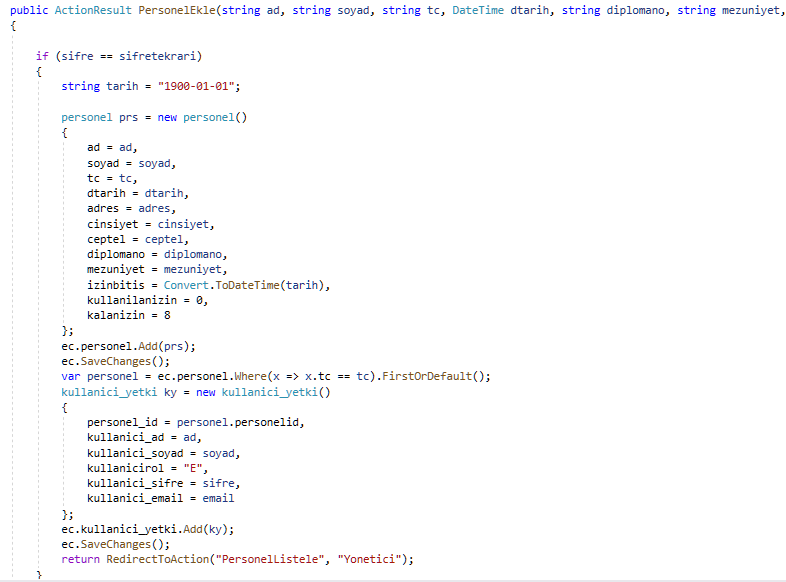
Şekil 5.9. Faturalar Kod

### **5.4. Personel CRUD İşlemleri**

Personeller ekle sayfasında personele ait bilgileri girerek

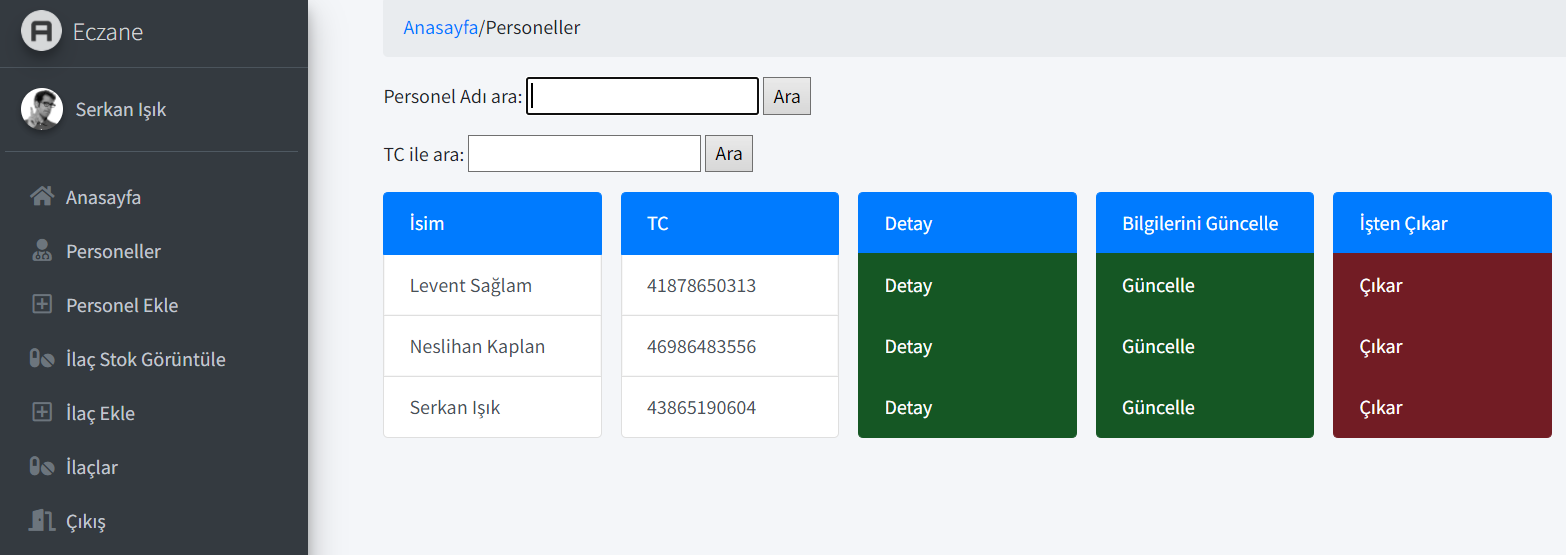


Şekil 5.10. Personel Ekle Sayfası

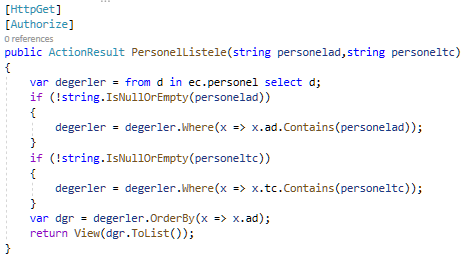


Şekil 5.11. Personel Ekle Kod

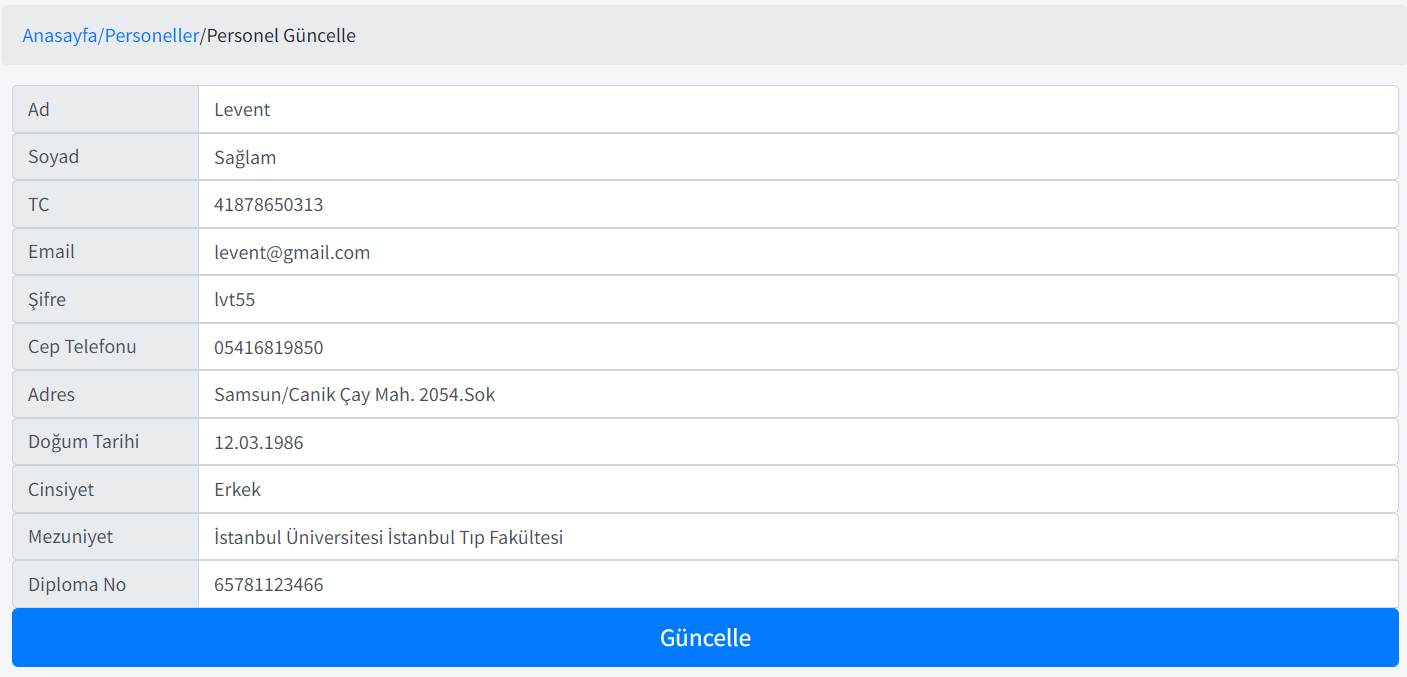
Bulmak istenen personel adı ve TC ile aratılabiliyor.



Şekil 5.12. Personeller Sayfası



Şekil 5.13. Personeller Kod



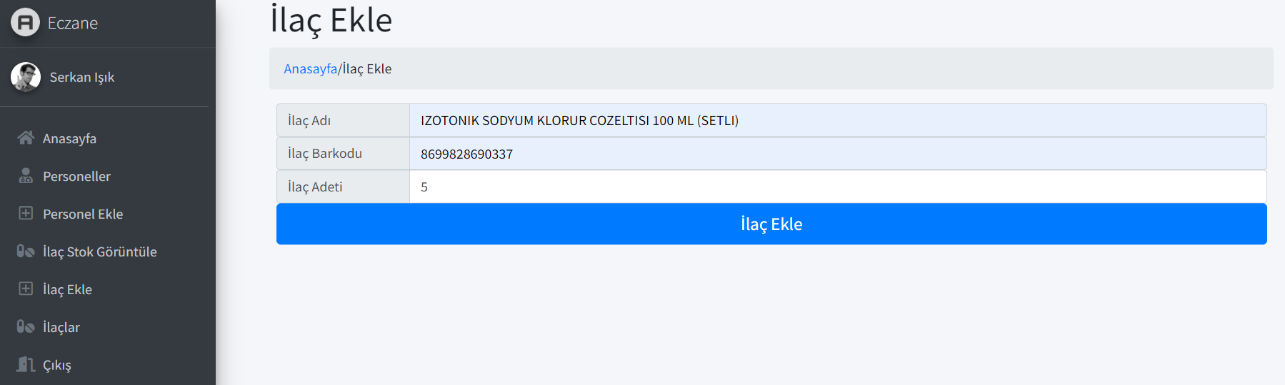
Şekil 5.14. Personel Güncelle Sayfası



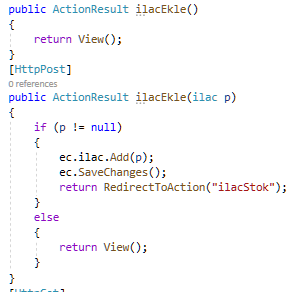
Şekil 5.15. Personel Güncelle Kod

### **5.6. İlaç CRUD İşlemleri**

Eklemek istenen ilacın adını, barkodunu ve kaç adet ekleneceğini girerek ilaç ekleme işlemini tamamlamış oluyoruz.

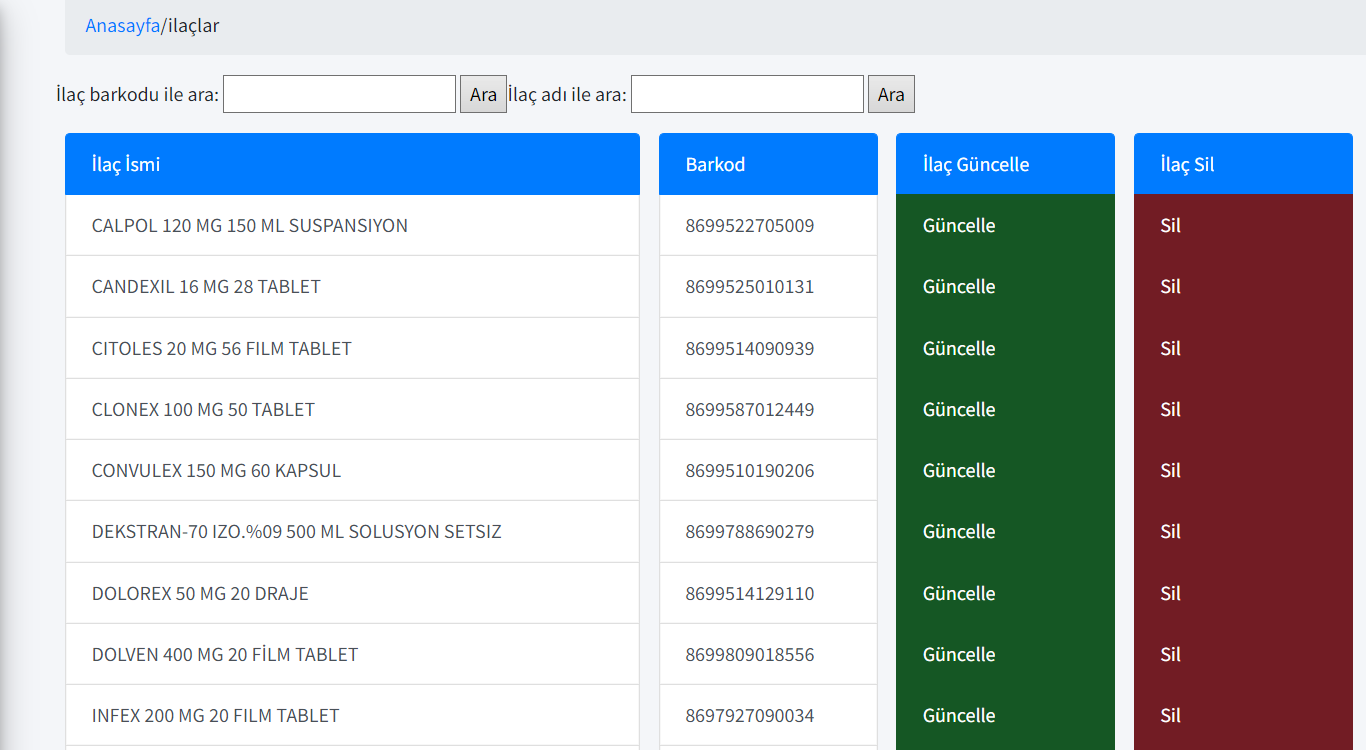


Şekil 5.16. İlaç Ekle Sayfası

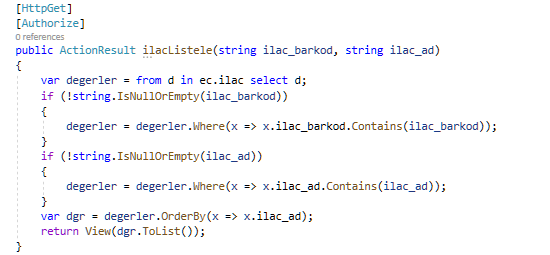


Şekil 5.17. İlaç Ekle Kod

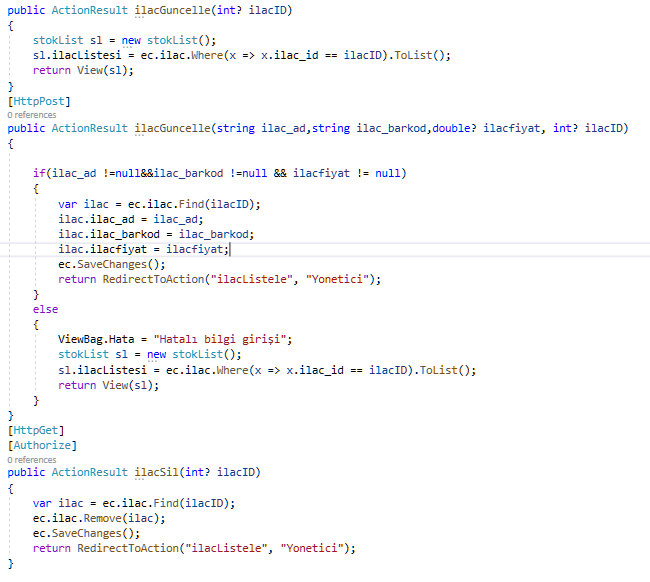
İlaçlar sekmesinde sisteme kayıtlı ilaçları görüyoruz. Aranan ilacı barkodu ve ismi aracılığıyla aratabiliyoruz. Güncellemek istediğimiz ilacı “Güncelle” butonuna tıklayarak güncelleme ekranına yönlendiriliyoruz ve yapacağımız değişiklikleri onaylıyoruz. Silmek istediğimiz ilacı “Sil” butonuna tıkladığımızda gelen uyarı mesajını onayladıktan sonra silme işlemi gerçekleşiyor.



Şekil 5.18. İlaçlar Sayfası

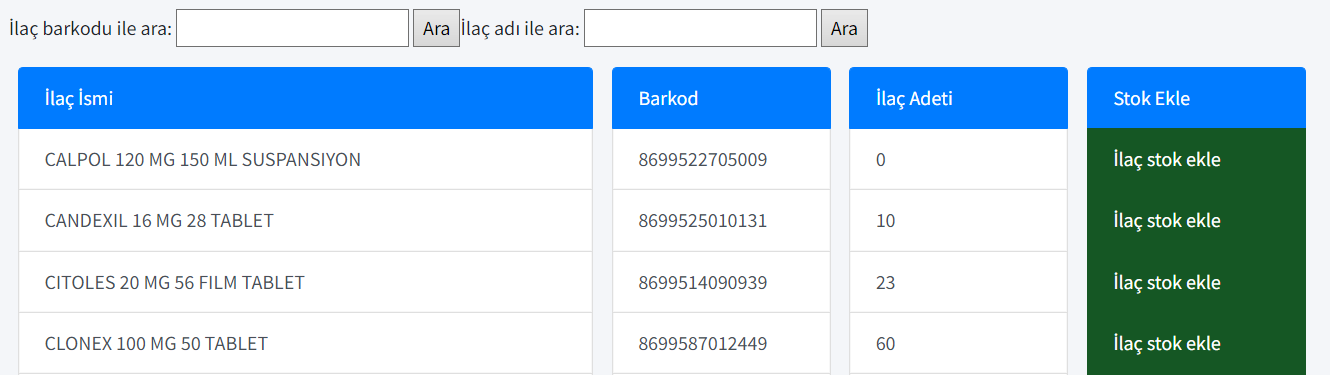


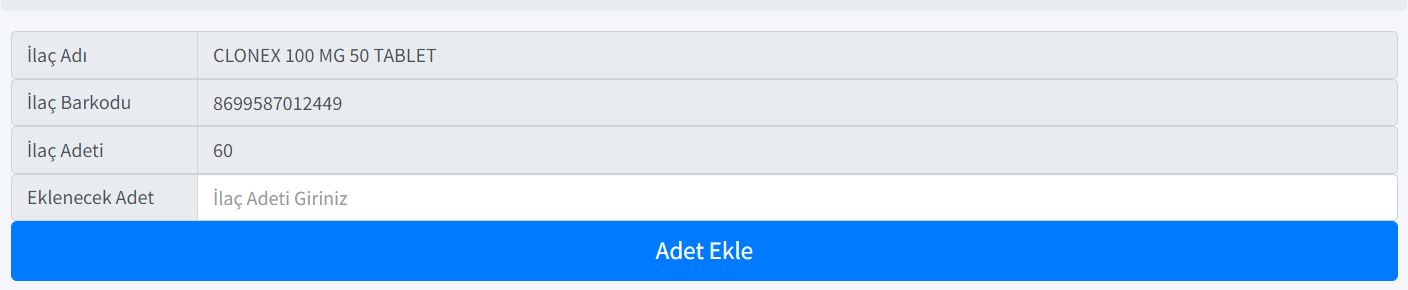
Şekil 5.19. İlaçlar Kod



Şekil 5.20. İlaçlar Güncelle ve Sil Kodları

Önceden kayıt edilmiş ilaçların stokunu güncelleyebiliyoruz.

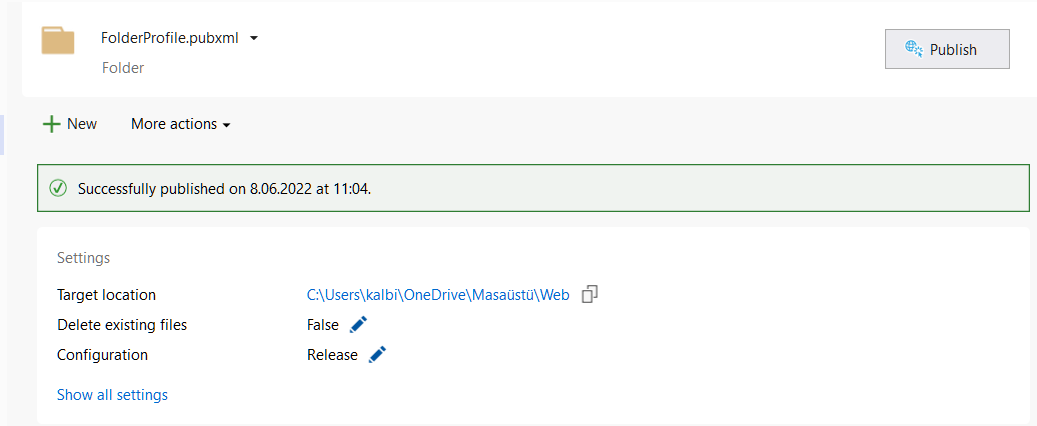




Şekil 5.21. İlaç Stok Ekle Sayfası

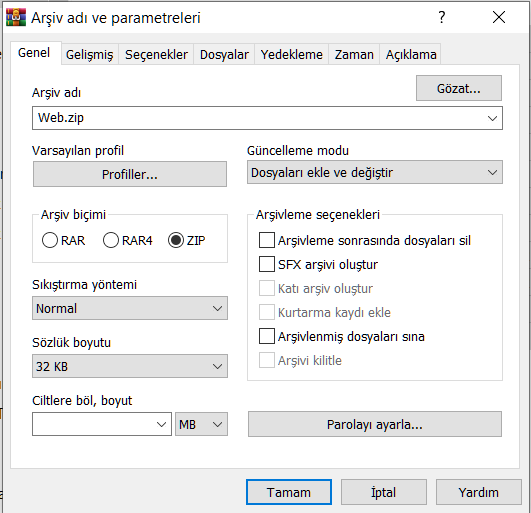
# 6. Projeyi Somee Sunucusuna Yükleme

Projeyi web sunucusuna yüklemek için önce proje dosyalarını “publish” etmemiz lazım.



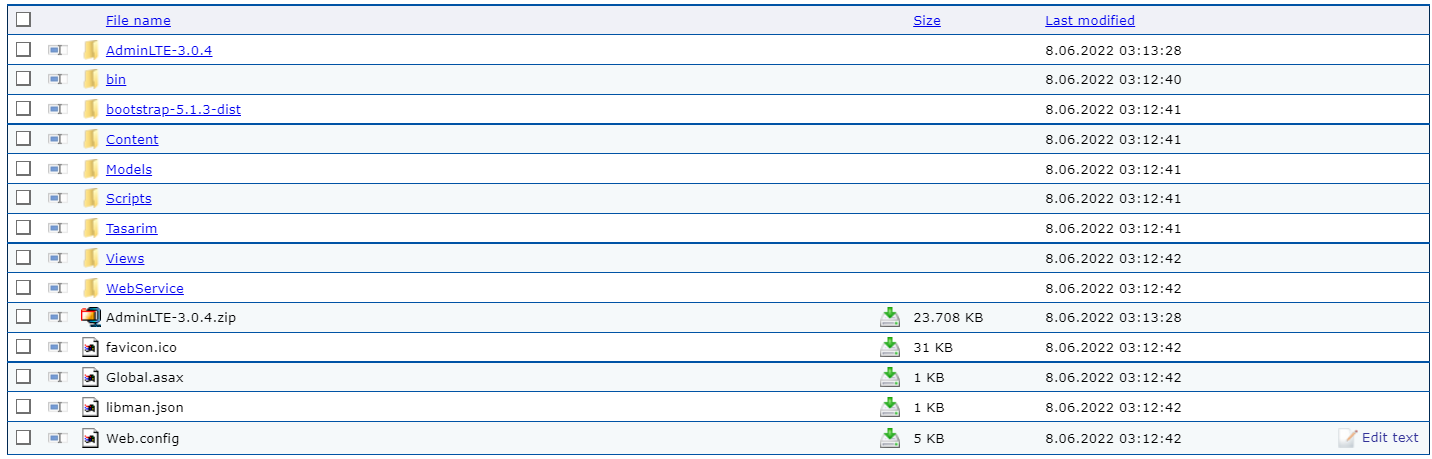
Şekil 6.1. Projeyi Publish Etme

Publish ettikten sonra oluşan dosyaları zip formatında arşive ekliyoruz.



Şekil 6.2. Proje Dosyalarını Zipleme

Dosyalar ziplendiğinde Somee’de “FileManager” kısmında ziplediğimiz dosyayı upload ediyoruz.

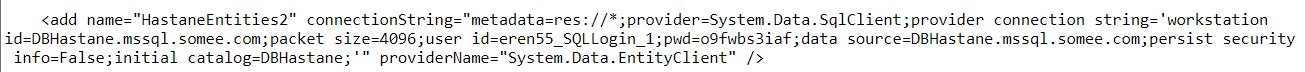


Web sayfamızı bu haliyle açmaya kalkarsak “customErrors” ve “ConnectionString” hatası alıyoruz. Bu hataları düzeltmek için “Web.config” dosyamızı “Edit text” ile açıp içerisine eksik kodları ekliyoruz.

**Eksik Kodlar:**

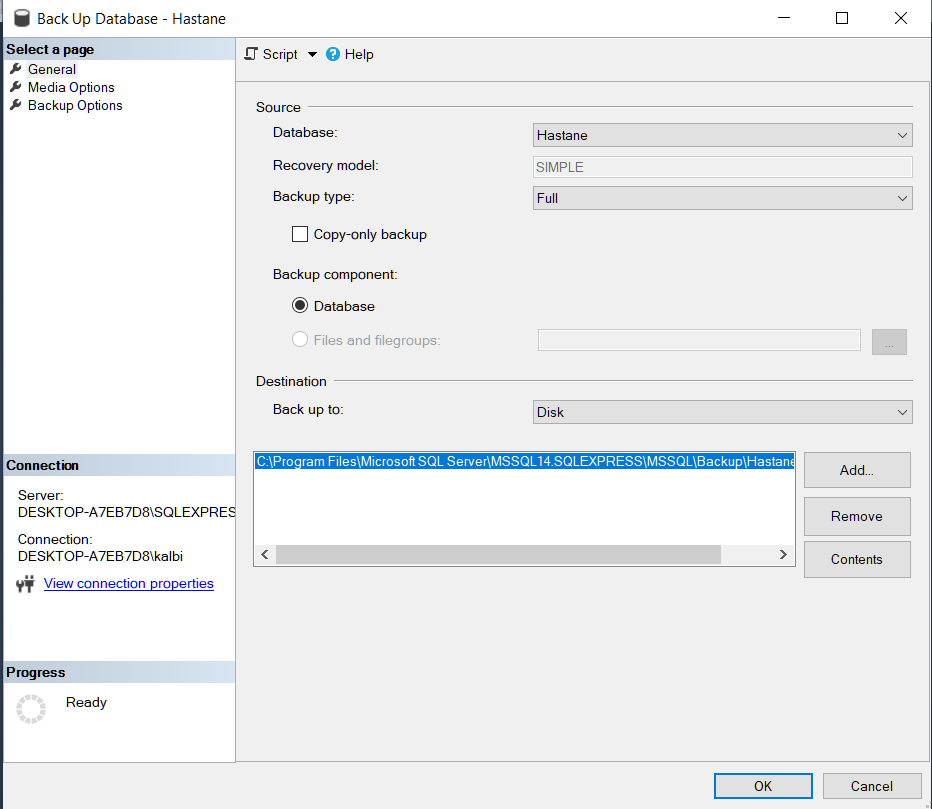
System.web’in içerisine <customErrors mode="Off"/> yazıyoruz.

connectionString’in içerisine;



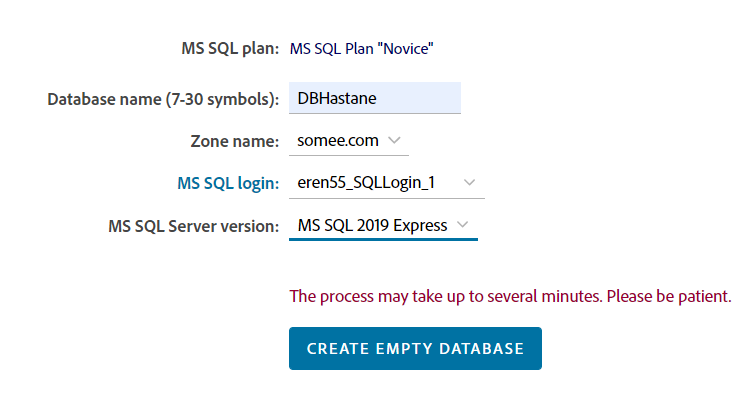
Yukarıdaki kodu yazıyoruz.

Veri tabanımızı sunucuya eklemek için öncelikle veritabanımızın backupını oluşturmamız gerekiyor. Backupı oluşturmak için veri tabanımıza sağ tıklayıp “task” sekmesinden “Back Up” seçeneğini seçiyoruz.



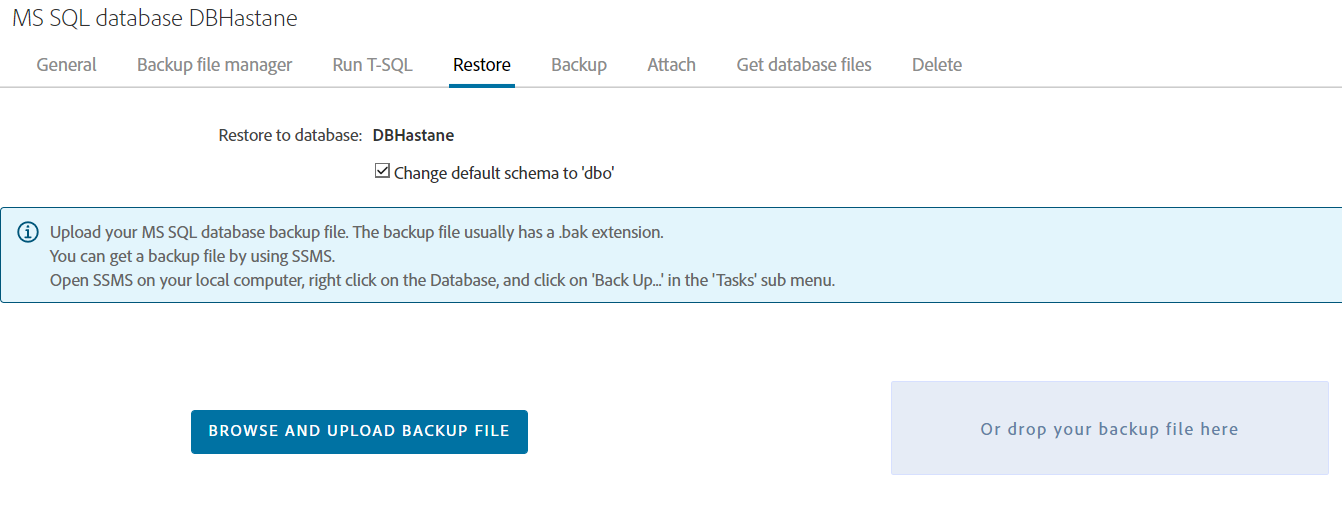
Şekil 6.3. Veri Tabanı Backup Oluşturma

“Ok” butonuna tıkladığımızda “.bak” uzantılı bir dosya elde ediyoruz. Somee’de “Database” sekmesinde boş bir veri tabanı oluşturuyoruz.

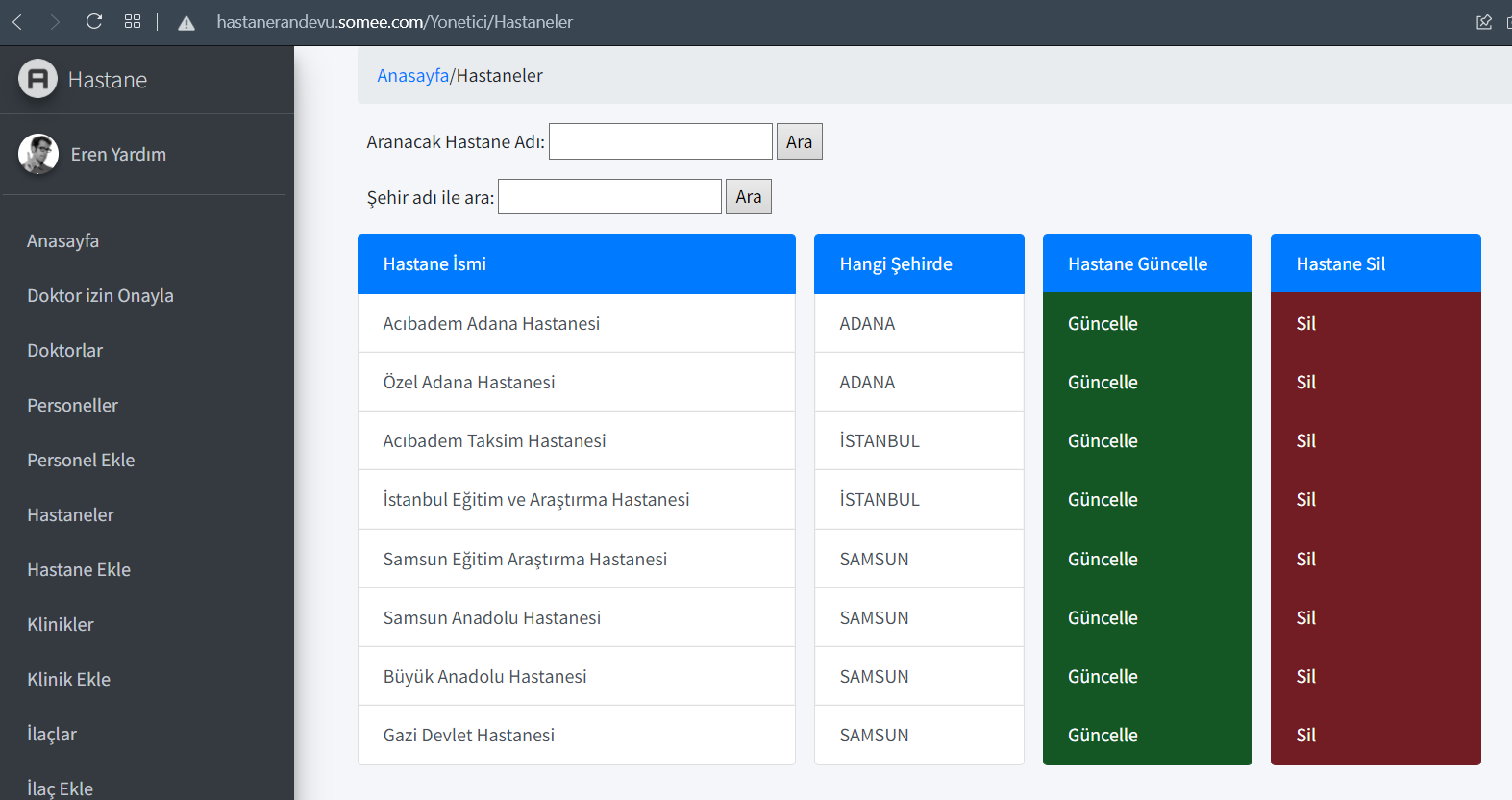


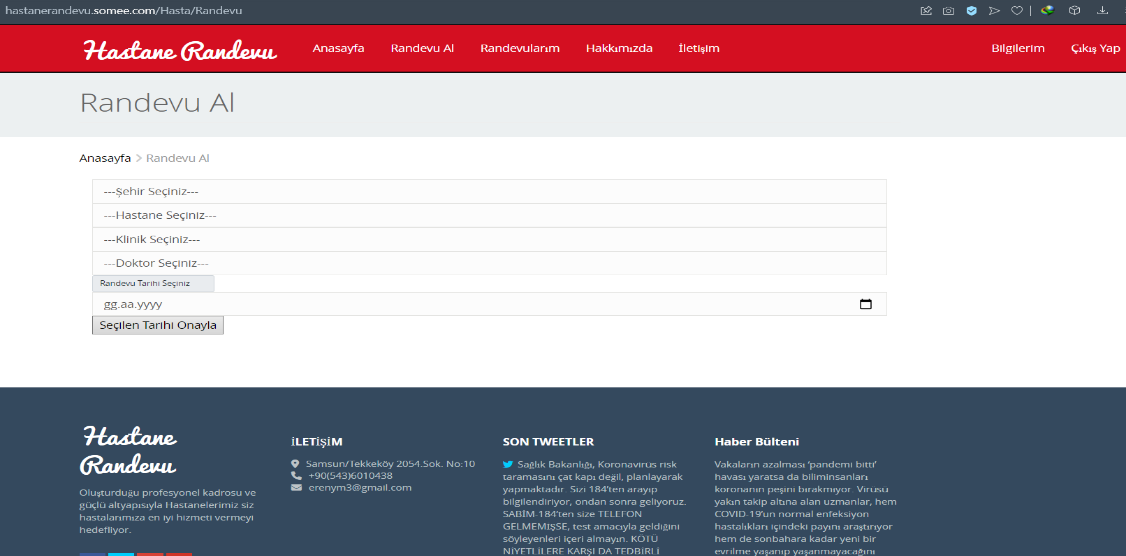
Şekil 6.4. Veri Tabanı Oluşturma

Oluşan bu veri tabanını doldurmak için veri tabanını restore et seçeneği ile önceden oluşturduğumuz “.bak” dosyasını upload ediyoruz.



Böylelikle web sunucumuz hazır hale geliyor.





Şekil 6.5. Sunucu Üzerinden Açılmış Sayfalar

# 7. SONUÇLAR VE ÖNERİLER

Projenin bu haliyle bir hasta kolay bir şekilde istediği hastaneden, istediği doktordan randevu alabilir. Doktor kendinden randevu alan hastaları görebilir ve vakti geldiğinde hastasını muayene edebilir.

Sistem üzerinden hastane, klinik ve personeller üzerinde CRUD işlemleri yapılabilir. Muayene sonucu hastaya verilen ilaç reçetesi eczane sistemine “Web Service” ile iletildiğinden hasta ilaçlarını eczaneden kolay bir şekilde alması sağlanmıştır.

Eczane sistemi üzerinde ilaç ve personel CRUD işlemleri yapılabilir. Hastaya verilecek ilaçlar ve hastanın ödeyeceği tutar otomatik şekilde ekrana getirilir.

**KAYNAKLAR**

1. <https://medium.com/@kdrcandogan/mvc-nedir-mvc-yaşam-döngüsü-life-cycle-8e124f24650c>
2. https://www.webtekno.com/microsoft-visual-studio-nedir-h92228.html
3. https://www.penta.com.tr/markalar/microsoft/urunler/microsoft-sql-server/microsoft-sql-server-nedir/
4. https://www.srdrylmz.com/razor-view-engine/
5. <https://birimler.dpu.edu.tr/app/views/panel/ckfinder/userfiles/95/files/TezYazisi.pdf>
6. <https://www.niobehosting.com/blog/mysql/>
7. https://www.mshowto.org/web-servis-nedir-nerelerde-kullanilir.html

**ÖZGEÇMİŞ**

Eren YARDIM 2001'de Samsun’da doğdu; ilk ve orta öğrenimini aynı şehirde tamamladı; Canik İMKB Anadolu Lisesi’nden mezun olduktan sonra 2019 yılında Kastamonu Üniversitesi Mühendislik Fakültesi Bilgisayar Mühendisliği Bölümü'ne girdi.

**ADRES BİLGİLERİ**

Adres: Çay Mahallesi

2054.Sokak, No. 10, D: 10

Tekkeköy / SAMSUN

Tel: (543) 6010438

E-posta: erenym3@gmail.com