**Yöntem:** Geliştireceğimiz proje kullanımı kolay, sade ve anlaşılır olmalı. Bunun sebebi ise engelliliğin topluma kazandırılmasındaki en büyük etkenlerden birisinin eğitim seviyesi ve yoksulluk olduğunu söyleyebiliriz. Bilişimden ve teknolojiden uzak olan ailelerin faydalanabilmesi için, projede sitemize ilk girdiği andan itibaren kolay bir şekilde kullanabileceği, anlayabileceği, sade bir tasarıma sahip ve bu sadelik içerisinde işlevsel bir tasarım oluşturmamız gerekmektedir. Bunun için bahsedilen teknolojilerden frontend kısmında anguları kullanacağız. Angular içerisidne site yapısı için HTML, güzel ve kullanışlı bir tasarım oluşturmak için CSS ve Primeng kullanacağız. Angular ise sahip olduğu yapı sayesinde sitemizi işlevsel ve etkileşimli hale getirecek.

Ön yüze veri sağlayacak olan mikroservisi ise Java springboot ile geliştireceğiz. Bize sağlayacağı güvenlik imkanları ve veri iletişim hızı sayesinde acil vakalarda açık vermeden verilerin iletimini gerçekleştirebiliriz. Yine springboot ile geliştirilen api sayesinde tam entegre bir sistem hasline gelen projemiz geliştirmeye ve değiştirilmeye açık olacaktır.

Verileri depolayacağımız ve veriler üzerinde işlem yapacağımız veri tabanımız ise ücretsiz olarak bütün ihtiyaçlarımızı karşılayan postgresql olacak. Barındırdığı sql dili ile yapmak istediğimiz işlevleri hızlı ve güvenli bir şekilde yerine getirecektir.

projemiz api sayesinde iletişim kuracak olup belirtilen yollar dışında gelen her isteği red edecektir. Belirtilen yollar ile iletişim sağlamaya çalışan kullanıcıların ise yetkileri kontrol edilerek yetkisiz yapılan istekler yine red edilecektir. Geliştireceğimiz sitede 3 adet kullanıcı olacak bunlar:

* Admin
* Engelli/Hasta
* Ziyaretçi

Her kulanıcı daha öncesinde yetkilendirilmiş bilgilere ulaşabilecektir. Kişisel verilerin korunması amacı ile kullannıcı giriş yapmadığı veriler umumi ortama açılmayacak ve açılan bilgiler ise yalnızca salt okunur yani değiştirilemez olacaktır. Her kullanıcının erişiebileceği servisler ve rolleri bellidir.

Admin: Yönetici yetkisi veri havuzumuza yeni hastalık veya engel ekleyebilir, veri havuzundaki eklenmiş olan engelleri veya hastalıkları güncelleyebilir veya silebilir. Hasta yada engelli vatandaşlarımızın eklediği hastalık veya engelleri kaldıramaz veya değiştiremez.

Engelli/Hasta yetkisi : Engelli veya hasta vatandaşlarımız kişisel bilgilerini ekleyebilir ve veri tabanındaki yönetici tarafından eklenmiş olan engelleri veya hastalıklardan sahip olduklarını kendisine tanımlayabilir. Diğer Engelli veya hastaların bilgilerini görüntüleyebilir ancak değiştiremez veya silemez. Veri tabanımıza yeni hastalık ekleyemez veya veri tabanında bulunan hastalık veya engelleri değiştiremez ya da silemez.

Ziyaretçi yetkisi : Ziyaretçiler qr kodu okutarak veya başka bir yöntem ile gönderdikleri url ile gönderilen hasta veya engelli idsine bağlı olan hastaların veya engellilerin verilerine ulaşabilir ve görüntüleyebilir. Engelli veya hastaların bilgilerini değiştiremez veya silemez. Veri tabanımıza yeni hastalık ekleyemez veya veri tabanında bulunan hastalık veya engelleri değiştiremez ya da silemez.

***YETKİLENDİRME FOTOĞRAFI***

Bütün kullanıcılar sitemizin giriş yapma ve kayıt olma sayfasına erişebilir. Ziyaretçiler kayıtol sayfasından tc kimlik numarası, isim soy isim, telefon numarası girerek ve şifre oluşturarak sitemize kayıt olabilir. Girilen bilgiler kayıt edilmeden önce veri tabanından sorgulatılarak kontrol edilir. Eğer aynı tc numarasına sahip başka bir kişi var ise kayıt işlemi iptal edilerek kullanıcıya hata mesajı gösterilir. kayıt olan kullanıcılar Engelli/Hasta yetkisi kazanır.

Giriş yap sayfasındaki gerekli bilgiler doldurularak sitemizin kullanıcı panellerine giriş yapabilir. Girilen her bilgi kontrol edilerek sahip olduğu yetki paneline aktarılır.

Proje Geliştirme aşamaları: Projemiz 5 Aşamada gerçekleştirilecektir. Bunlar sıralı olmayıp birbirileri ile etkileşimde olacaktır. Aşamalar şu şekildedir:

* Veri tabanının tasarlanması ve geliştirilmesi
* Backend tarafının tasarlanması ve geliştirilmesi
* Frontend tarafının tasarlanması ve geliştirilmesi
* Test aşaması
* Hata giderme aşaması

***Veri Tabanının Tasarlanması ve geliştirilmesi:***

Veri tabanımız postgreSQL tarafından sağlacak olup verilerimiz sql dili ile ilenecektir. Veri tabanında tablolar oluşturularak ilişkileri kurulacaktır.

Hasta Tablosu : Her hasta veya engellinin bilgilerinin bulunduğu hasta adında bir tablo oluşturacağız. Siteye kayıt olurken ve kayıt olduktan sonraki Hasta veya engelli vatandaşlarımızın girdiği veriler bu tabloya yazılacak ve işlenecek. Kayıt olan her hasta veya engelli vatandaşa bir adet id verilecek. Veri çekme, kayıt etme, silme ve güncelleme işlemleri bu id ile yapılacak. Tabloda bulunan veriler tipleri ile aşağıda belirtirlmiştir.

* İD
* İSİM
* SOYİSİM
* TC NO
* Telefon no
* Yakın telefon
* Yakın telefon2
* Doğum tarihi
* Cinsiyeti

Hastalık tablosu : Yalnızca yönetici tarafından işlenebilen diğer kullanıcılara salt okunur olan hastalık veya engellerin bulunduğu tablodur. Yönetici hastalık ekleme, silme, güncelleme gibi işlemlerini bu tablo üzerinde gerçekleştirir. Eklenen her hastalık veya engele bir id verilir ve bütün işlemler bu id ile yapılır. Hastalık tablosunun barındırdığı stünlar veri itpleri ile birlikte aşağıda verilmiştir.

* İd
* Hastalık\_adı
* Hastalık-tanımı
* Yapılması gerkeneler
* Yapılmaması gerekenler

Hasta-hastalık tablosu: Hasta-hastalık tablosu Hasta veya engelli kullanıcıların sahip olduğu hastalıkları veya engelleri kendilerine tanımladığı tablodur. Bu tablo 3 adet veri barındırır. Bunlar aşağıda verilmiştir :

* İd
* Hasta id
* Hastalık id

Hasta-hastalık tablosu hasta tablosu ile hastalık tablosunun id leri ile foregin key ilişkisi kurar. Her hasta veya engelli birden fazla hasta veya engele sahip olabilir bu neden ile hastalıkları hastaya tanımlamak için hastaların hastalıklarını tutan bir tabloya ihtiyacımız vardı. Burada bu gereksinimi hasta-hastalık tablosu karşılar. Aşağıdaki resim ile bu durum örneklendirilmiştir.

***HASTA HASTALIK ÖRNEĞİ***

Resimdede görüldüğü gibi 4,5 ve 6 numaralı id ye sahip olan satırların hasta id veriş kısmında 5 rakamı yazmakta bu 5 rakamı 5 numaralı id ye sahip olan hastamızı belirtmekte. 4,5 ve 6 rakamlı id ye sahip verilerin hastalık kısımları ise farklı hastalık id lerine sahip. Anlaşılacağı üzere 5 numaralı id ye sahip olan hastamız hastalık kısmında yazan idli hastalık veya engellere sahiptir.

Veri tabanının oluşturulmuş tabloları ve ilişkilendirmeleri aşağıdaki şekilde gösterilmiştir.

***VERİ TABANI İLİŞKİ GRAFİĞİ***

***Frontend tarafının tasarlanması ve geliştirilmesi :***

Projenin Arayüz tarfı Bahsedilen teknolojilerden olan angular ile geliştirildi.

Projede 7 adet sayfa bulunmakta. Bu sayfalar aşağıda maddeler halinde verilmiştir.

* Ana sayfa
* Giriş yap sayfası
* Kayıt ol sayfası
* Şifremi unuttum sayfası
* Yönetici paneli
* Kullanıcı paneli
* Hasta/Engelli bilgi sayfası

Bahsedilen sayfalar açıklamaları ile birlikte aşağıda verilmiştir.

Anasayfa : Anasayfa sitenin tanımını yapan siteye ilk ulaştığımız yerdir. Bu sayfada projenin tanıtımı yapılmakta. Hakkımızda kısmında projenin ne işe yaradığı faydaları ve özellikleri bulunmakta. Hizmetimiz kısmında ise kullanıcılara sunduğumuz hizmet şablonlar ile açıklanmakta. Sık Sorulan Sorular kısmında ise kullanıcıların merak ettiği ve sıkça sorulan sorular cevapları ile birlikte yer almakta. İletişim kısmında kullanıcıların geliştiricilere e posta göndererk ulaşabileceği bir alan bulunmakta. Bunun dışında footer kısmında iletişim bilgileri ve sayfa haritası bulunmakta.

Kayıt ol sayfası: Kayıt ol sayfası Ad , soyad, e posta, ve şifre bilgilerinizi girerek yeni bir hesap açabileceğiniz bir sayfadır. Buradan kayıt olarak hasta/engelli profili oluşturulur.

Giriş yap sayfası : giriş yap sayfası Kayıtlı kullanıcıların eposta ve şifreleri ile kullanıcı panellerine giriş yapacakları sayfadır. Bu sayfa bilgilerinizi kontrol ettikten sonra sizi yetkilenirerek kullanıcı panelinize yönlendirir.

Şifre yenile sayfası : Şişfre yenile sayfası email adresi ve telefon numaranızı alarak sistemden dığruluklarını kontrol eder. Eğer bilgiler doğrulanır ise sistem kayıtlı e postanıza güvenlikli yeni bir şifre gönderir.

Kullanıcı paneli : Kullanıcı paneli Kullanıcıların bilgilerinin görünütlendiği değiştirilebildiği ve silinebildiği sayfadır. Bu sayfa sayesinde bilgilerinizi ekleyebilir değiştirebilir ve silebilirsiniz. Qr kod oluşturucusu ile qr kodunuzu indirebilirsiniz. Şifre değiştir kısmından eski şifrenizi doğrulayarak şifrenizi yenisi ile değiştirebilirsiniz. (MADDELER HALİNDE YAZILABLİİR). Tanımlı hastalıklar kısmında tanımlı hastalıklarını görüntüleyebilir ve silebilirsiniz. Hastalıklar tablosundan ise kendinize hastalık tanımlayabilirsiniz. Hastalık tablosundan tanımlı hastalıklar tablosuna tanımladığımız hastalıklar artık hastalıklar tablosunda görüntülenmeyeceğinden bir hastalık birden fazla kez tanımlanamaz.

Engelli/ hasta bilgi sayfası : Engelli/hasta bilgi sayfası ile id si url de gönderilen hastanın kayıtlı bilgilerine ve üzerindeki tanımlı hastalıklara ulaşabilirsiniz. Tek bir tıklama ile 112 acil çağrı butonuna tıklayarak acil çağrı merkezini arayabilirsiniz.

Yönetici paneli: Yönetici paneli Yöneticinin hastalık ekleyip silebileceği ve güncelleme yapabileceği paneldir.

Angular projelerinde api servislerinden veri çekme işleminde projeyi parçalara ayırmamız gerekir. Bu parçalar genel olarak model, view ve service olarak adlandırılır. Her bir parça farklı bir amaca hizmet eder. Böylece tekrar kullanılabilir, geliştirmeye açık bir proje ve düzenli kod yazımı elde ederiz.

Model : Angularda Modeller veri aktarımında kullanılan nesnelerdir. Api den veri çekme işlemi yapılırken gönderdiğimiz istek sonucu dönen veriler angulara json veri formatında gelirken kullanım kolaylığı sebebi ile angular bu verileri gelecek veri demeti için oluşturulmuş modellere dönüştürür. Bu sayede cevap olarak gelen veriyi işlerken json formatı ile değil java nesneleri ile işleyebiliriz.

Projemizde veri tabanından, yönetici paneli için hastalık tablosunda bulunan gerekli veriler, kullanıcı panelinde görüntülenecek olan hasta hastalık tablosundaki hastaya ait hastalık bilgileri, hasta tablosundaki hasta bilgileri, hastalık tablosundaki hastalık bilgileri ve şifre değiştirmek için ise kullanıcı taBLOSUNDAKİ ŞİFRE BİLGİLERİNE İHTİYAÇ VARDIR. Gerektiği yerlerde bu bilgiler sunucudan istenerek işlenecektir. Bu neden ile yanıt olarak gelen bu bilgileri alabilmek için yazılan her servisin yanıtında gelecek olan bilgileri modele çevirmemiz gerekmektedir.

***MODEL RESMİ***

Yukarıdaki resimde görüldüğü üzere json veri formatındaki gelen verilerin java nesnesine çerliebilmesi için isimlerinin aynı olması ve veri tiplerinin uyumlu olması gerekmektedir.

Uygulama sayfalarında gösterilmek üzere hastalık bilgileri için istek yolladığımız zaman bize dönecek olan hastalık verileri için HastalikDTO adında bir model tanımlıyoruz.

Servisler : Servisler api iletişiminde sunuculara istek gönderdiğimiz katmandır. Hangi sunucuya ne istek gönderileceğini bu katmanda belirleriz. Gönderilen isteklerin sonuçları yine bu katmanda yakalanır.

View : View katmanı elde ettiğimiz verilerin görüntüleme işleminin yapıldığı katmandır. View katmanında görüntülemek istediğimiz veriler için html css gibi teknolojileri kullanarak sayfalar oluştururz ve model ile gelen verileri anguların data binding özelliği ile gösterilmesini istediğimiz kısımlarda kullanıcıya sunabiliriz.

Bir api ile iletişime geçmek istenildiği zaman servis ile isteğk gönderilir. gelen cevap hazırlanmış olan modele dönüştürülür ve view katmanına gönderilerek kullanıcıya sunulması sağlanır.

Sunucu tarafından gelen isteğin kimlik doğrulaması yapılarak yetkisi kontrol edilir. Eğer iki aşamadanda geçer ise istek işleme alınır. İstek yapılırken bu aşamaları doğrulamak için http isteğinin header kısmında token gönderilir. Gönderdiğimiz token kimlik ve yetki bilgilerimizi içerir. Token gönderme işlemi her istekte bulunmalıdır. Gönderilen her isteğin içerisine token ekleme işlemi fazladan iş gücü olacağından dolayı anguların interceptor özelliğini kullanırız.

İnterceptor istemci ile sunucu arasında oluşan iletişimi kontrol eden bir yapıdır. İstemci ile sunucu iletişimin tam ortasında yer alır ve gelip giden bütün istek ve yanıtlar interceptor kontrolünde sağlanır. İnterceptor ile istemci ve sunucu arasındaki letişimde araya girerek gerekli kayıt, yönlendirme değer ekleme gibi işlemleri tek bir yapı ile sağlayabiliriz.

***INTERCEPTOR RESİM***

Yapılan isteklerde kimlik doğrulaması ve yetki kontrolü için göndereceğimiz header öğesini ınterceptor yapısına tanımlıyoruz böylece her seferinde göndermemiz gereken header öğelerini tek seferlik bir tanımlama ile her istek ile birlikte gönderebiliriz.

***INTERCEPTOR TANIMLAMASI***

Kullanıcı sayfa ile etkileşiminde api servisi ile iletişim kurulacağı zaman geliştirilmiş olan servis metotlarından tanımlı olan metot çağrılır. Bu metotun çağrılması ile service metodu kendisine verilen adrese http protokolü ile istek gönderir

***SERVİS KULLANIM ÖRNEĞİ***

Interceptor gönderilmek istenilen bu http isteğini yakalar ve içerisine kullanıcının jwt tokenını http isteğinin header kısmına ekler. Böylece sunucunun yapcağı kimlik doğrulaması ve yetki kontrolünden geçerek isteğimiz işleme alınabilr