**WEB TEKNOLOJİLERİ FİNAL PROJESİ RAPORU**

**TIBBİ TAKİP SİSTEMİ**

Bu proje, bir hastane yönetim sistemini simüle etmek amacıyla geliştirilmiş bir web uygulamasıdır. Uygulama, kullanıcıların doktor ve hasta bilgilerini yönetmelerine, giriş yapmalarına ve kaydolmalarına olanak tanıyan bir platform sağlar. Ayrıca, uygulamanın Docker ile kapsüllenmesi, yazılımın taşınabilirliğini artırarak herhangi bir ortamda kolayca çalıştırılmasını mümkün kılar.

**Proje Özellikleri**

**1. Giriş ve Kayıt**

**Uygulama, kullanıcıların sisteme giriş yapabilmesi veya kaydolabilmesi için bir kullanıcı doğrulama mekanizması içerir. Kullanıcı türleri aşağıdaki şekilde ayrılmıştır:**

* **Admin:**
  + **Tüm kullanıcıları yönetebilir ve görüntüleyebilir.**
  + **Doktor ve hasta ekleme-silme gibi yetkilere sahiptir.**
  + **Örnek Admin Kullanıcı Bilgileri:**
    - **Kullanıcı Adı: admin**
    - **Şifre: admin123**
* **Doctor:**
  + **Kendi hastalarını yönetebilir ve görüntüleyebilir.**
  + **Hastalarına özel hastalık bilgisi, ilaç reçetesi ve tedavi kayıtları ekleyebilir.**
* **Patient:**
  + **Kendi tedavi geçmişini ve reçetelerini görebilir.**
  + **Hastane doktorlarından online randevu alabilir.**

**2. Veritabanı Entegrasyonu**

Uygulama, tüm kullanıcı bilgilerini ve ilişkili verileri saklamak için **SQL Server** veritabanını kullanmaktadır. Veritabanı bağlantıları, modern ORM (Object Relational Mapping) teknolojilerinden biri olan **Entity Framework Core** kullanılarak gerçekleştirilmiştir. Bu sayede, veritabanı işlemleri basitleştirilmiş ve uygulama performansı optimize edilmiştir.

**3. Session Yönetimi**

Kullanıcıların giriş yaptıktan sonra oturum bilgileri, **Session** mekanizması kullanılarak saklanır. Oturum yönetimi sayesinde:

* Kullanıcı kimlik doğrulama bilgileri korunur.
* Kullanıcının türüne göre (Admin, Doctor, Patient) özel yetkilendirme ve yönlendirme yapılır.
* Güvenlik önlemleri artırılmıştır.

**4. API Destekli Yapı**

Uygulama, **API Controller** yapılarına sahiptir. Bu controller'lar, kullanıcı bilgileri ve işlem yönetimi gibi işlevleri API tabanlı bir yaklaşımla sunar.

* **Örnek API İşlemleri:**
  + Yeni kullanıcıların eklenmesi.
  + Kullanıcı verilerinin güncellenmesi veya silinmesi.
  + Doktorların hasta kayıtlarına erişimi ve veri ekleyebilmesi.

**5. Docker Entegrasyonu**

Uygulama, **Docker container** içinde çalışacak şekilde yapılandırılmıştır. Projede bulunan **Dockerfile**, uygulamanın tüm bağımlılıklarıyla birlikte bir container'a paketlenmesini sağlar. Docker sayesinde:

* Uygulama taşınabilir hale gelir ve herhangi bir ortamda kolayca çalıştırılabilir.
* Geliştirme, test ve üretim ortamları arasında tutarlılık sağlanır.

**Teknolojiler**

* **ASP.NET Core MVC:** Uygulama, modern web geliştirme standartlarını karşılayan **ASP.NET Core MVC** çatısı kullanılarak geliştirilmiştir. Bu yapı, **Model-View-Controller (MVC)** mimarisini esas alır ve temiz kod yazımı ile güçlü bir yapı sunar.
* **Entity Framework Core:** Uygulamanın veritabanı işlemleri için kullanılan bu ORM aracı, SQL Server veritabanına erişimi kolaylaştırır ve performansı artırır.
* **SQL Server:** Uygulamanın tüm veri yönetimi ve saklama işlemleri, güvenilir ve güçlü bir veritabanı olan **SQL Server** kullanılarak gerçekleştirilmiştir.
* **Docker:** Uygulamanın platform bağımsız bir şekilde çalışabilmesi ve taşınabilir olması için kullanılmıştır. Docker container'ları sayesinde uygulama, geliştirildiği ortamdan bağımsız olarak her yerde aynı şekilde çalışabilir.

**Projenin Teknik Detayları**

**Veritabanı Yapısı**

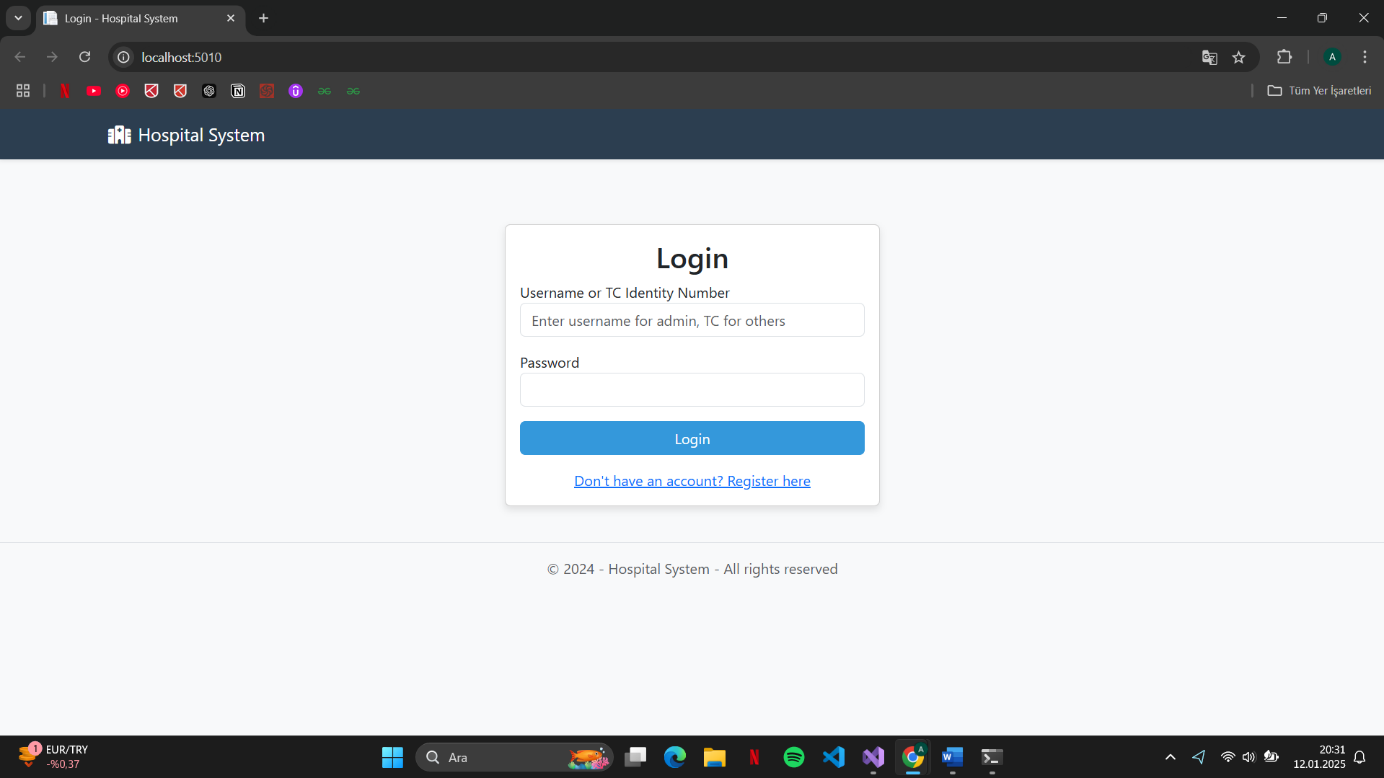
Veritabanı, hastane yönetim sisteminin ihtiyaçlarını karşılamak üzere aşağıdaki ana tabloları içerir:

1. **Admin:** Kullanıcı bilgilerini (ID, Ad, Soyad, Kullanıcı Türü, Şifre vb.) saklar.
2. **Doctors:** Doktorlara özel bilgileri (ID, Uzmanlık Alanı vb.) içerir.
3. **Patients:** Hastaların bilgilerini (ID, Doğum Tarihi, İlaç Bilgisi, Hastalık Bilgisi, Cinsiyet vb.) içerir.
4. **Branches:** Doktorların branşları tutulur.(Kardiyoloji, Dermatoloji vb.)

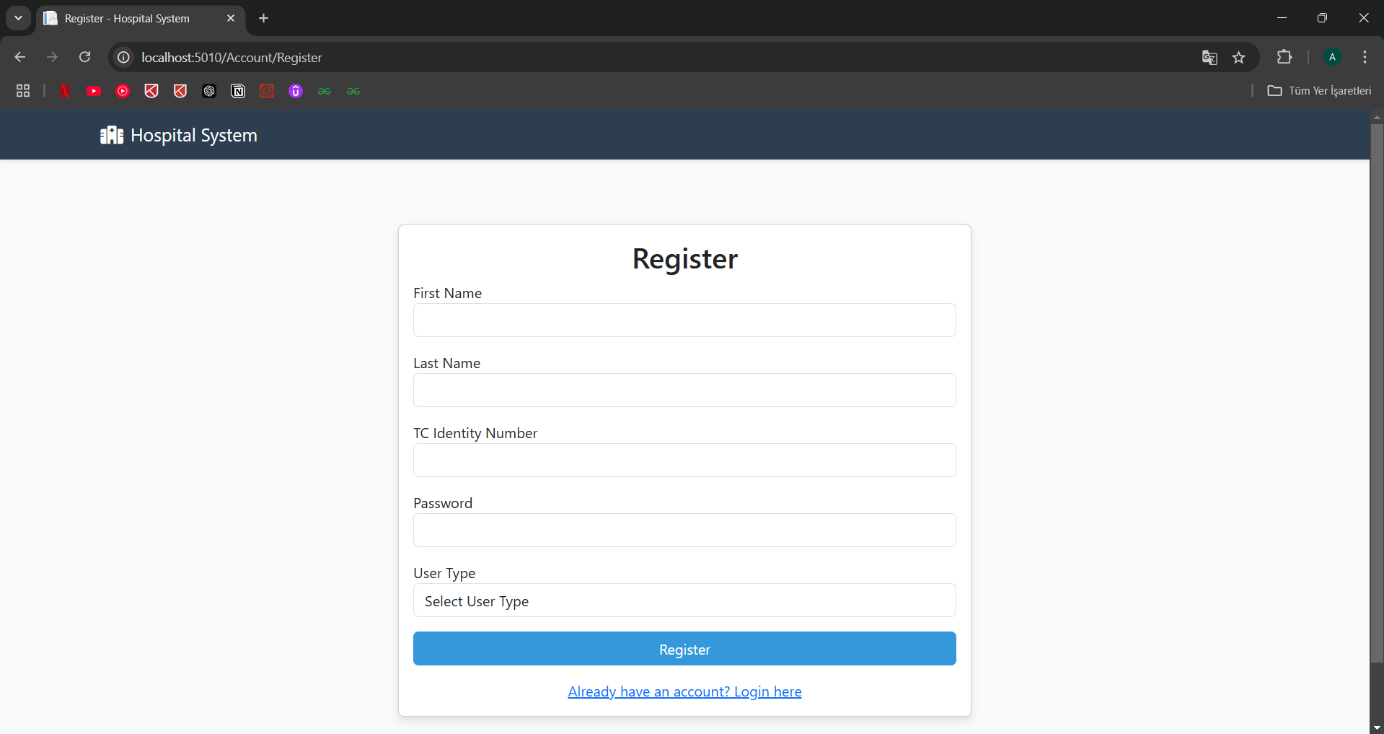
**Sonuç ve Değerlendirme**

Tıbbi Takip Sistemi, bir hastane yönetim sisteminin temel işlevlerini simüle etmek için tasarlanmış, kullanıcı dostu bir web uygulamasıdır. Proje, modern teknolojilerle geliştirilmiş ve taşınabilirliği artırmak için Docker entegrasyonu ile desteklenmiştir. Bu proje, hem kullanıcı dostu bir arayüz hem de güçlü bir altyapı ile, hastane yönetimi gibi karmaşık sistemlerin nasıl optimize edilebileceğini göstermektedir.

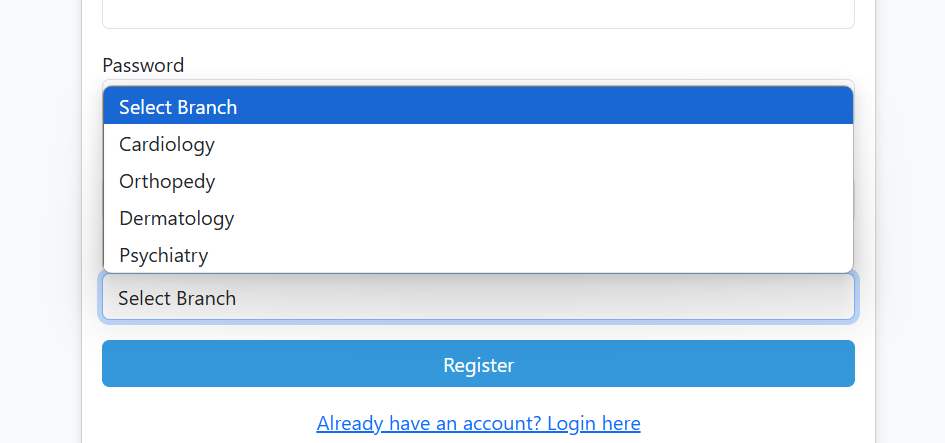
Aşağıda projenin kendisine ait ekran görüntüleri bulunmaktadır:



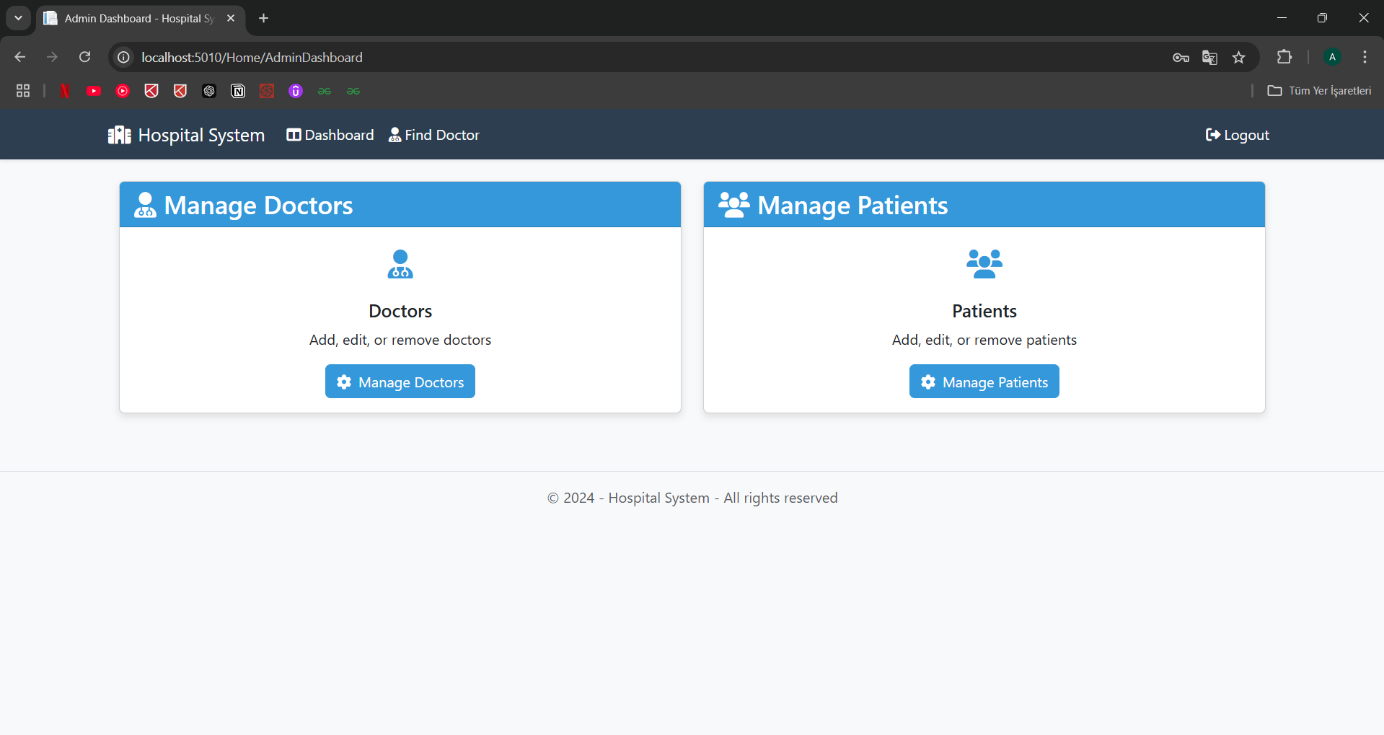
-Bu login sayfasına ait bir ekran görüntüsüdür. Kişi TC kimlik numarasıyla ve şifresiyle giriş yapabilir.



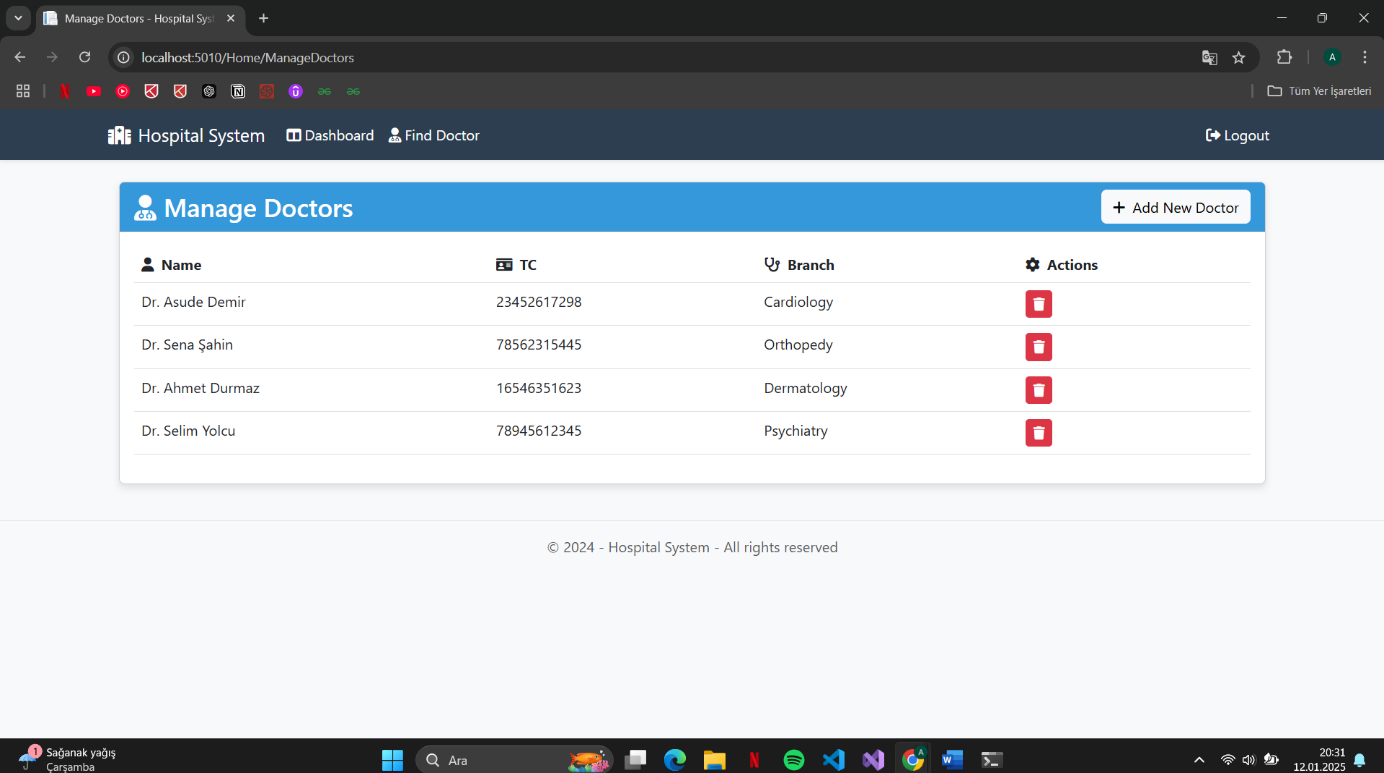
-Bu register sayfasına ait ekran görüntüsüdür. User Type kısmında Patient ve Doctor kısmı bulunmaktadır. Patient seçerse hastanın yaşı sorulmaktadır. Doctor seçilirse doktorun branşı sorulmaktadır.



-Bu şekilde branş seçebilir kayıt olan kişi.

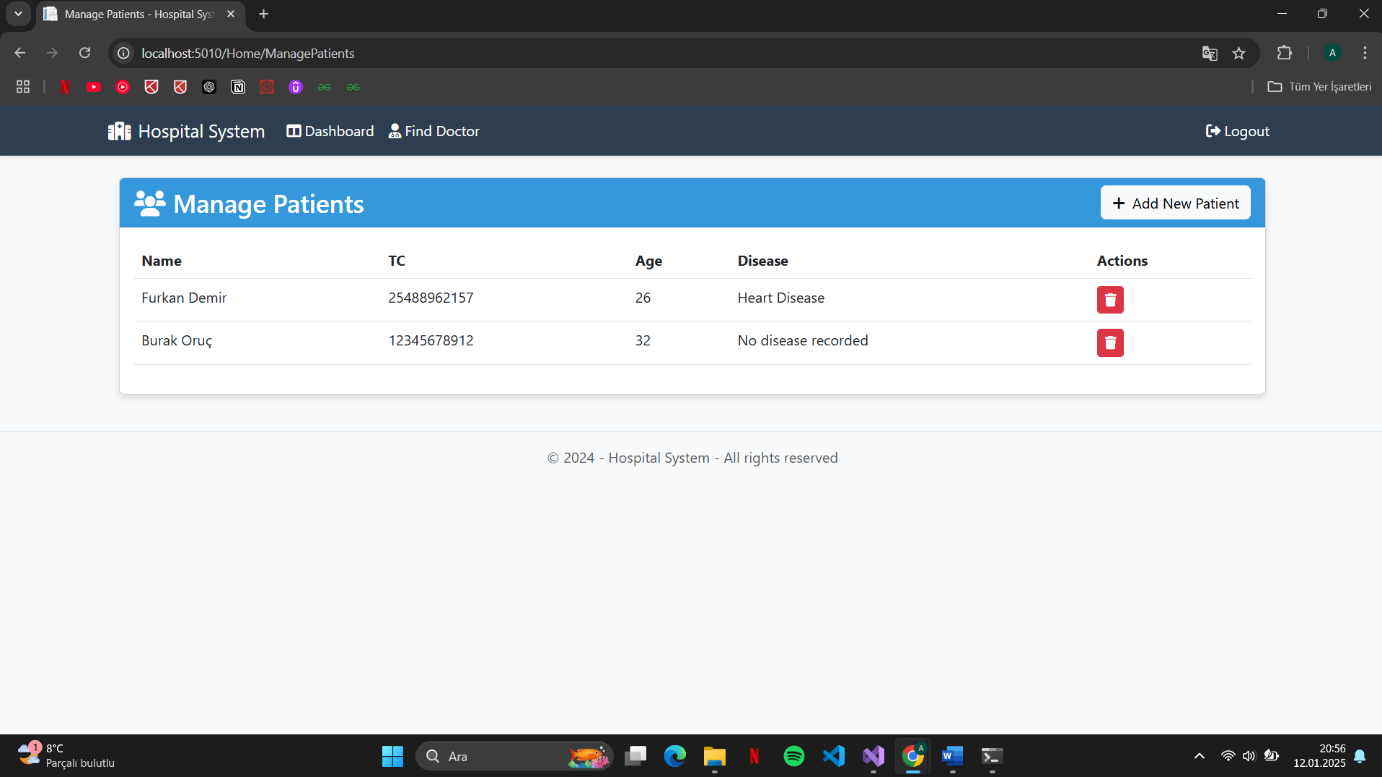


-Admin kısmıdır. Admin hem doktorları hem de hastaları görünteleme ve yönetme imkanına sahiptir. Manage Doctors seçilirse aşağıdaki gibidir;

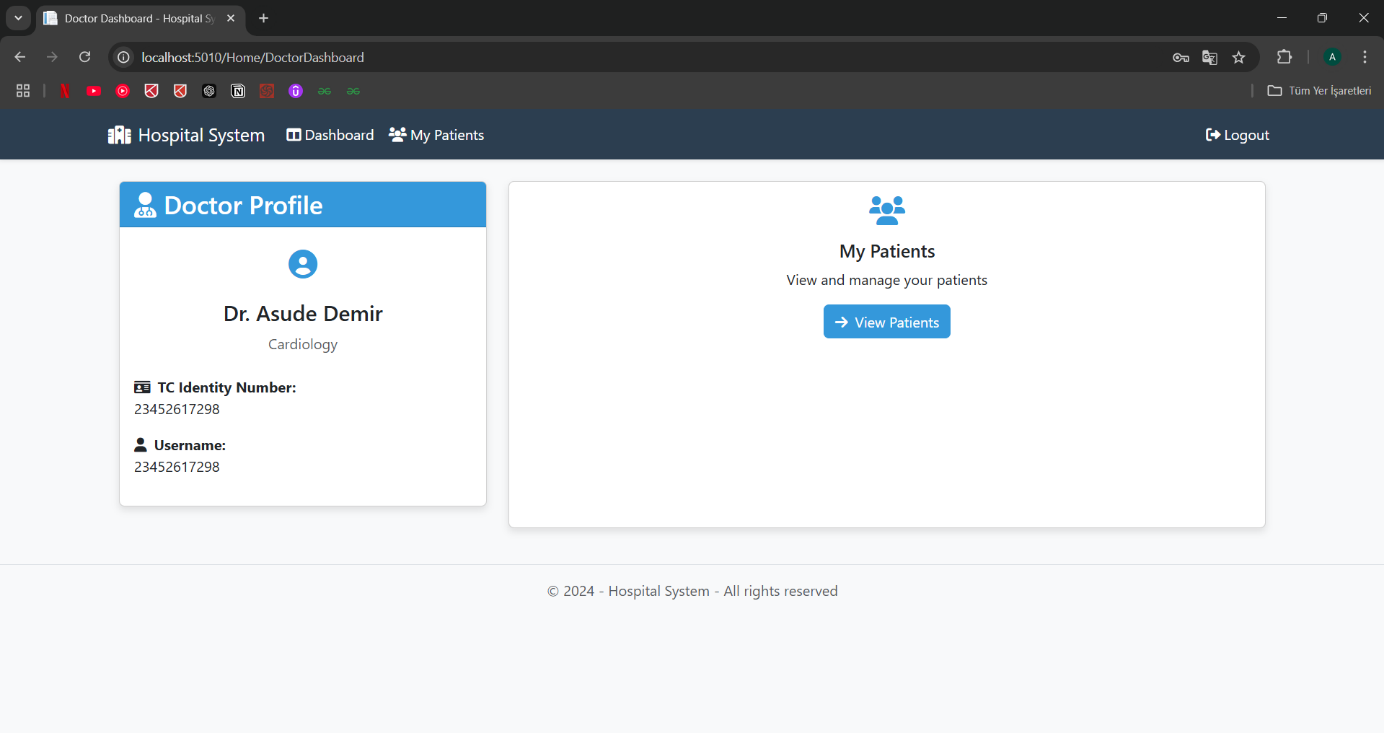


-Bu ekranda yeni doktor eklenebilir. Ve doktorlar silinebilir.

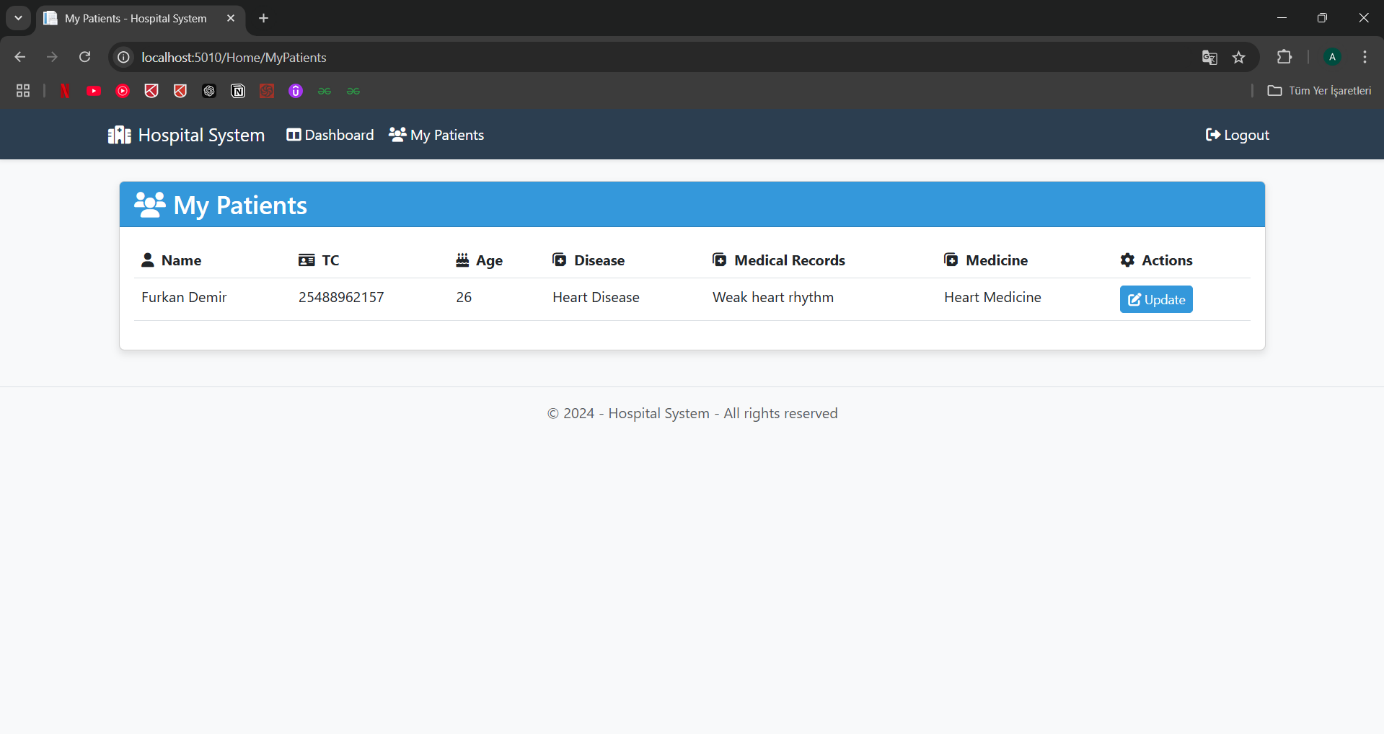
Manage Patients seçilseydi eğer;



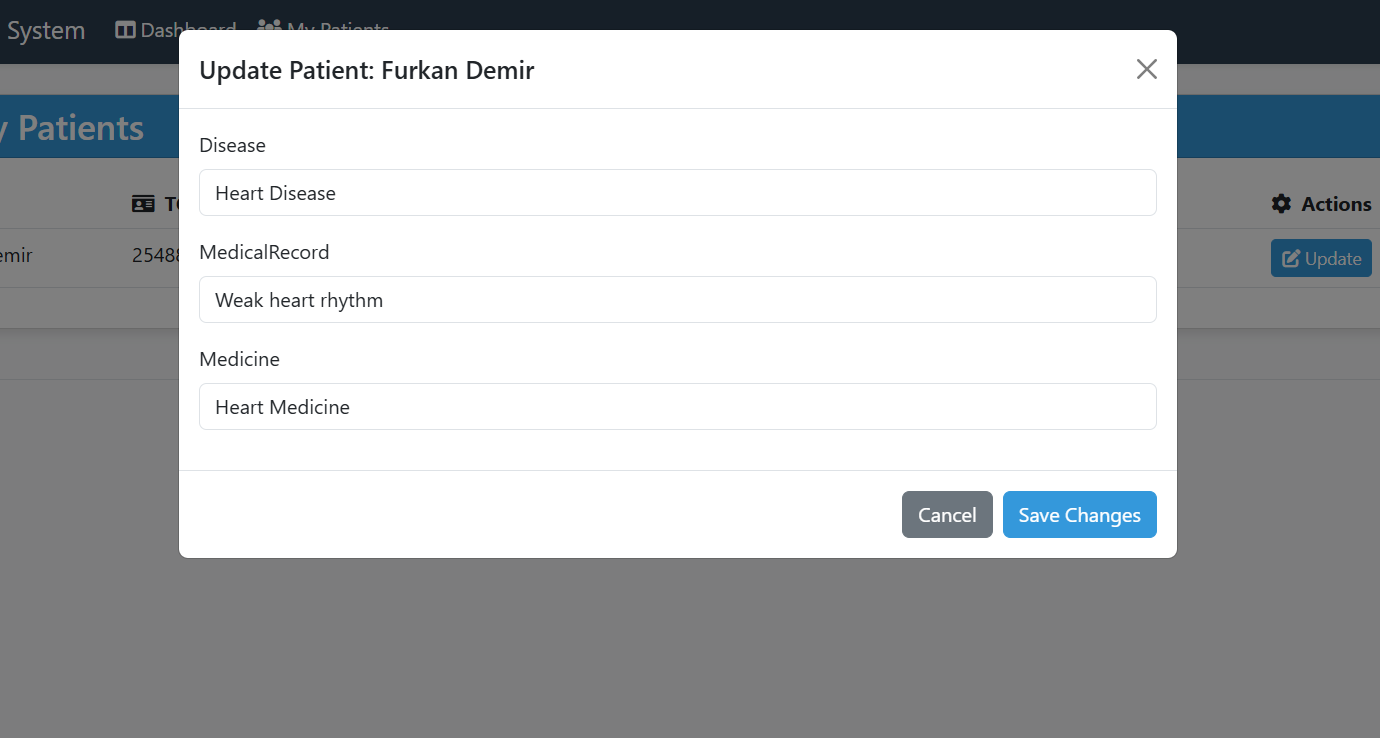
-Yeni hasta eklenebilir ve eski hastalara silinebilir.



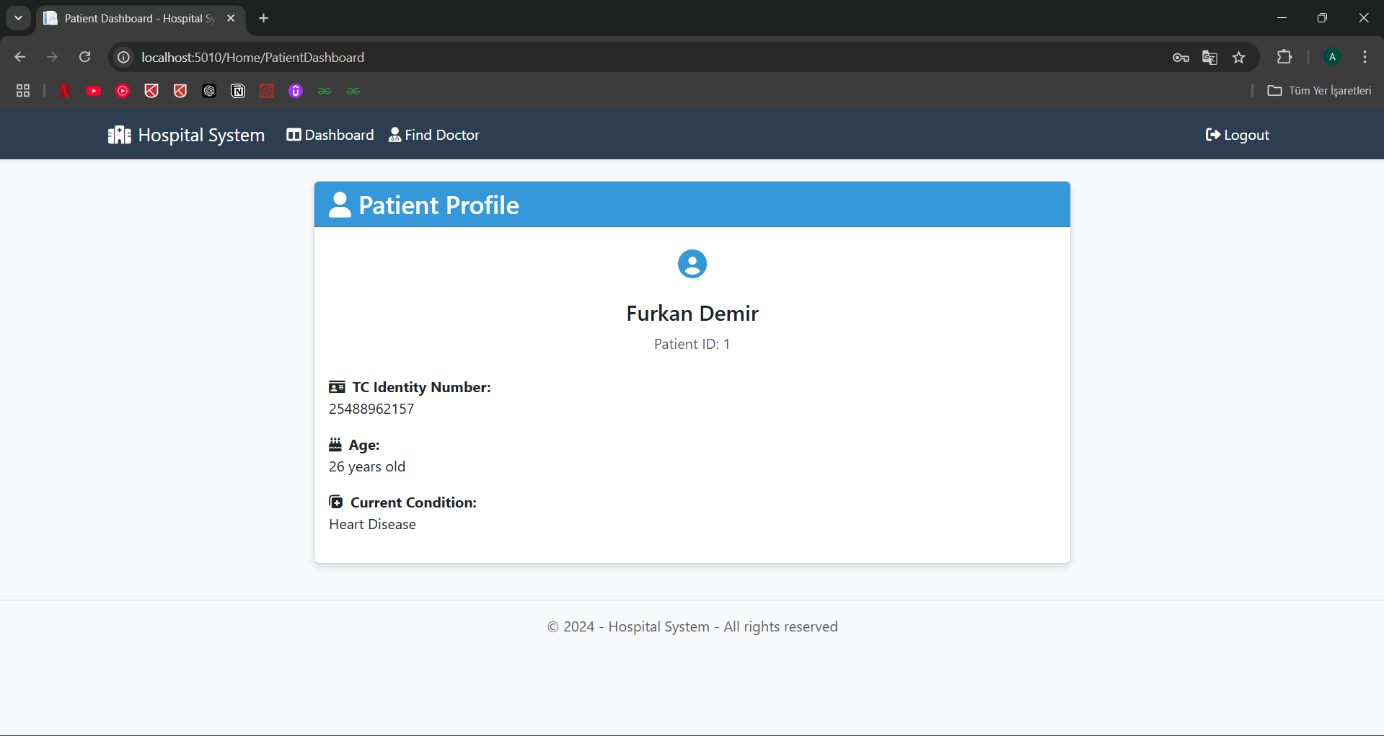
-Eğer bir doktor giriş yaptıysa doktorun sayfası bu şekildedir. Solda kendi kişisel bilgileri ve sağda da hastalarının bilgileri bulunmaktadır. View Patients dediğimizde ise;



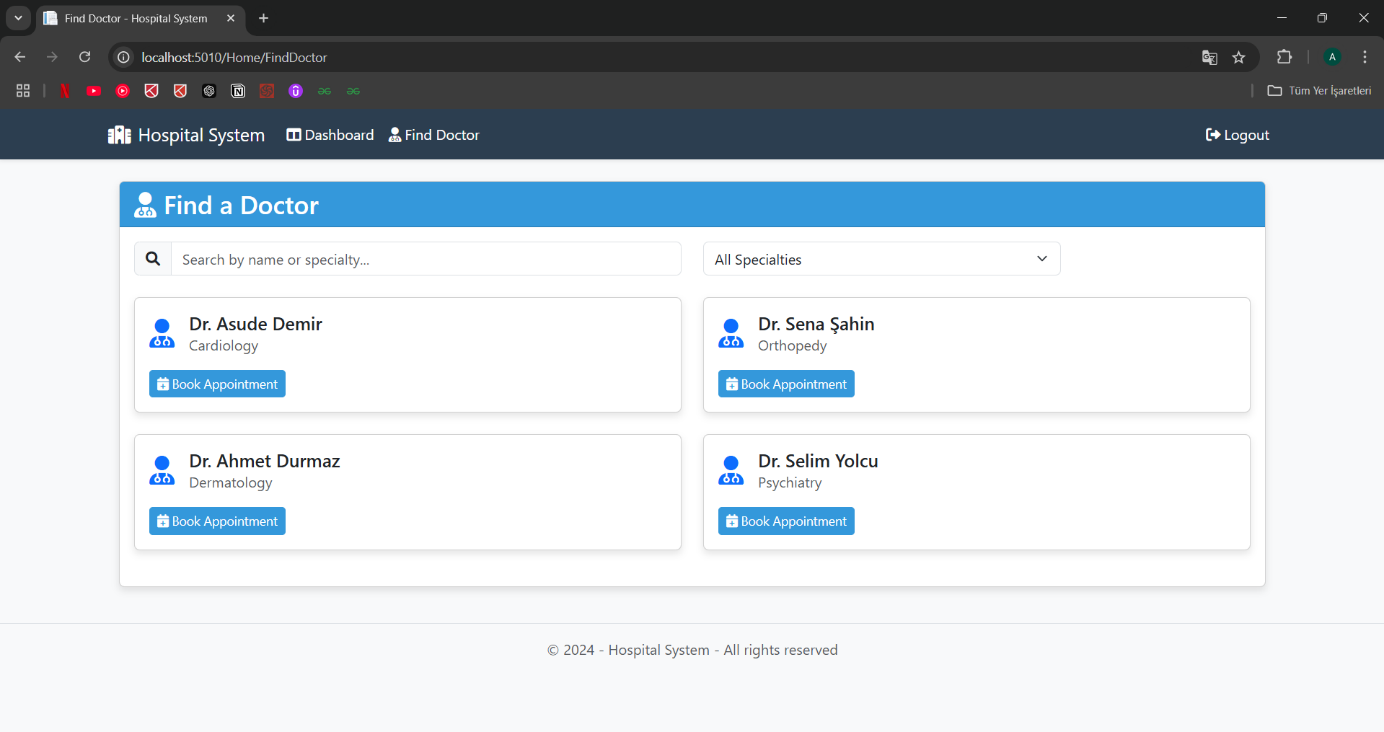
-Bu şekilde doktorun hastaları hakkındaki bilgilere ulaşabildiğini görmekteyiz. Ve doktor eğer isterse Update butonundan hastanın Disease, MedicalRecords ve Medicine kısmına yeni şeyler ekleyebilir ve eskisini silebilme yetkisine sahiptir. Update kısmı da aşağıdaki gibidir;



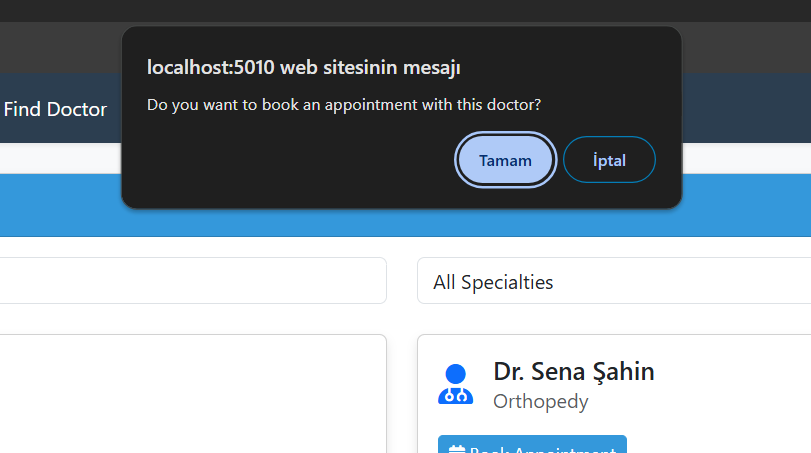
-Bu şekilde hastanın bilgilerini güncelleyebiliriz.



-Bu ekran görüntüsü de bir hastaya aittir. Bu şekilde kendi bilgilerini görebilir. Ve yukarıdaki Find Doctor’a tıklayınca da aşağıdaki şekilde gözükmektedir;



-Bu şekilde hastaneye ait doktorların bilgisi gelmektedir. İstenilen doktorun üzerindeki Book Appointment tuşuna basarak doktordan randevu alınır ve hasta doktorun sayfasına düşer.



Bu şekilde görülmektedir.

