

FitLife - Online Fitness Koçluğu Platformu Projesi

Eren GÜNEŞ

200202026

Kocaeli Üniversitesi Bilgisayar Mühendisliği (İÖ)

gunes18_eren@hotmail.com

Yusuf GÖKSU

210202027

Kocaeli Üniversitesi Bilgisayar Mühendisliği (İÖ)

goxuyusuf@gmail.com

I. ÖZET

Rapor Yazılım Laboratuvarı I dersinin 2.Projesi'nin nasıl yapıldığını anlatmak ve içeriğin açıklanması için yazılmıştır. Bu rapor içerisinde Giriş başlığı altında Yönetici Paneli, Öğretmen Paneli, Öğrenci Paneli, Veri tabanı olmak üzere 4 adet alt başlık bulunmaktadır. Bu başlıklar ile proje kısaca anlatılamaya çalışılmıştır. Daha sonrasında Yöntem başlığı altında projenin kodunun nasıl yazıldığından bahsedilip Deneyisel Sonuç başlığında ise yapılan program kullanıcının ve yöneticinin gözünden nasıl gözükeceği anlatılmıştır. Kazanımlar bölümünde proje yapılırken projenin bize katkılarından bahsedilmiştir. Sonuç kısmında ise proje kapsamında yapılan adımların sonuçları anlatılmıştır. Raporun sonundaysa proje yapılırken kullanılan kaynaklara, kaynakçalara yer verilmiştir.

II. GİRİŞ

Bu rapor belgesi 2023-2024 Bahar dönemi BLM307 Yazılım Laboratuvarı I dersinin 2. projesini açıklamaya yönelik bilgiler içermektedir. Projede programlama dili olarak ASP.Net, Java Spring, React ve Python dillerine izin verilmiştir. Bu proje kapsamında bizden online olarak fitness antrenörlerinde hizmet alabilecekleri bir web uygulaması yapılması istenmiştir. Uygulama yapılırken bir bulut veritabanı kullanılması istenmiştir. Proje bireylerin kişisel fitness hedeflerine ulaşmalarına yönelik özel antrenman programları, beslenme planları ve etkileşim seçenekleri sunmalıdır. Kullanıcılar, kaydolduktan sonra antrenörlerle iletişim kurabilir, özel antrenman programları ve beslenme planları alabilir, ilerlemelerini takip edebilir ve çevrimiçi dersleri izleme imkanına sahip olabilirler. Bu proje, sağlıklı yaşam ve fitness hedeflerine ulaşmak isteyen bireylere özel rehberlik ve destek sunmayı amaçlamaktadır.

A. Kullanıcı Kayıt Ve Giriş

Kullanıcılar email ve şifre girişi yaparak giriş yapılabilir. Bir giriş ekranı tasarlanarak burada email ve şifre kontrolleri gerçekleştirilmelidir. Kullanıcılar kişisel bilgilerini (cinsiyet, telefon numarası vb.) girerek sisteme kayıt olabilmelidirler. Kullanıcılar eğer şifrelerini unuturlarsa şifrelerini değiştirebilecekleri şifre unuttum seçeneği olmalıdır. Uygulamada üç adet rol yer almalıdır. Bu roller : Admin, Danışan ve Antrenör olmalıdır.

B. Antrenör Rolü

Antrenörler sisteme giriş yaparak profil bilgilerini görüntüleyebilmeli. Aynı zamanda profil bilgilerini değiştirebilmelidir. Bulunması gereken profil bilgileri; ad, soyad, uzmanlık alanları (kas kazanma, kilo koruma, kilo aldırma, kilo verdirme), iş deneyimleri ve iletişim bilgileri gibi bilgiler bulunmalıdır.

Antrenörlere danışanlar atanmalıdır. Antrenör kendisine atanan danışanların kişisel bilgilerine ve programlarındaki ilerlemeleri görebilmelidir.

Antrenör danışanlara beslenme ve antrenman programlara yapabilmelidir.

Beslenme programları hedef (kilo alma, kilo verme, kiloyu koruma, kas kazanma), günlük öğün ve kalori hedefi içermelidir. Antrenör beslenme programı oluşturabileceği gibi beslenme programlarında değişiklikte yapabilmelidir.

Antrenman programları egzersiz adı, hedefleri (kilo alma, kilo verme, kiloyu koruma, kas kazanma), set ve tekrar sayıları, video rehberleri, programa başlama tarihini ve programın süresini içermelidir. Antrenör antrenman programı oluşturabileceği gibi beslenme programlarında değişiklikte yapabilmelidir.

Web sitesinde sohbet kısmı olmalıdır. Burada antrenör danışanlar ile mesajlaşabilmelidir.

Bir antrenör başka bir antrenörün işlemlerini görmemeli ve müdahale edememelidir.

C. Danışan

Danışanlar sisteme giriş yaparak profil bilgilerini görüntüleyebilmeli. Aynı zamanda profil bilgilerini değiştirebilmelidir. Bulunması gereken profil bilgileri; ad, soyad ve iletişim bilgileri gibi bilgiler bulunmalıdır.

Danışanlar antrenörlerin onlara atadığı beslenme ve antrenman programlarını görüntüleyebilmelidir.

Danışanlar beslenme ve antrenman programları sonucunda ilerlemelerini sisteme kayıt edebilmelidir. İlerleme kayıtları; kilo, boy, vücut yağ oranı, kas kütlesi, vücut kitle indeksi gibi verileri içermelidir.

Danışanlar kaydettikleri ilerlemeleri bir görsel tablo ile görebilmelidir.

Bir danışan başka bir danışanın işlemlerini görmemeli ve müdahale edememelidir.

Web sitesinde sohbet kısmı olmalıdır. Burada danışanlar antrenörlerle ile mesajlaşabilmelidir.

D. Admin

Admin sistemdeki bütün kullanıcıların verilerini görüntüleyip düzeltebilmelidir.

Admin yeni bir kullanıcı oluşturabilmelidir.

Kullanıcıların hesaplarını aktif ve inaktif edebilmelidir.

Admin bütün kullanıcıların hesaplarına erişerek sanki o kullanıcı giriş yapmış gibi sistemde işlem yapabilmelidir.

E. Database

Projede bulut veritabanı hizmeti veren bir veritabanı kullanılmaktadır.

Veritabanı ilişkisel olmalıdır. Database şu tabloları içermelidir : Kullanıcı, Antrenör, Antrenman Programları, Beslenme Planları, Mesajlaşma, Kullanıcı-Antrenör İlişki. Database normalizasyon edilmiş bir şekilde oluşturulmalıdır.

III. YÖNTEM

A. Database

Projemizde bulut veritabanı olarak firebase kullandık. Firebase diğer databaselerden farklı olarak tablo ve elemanlar yerine collection ve documentation içeriyor. İlişkisel bir veritabanı olabilmesi için collectionların documentationlarındaki attributeler ile diğer documentationlardaki attributeler birbirleri ile ilişkilendirilmiştir.

Veritabanımızı 7 collection ile oluşturduk. Collection içerikleri aşağıda anlatılmıştır.

- 1) Admin : Admini bilgilerinin tutulduğu collection bu collectiondır. Burada adminin uid si rolü ve diğer kişisel bilgileri tutulur. Adminin uid attributesi ile documentationun uid si birbiri ile eşleştirilmiştir. Bu id ler authocation daki kullanıcı id leri ile aynıdır.
- 2) Danışan : Bu tablo Danışan bilgilerinin tutulduğu tablodur. Danışanın uid si , kas, yağ oranları dışında kişisel diğer bilgileride burada bulunur. Danışanın uid attributesi ile documentationun uid si birbiri ile eşleştirilmiştir. Bu id ler authocation daki kullanıcı id leri ile ilişkilendirilmiştir.
- 3) Antrenör : Bu tablo Antrenör bilgilerinin tutulduğu tablodur. Antrenörün uid si ve kişisel bilgileri burada bulunur. Antrenörün uid attributesi ile documentationun uid si birbiri ile eşleştirilmiştir. Bu id ler authocation daki kullanıcı id leri ile ilişkilendirilmiştir.
- 4) KullanıcıAntrenör : Bu collection ile hangi danışan hangi antrenör ile ilişki içerisinde bunun bilgileri tutuluyor. Buradaki collectionda danışan uid si ile antrenör uid si tutuluyor. Bu uid ler antrenör ve danışan collectionundaki id ler ile ilişkilendirilmiştir.
- 5) AntrenmanProgram : Bu collectionda antrenman programları tutulmaktadır. Antrenman programı bilgilerinin dışında danışan id si tutularak hangi danışanın antrenman programı olduğu danışan tablosu

ile ilişkilendirilmiştir.

- 6) BeslenmeProgram : Bu collectionda beslenme programları tutulmaktadır. Beslenme programı bilgilerinin dışında danışan id si tutularak hangi danışanın beslenme programı olduğu danışan tablosu ile ilişkilendirilmiştir.
- 7) KullanıcıDurum : Danışanların ilerlemeleri bu collectionda tutuluyor. Boy, kas vb. özellikler burada bulunuyor. Ayrıca kullanıcının uid si ve ilerlemenin tarihide burada tutularak danışan tablosuyla ilişkilendirilmiştir.
- 8) Mesajlaşma : Danışanların ve antrenörlerin bütün mesajlaşmaları bu collectionda tutuluyor. göndericiId, alıcıId , mesaj içeriği ve zaman özellikleri burada bulunuyor. Gönderici ve alıcı id'leri primary keyimiz zaman ise mesajları geçmişten günümüze sıralamak için tuttuğumuz bir değişkendir.

B. Giriş Ekranı

Giriş ekranını main componentinde gerçekleştirdik. Ekranı iki tane input yerleştirdik ve bu input değerlerinin dictionary bir değişken tanımlayarak burada tuttuk. Ekranı birde buton koyduk. Bu butonun onclick içerisinde authFunc diye bir fonksiyon yazdık. Bu fonksiyonun içerisinde signInWithEmailAndPassword fonksiyonunu kullanarak authentication daki email password kaydının kontrolünü gerçekleştirdik. Kontrol eğer doğruysa databasede sırasıyla danışan, antrenör ve admin collectionlarında doc ve getdoc fonksiyonlarını kullanarak kullanıcının bilgilerini aradık. Kullanıcıyı bulduktan sonra rolüne bakarak ilgili adrese window.location diyerek yönlendirdik. Eğer kullanıcı yanlış giriş yaptıysa da ekrana erroru yazdırdık. Giriş ekranına ayrıca kayıt ol ve şifremi unuttum diye iki tane yazı ekledik ve bu yazıların onclickleri için ayrı fonksiyonlar yazdık. Bu Fonksiyonlar şifremi unuttum ve kayıt ol adreslerinde window.location ile yönlendirme gerçekleştiriyor.

C. Danışan

Danışan için özel bir navbar tasarladık ve bu navbarı sayfanın en soluna yerleştirdik. Navbar içerisinde iconları kullanarak ufak görseller ekledik. Link to kullanarak bu görsellere basılması durumunda ilgili işlemin sayfalarına yönlendirmeleri gerçekleştirdik.

Profil sayfasında öncelikle localstorage'den gelen kullanıcı bilgilerini çektik ve bu kullanıcı bilgilerini ekrana koyduğumuz inputlara vererek kullanıcının profiline girdiğinde inputlar içerisinde kendi bilgilerini görmesini sağladık. Sonrasında aşağıya bir buton koyduk ve bu butonun actionu için bir dbupdate fonksiyonu yazdık. Bu fonksiyon içerisinde inputların şuanki değerlerini kullanarak doc ve updatedoc fonksiyonları ile veritabanında güncelleme gerçekleştiriyor. Aynı zamanda localstorage'den var olan veriyi remove ile silip set ile tekrar yüklüyoruz.

İlerleme işlemlerinin yapılacağı sayfada iki tane card oluşturduk ve bu kartları altlı üstlü yerleştirdik. Birinci cardda kullanıcının kilo, boy vb. bilgilerini databaseden çeken yazdırdık. Diğer cardda ise yeni kayıtlar eklenebilmesi için inputlar ve buton koyarak butonun onclickine tabloya veri eklemesi yapan bir fonksiyon yazdık. Tabloyuda cardların yanında konumlandırdık. Ayrıca ekrana 5 tane buton yerleştirdik bu butonlardan key olarak grafik componentlerine yönlendirme işlemi window.location ile yaptık.

Grafikleri oluştururken rechart kütüphanesini kullandık. İlerleme işlemlerinden gönderdiğimiz grafikveriyi localstorage'den çekerek rechart kütüphanesi yardımı ile görselleştirdik.

Programlar sayfasında kullanıcının bilgilerini localstorage'den çekerek doc ve getdoc ile kullanıcının beslenme ve antrenman programı tablolarındaki bilgilerini alıp ekranda gösteriyoruz. Ekranda iki tane card oluşturduk ve bunları yan yana koyarak görselleştirdik.

D. Antrenör

Danışan için tanımladığımız Navbarı bu sefer antrenör için de eşsiz bir NavBarAntrenor tanımladık ekranın sol tarafında antrenörlerin kendileriyle ilgili gidebileceği iconlar mevcuttur. Bu iconları fontawesome kütüphanesinden çektik ve tek tek link to kullanark iconları basılabilir hale getirdik. Antrenör ilk açıldığında eğer sıfırdan kayıt oluyorsa antrenör; özellik seçme kısmına aktarılıyor, burada antrenör, antrenörlere özel deneyim, kilo aldırma, kilo verdirme, kilo koruma, kas kazanma özelliklerini kendisi belirleyebilir. Bu belirlemeleri yaptıktan sonra firebase üzerinde antrenörün bilgileri kaydedildi. Antrenör ana sayfamıza yönlendirdik. Burada bu componentimiz çalışmadan hemen önce useEffect özelliğimizi kullanarak firebase database'imizden KullaniciAntrenor tablosunun hepsini çekiyoruz ve içerisinde antrenörün keni id'si ile eşleşen tüm kayıtları ogrenciId isimli dizimize atıyoruz. böylece bize kayıtlı olan danışanların id'lerini tek bir dizide toplamayı başardık . Ondan sonra AntrenmanProgrami isimli tablomuzun tamamını çekiyoruz ve daha önceden elimizde olan öğrencilerin id'lerini kullanarak buradaki öğrencilerin antrenman programlarını elde etmiş oluyoruz ve bunu AntrenmanBilgileri isimli dizimize atıyoruz. Şu anda elimizde bizden eğitim alan tüm danışanların id'leri ve onların antrenman bilgileri mevcut. Danışan id'lerini kullanarak danışanların bütün bilgilerini OgrenciBilgi isimli const değişkenimize atıyoruz ve böylece elimizde bütün istediğimiz veriler olmuş oluyor. Antrenörden eğitim alan kişileri ekrana yazdırmak istiyoruz, bunu şekilli bir halde yapmak için accordion isimli bir css tasarladık, bu css o div'in açılıp kapanabilmesine yarıyor. Accordion diye bir const değişken tanımladık ve bu değişkeni her seferinde kontrol ediyoruz, toggleAccordion isimli fonksiyonumuza yönlendiriyoruz ve burada index sayımıza göre hangi div'in açılıp kapanacağını belirliyoruz, eğer birisi açılırsa diğeri kapanıyor aynıına tıklarsak içindeki veriler gözüküyor. ÖğrenciBilgi isimli dizimizi map etmemiz gerekiyor ki program otomatik olarak kaç tane kişi varsa onların accordion yapısını oluştursun. Her divde isim ve soyad gözüküyor

en sağdaki üçgen işaretleri ile div'imizin içeriğini açıp kapayabiliyoruz. İsmimizin yanında Antrenman bilgilerine bak yazan bir butonumuz var bu butona tıkladığımızda Antrenman bilgileri sayfasına yönlendirme işlemimizi yapıyoruz, Antrenör burada seçtiği danışanın antrenman bilgilerini görüntüleyip güncelleme işlemlerini yapabiliyor. Bu butonun sağında ise Beslenme bilgilerine bak yazılı butonumuz var bu butona tıkladığında seçilen danışanın beslenme bilgilerini görüntüleyip güncelleme işlemlerini antrenör yapabilmektedir. En sağdaki üçgene basılıp açılıp kapatabiliyoruz Aynı zamanda burada Boy tablosu değişim grafiğini, kilo tablosu değişim grafiğini, kas kütlesi değişim grafiğini , yağ oranı değişim grafiğini ve kitle indeksi değişim grafiğini görüntüleyebileceğimiz butonları tanımladık.

Sizden Eğitim Alan Kişiler		
Fatih Aksoy	Antrenman Bilgilerine Bak	Beslenme Bilgilerine Bak
Selda Bağcan	Antrenman Bilgilerine Bak	Beslenme Bilgilerine Bak
Sebastian Szymanski	Antrenman Bilgilerine Bak	Beslenme Bilgilerine Bak
Kerem Aktürkoglu	Antrenman Bilgilerine Bak	Beslenme Bilgilerine Bak
Mehmet Görmez	Antrenman Bilgilerine Bak	Beslenme Bilgilerine Bak

Fig. 1. Antrenor Sayfası

E. Admin

Admin sistemem giriş yaptığında karşısına çıkan sayfada bütün sisteme kayıtlı kullanıcıları görebilmesi için için danışan, antrenör ve admin tablolarını doc ve getdoc fonksiyonlarını kullanarak çektik. Daha sonrasında çektiğimiz verileri mapping kullanarak ekranda hep bir kişiyi bir card temsil edecek şekilde görselleştirdik. Her kartın içerisine o kullanıcıyı aktif ve inaktif etmek için bir buton yerleştirdik. Butonun onclicki için bir aktiflikdurumu fonksiyonunu yazdık. Fonksiyonun içerisinde butona basıldığı anki karttaki kullanıcının bilgilerini fonksiyona yolladık ve o kullanıcının bilgilerini update fonksiyonuna vererek aktifliğinin değişmesini sağladık. Aktiflik butonu dışında kullanıcının verilere erişebilmesi için birde cardlara giriş yap diye bir başlık yazdık. Bu başlığın onclicki için ise yonlendir diye bir fonksiyon yazdık. Bu fonksiyonun içerisinde yinde kullanıcının bilgilerini alıp sanki o kullanıcı sisteme girmiş gibi authentication işlemine soktuk. Bu şekilde admin o kullanıcının bütün işlemlerini gerçekleştirme yetkisi vermiş olduk. Ekstra olarak ana navbarımıza bir kontrol yerleştirdik. Ana navbarda storageden adminid yi kontrol ederek eğer admin girişi varsa navbara bir geri dön kısmı ekledik. Bu geri dönün onclickine de geridön fonksiyonunu yazdık. Bu fonksiyon içerisinde window.location ile adminin sayfasına yönlendirdik.

IV. DENEYSEL SONUÇLAR

Web sayfası açıldığında kullanıcıyı e-posta ve şifre girişi yapması için bir arayüz karşılar. Kullanıcı e-posta ve şifre girdikten sonra giriş butonuna basması ile giriş yapan kişinin

rolüne göre ilgili ekran açılır.

Kullanıcılar giriş ekranından şifremi unuttum ekranına geçiş yapabilir. Şifremi unuttuma basılması halinde yönlendirilen sayfa bir e-posta girdisi alır. Girilen e postaya şifre değiştirmek için bir onay maili gönderilir. Giriş ekranında aynı zamanda kayıt ol ekranınada geçiş yapılabilir. Kayıt ol a basılmasın halinde yönlendirilen sayfada kullanıcı bilgileri alınır ve kayıt işlemi gerçekleştirilir.

A. Antrenör

Antrenör girişi yapıldığı zaman bizi antrenör sayfası karşılıyor bu sayfada en üstte Sizden eğitim alan kişiler yazıyor ve burada üst üste bizden eğitim alan kişilerin isimleri ve soy isimleri yazmaktadır. Sağ tarafta üçgen mevcut bu üçgene basıldığında div açılıyor ve kişinin cinsiyeti, doğum tarihi, telefon numarası, boyu, kilosu, kas kütlesi, yağ oranı, kitle indeksi bilgilerini görüntüleyebiliyoruz. Burasının aşağısında da Boy tablosu değişim grafiğini, kilo tablosu değişim grafiğini, kas kütlesi değişim grafiğini, yağ oranı değişim grafiğini ve kitle indeksi değişim grafiğini görebilmek adına yeşil butonlar mevcuttur. Sonra bizi İsim soy isimlerin sağ tarafında Antrenman Bilgilerine Bak yazan mavi bir buton karşılıyor, bu butonun sağ tarafında ise Beslenme Bilgilerine Bak yazan sarı bir buton karşılıyor. Sol taraftaki antrenöre özel oluşturduğumuz navbarda sitemizin ana sayfasına gidebiliyoruz ve orada Fitness Sayfamıza Hoş geldiniz yazısı animasyonlu bir şekilde bizi karşılamaktadır. Profil iconuna tıkladığımızda ise antrenörün profiline erişebiliyoruz ve burada antrenörün bütün özelliklerini güncelleyip görüntüleyebiliyoruz. Son olarak dumbell iconuna tıkladığımızda ise bizi antrenörün ana sayfası olan antrenör sayfasına yönlendiriyor ve ilk başta söylediğim bizim danışanlarımızın bütün verilerine burada erişebiliyoruz.

B. Danışan

Danışan giriş yaptığı zaman ilk başta kullanıcıyı home sayfası karşılıyor. Bu sayfada kullanıcıyı sayfamız hoş geldiniz diye bir yazı karşılıyor. Sayfaların hepsinde 2 tane navbar bulunuyor. Bunlardan bir tanesi yukarıda kullanıcının her zaman çıkış yapabilmesi için bulunan bir navbar. Diğeri ise sol tarafta kullanıcının yapacağı işlemlerin seçimini yaptığı yer.

Kullanıcı eğer profil bilgilerim sayfasına girerse kullanıcı burada bütün profil bilgilerini görebiliyor. Profil bilgilerinde değişiklik yaparak sisteme kayıt edebiliyor.

Kullanıcı eğer programlar sayfasına girerse burada beslenme ve antrenman programlarını görebiliyor. Buradan antrenörünün ona atadığı programları takip edebiliyor. Aynı zamanda programlar için eklenen videolarada oradan ulaşabiliyor.

Kullanıcı ilerleme sayfasına girerse burada kullanıcı kendi kilo, boy, vücut yağ oranı, kas kütlesi ve vücut kitle indeksini görebiliyor. Bu bilgilerin yanındaki oklarla da

gelişim sağlayıp sağlamadığını takip edebiliyor. Kullanıcı bu sayfada ilerlemelerini ilerleme kaydet kısmındaki boşlukları doldurarak kaydedebiliyor. Önceki ilerlemeleri ve yeni kayıt ettiği ilerlemeleri bir tabloda kullanıcıya gösteriliyor. Aynı zamanda eğer kullanıcı bütün ilerleme kayıtlarını görselleştirilmiş bir tabloda görmek isterse ilgili özelliğin butonuna tıklayarak o özelliğinin görsel grafiğini görebiliyor.



İlerleme Kaydet	
Boy:	5
Kilo:	6
Vücut Yağ Oranı:	3
Kas Kütlesi:	2
Vücut Kitle İndeksi:	8
Tarih:	27.12.2023
Kaydet	

Fig. 2. İlerleme Sayfası

C. Admin

Admin giriş yaptığı zaman ilk başta home sayfası ile karşılaşır. Bu sayfada admini sayfamız hoş geldiniz diye bir yazı karşılıyor. Sayfaların hepsinde 2 tane navbar bulunuyor. Bunlardan bir tanesi yukarıda adminin her zaman çıkış yapabilmesi ve kullanıcılara giriş yaptıktan sonra geri dönebilmesi için bulunan bir navbar. Diğeri ise sol tarafta adminin yapacağı işlemlerin seçimini yaptığı yer.

Admin kullanıcıları görebildiği sayfaya girerse bu sayfada admin sisteme kayıtlı olan her kullanıcıyı görebiliyor. Burada her kullanıcıyı bir card içerisinde görüyor. Bu cardların içerisindeki giriş yap ile sanki o kullanıcıymış gibi sisteme

giriş yapabiliyor. Bu şekilde kullanıcın her işlemini değıştirme yetkisine sahip oluyor. Aynı zamanda kullanıcı cardlarının altındaki aktiflik butonuna basarak kullanıcıların sistemde aktif ve inaktif olmasını sağlayabiliyor.

Admin eğer profil bilgilerim sayfasına girerse burada bütün profil bilgilerini görebiliyor. Profil bilgilerinde değışiklik yaparak sisteme kayıt edebiliyor.

V. KATKILAR

İlk defa bir web projesi yaptığımız için bu proje kapsamında web sayfaları hakkında çok fazla bilgiye sahip olunmuştur. Bu proje bir bulut veritabanı (firebase) kurma , database işlemleri gerçekleştirme, tablolar arasında ilişkisel veriler kurma ve bunları uygulamaya geçirmemizdebize yardımcı olmuştur.

Projede React ve javascript kullandığımız için React ve javascript kullanımı öğrenilmiştir.

Reactta firebase'deki databasemizi bağlamayı ve veritabanındaki işlemlerin arayüzde gösterilmesi öğrenilmiştir. Proje bize yabancı kaynaklardan araştırma yapmayı ve bunları anlayıp kodsız olarak projeye entegre etmeyi öğretmiştir.

VI. SONUÇ

Bizden bu projede bir web sitesi yapılması istenmiştir. React, Javascript ile web sitelerinin nasıl oluşturulduğunu ve firebase (bulut veritabanı) ile beraber nasıl kullanıldığını bu proje ile öğrenmiş olduk.

Programın çalıştırılabilir olması için internete bağlı bir cihazda web sitemizi açmanız gerekir.

Bizim açımızdan eğlenceli ve öğretici bir proje olmuştur.

VII. KAYNAKÇA

- 1) <https://www.youtube.com/watch?v=INfBIYAk3Lwlist=PLIHume2cwmHeydP0GkOzSxJHT1ph1BrWj>
- 2) <https://firebase.google.com/docs/auth>
- 3) <https://stackoverflow.com/questions/72844990/clarification-about-using-firebase-and-react-js>
- 4) <https://javascript.plainenglish.io/firebase-authentication-with-firestore-database-78e6e4f348c6>
- 5) <https://www.udemy.com/course/programcilik-kursu/>
- 6) <https://www.geeksforgeeks.org/how-to-upload-image-and-preview-it-using-reactjs/>
- 7) <https://blog.logrocket.com/firebase-cloud-storage-firebase-v9-react/>