

PROJE 2

Egemen Çakır

Kocaeli Bilgisayar Mühendisliği
Email: cakiregemen0@gmail.com

I. GİRİŞ

Bu projede kullanıcıların online olarak fitness antrenörlerinden hizmet alabilecekleri bir web uygulaması yapılması beklenmektedir. Bu platform, kullanıcıların kişisel fitness hedeflerine ulaşmalarına yardımcı olacak antrenman programları, beslenme planları ve iletişim imkanları sunmalıdır. Kullanıcılar, kayıt olduktan sonra antrenörlerle iletişim kurabilir, kişisel antrenman programları ve beslenme planları alabilir, ilerlemelerini izleyebilir ve online dersler izleyebilirler. Bu proje, fitness ve sağlıklı yaşam hedeflerine ulaşmak isteyen kişilere özelleştirilmiş rehberlik ve destek sunmayı amaçlar. Proje sayesinde cloud veritabanını daha iyi kullanmayı ve ayrıca daha önce web uygulamasında çok deneyimli olmadığım için react ve gerekli yazılım dillerini öğrendim ve kullandım. Ayrıca projenin kuralları:

- Kullanıcı Kayıt ve Giriş:
 - Kullanıcılar; ad, soyad, doğum tarihi, cinsiyet, e-posta adresi, telefon numarası ve profil fotoğrafı gibi temel kişisel bilgileri girerek kayıt olabilmelidir
 - Kullanıcılar, e-posta ve şifre ile giriş yapabilmelidir. Şifreler güvenli bir şekilde saklanmalı ve kimlik doğrulama işlemi yapılmalıdır.
 - Kullanıcıların unutulmuş şifrelerini sıfırlamaları için bir "şifremi unuttum" seçeneği bulunmalıdır.
 - Kullanıcılar; kişisel profil sayfalarını düzenleyebilmeli, profil resmi ekleyebilmeli ve temel bilgilerini güncelleyebilmelidir
- Kullanıcı Rollerini:
 - Uygulamada üç adet kullanıcı rolü bulunmalıdır: Admin, antrenör ve danışan.
 - Admin rolü, en kapsamlı erişim yetkilerine sahiptir ve genel uygulama yönetimini sağlar.
 - Adminler; tüm rollerdeki kullanıcıların hesaplarına erişebilir, yeni hesaplar oluşturabilir, hesapları etkinleştirebilir veya devre dışı bırakabilir.
 - Diğer kullanıcıların verilerini görüntüleyebilir ve düzenleyebilir.
 - Antrenör rolü, danışanlara egzersiz planları oluşturmak ve ilerlemelerini izlemek için gereken yetkilere sahiptir.
 - Antrenörler, kendi profil bilgilerini düzenleyebilir. Antrenörlerin profil bilgileri; ad, soyad, uzmanlık alanları (kilo aldırma, kilo verme, kilo koruma,

kas kazanma), deneyimleri, iletişim bilgileri gibi bilgilerden oluşmalıdır

- Kendilerine atanmış danışanların bilgilerine erişebilir. İlerlemelerini ve planlarını kontrol edebilir.
- Danışanları için özel egzersiz programları oluşturabilir, bunları güncelleyebilir ve danışanlarıyla paylaşabilir. Programlar; egzersiz adı, hedefleri (kilo alma, kilo verme, kiloyu koruma, kas kazanma), set ve tekrar sayıları, video rehberleri, programa başlama tarihini ve programın süresini içermelidir.
- Danışanları için özel beslenme planlarını oluşturabilir, bunları güncelleyebilir ve danışanlarıyla paylaşabilir. Danışanlarıyla iletişim kurabilir, mesajları yönetebilir.
- Antrenörler, kullanıcıları veya diğer antrenörleri yönetme yetkisine sahip değildir.
- Danışan rolü, kendi kişisel verilerini ve egzersiz planlarını yönetmek için gereken yetkilere sahiptir.
- Danışanlar, kendi kişisel profillerini görüntüleyebilir ve düzenleyebilir
- Egzersiz ve beslenme planlarına erişebilir
- Antrenörleriyle sistem üzerinden mesajlaşarak iletişim kurabilir.
- İlerleme kayıtlarını ekleyebilir, görüntüleyebilir ve düzenleyebilir. İlerleme kayıtları; kilo, boy, vücut yağ oranı, kas kütlesi, vücut kitle indeksi gibi verileri içermelidir.
- İlerlemeleri ile ilgili görsel raporlara erişebilir. Raporlar danışanların günlük ya da haftalık kilo, vücut kitle indeksi değişimleri, alınan ve yakılan kaloriler gibi bilgileri görsel olarak sunmalıdır
- Danışanlar, diğer kullanıcıları veya antrenörleri yönetme yetkisine sahip değildir.

• Danışan - Antrenör Eşleştirilmesi:

- Danışan sisteme kişisel bilgi ve hedeflerini girdikten sonra danışan-antrenör eşleştirilmesi yapılacaktır. Antrenör atama işlemi sistematik bir şekilde antrenörlerin uzmanlık alanlarına ve ilgilendikleri kişi sayıları göz önünde bulundurularak yapılacaktır.
- Her antrenörün en fazla 5 danışanı bulunmalıdır.

Bu proje cloud veritabanı ve web öğrenme açısından oldukça önemlidir. Bu projede react ve firebase kullandım.

II. ADIMLAR

Adım 1

İlk adım olarak react ve javaScript gibi yapıları öğrenmek oldu. Daha sonra ise giriş sayfasını açtım ve burada giriş ya da kayıt yapacak kullanıcı için giriş motivasyon sözleri koydum ayrıca navigatör bar oluşturup Login ve Sign In sayfalarına gitmek için iki adet buton koydum. Navigatör barı yaptıktan sonra app.js de bu iki butona basıldığında ve daha sonra yapılacak sayfalarda gezinmek için route yapısını oluşturdum.

Adım 2

Bu adımda Login ve sign sayfalarını yapmaya başladım . Login sayfasında form tagını kullandım ve burada kullanıcının password ve emailini almak için kullandım ve ayrıca 3 adet buton koydum. Bu butonlar:

- giriş butonu : Bu butona basıldığında firebase auth ile kullanıcının kayıtlı olan email ve passwordunu girmis olduğu veriler ile tutarlı olup olmadığı kontrol ediyorum ve kullanıcı hangi type ise ona özel sayfaya gönderiyorum ayrıca eğer kullanıcının hesabı disable olmuşa bilgiler tutarlı olsa bile giriş yapamaz ve eğer kullanıcı admin tarafında oluşturulmuşsa kilo ve boy gibi verilere sahip olamdığı için danisman ve antrenör olmasına göre setup-Danisman veya setupAntrenör sayfalarına yönlendirdim.
- şifre sıfırla butonu : Bu butonda kullanıcı şifresini unutmussa veya değiştirmek istiyorsa bu butondan işlemini gerçekleştirmesini sağladım. Bunun için kullanıcıyı giriş yaptırmam gerekiyordu bunun içinde kullanıcı eğer email ve password bilgilerini girmediyse girmesini istiyorum daha sonra giriş yaptırp yeni şifresini alıp firebase auth kısmında updatePassword methodu ile eski şifreini değiştirmiş oldum.
- kayıt butonu: Burada kullanıcı daha kayıt olmadıysa bu butona basıp kayıt olma sayfasına yönlendirdim.

Login sayfası bittirdim.

Adım 3

Bu adımda Sign In sayfasını yapmaya başladım. Öncelikle her kullanıcıda olması gereken fotoğraf , ad , soyad , cinsiyet, doğum tarihi , email , şifre ve danisman mı antrenör mü olduğu gibi bilgileri form tagı ile aldım ve kullanıcının girdiği email ve şifreye göre firebase auth için bir user oluşturdum ve bu kullanıcının id sini ve type ını projenini locale kayıttım. Daha sonra setupDanisman ve setupAntrenör sayfalarını oluşturup kullanıcıyı seçtiği type göre bu sayfalar yönlendirdim: Bu iki sayfa:

- setupDanisman sayfası: Bu sayfada kullanıcının hedefini , boyunu , kilo ve yağ oranı gibi verileri aldım ve bu veriler ile kütle indeksi , kas kütlesi gibi verileri hesaplayıp firebase cloudada danismanİlerleme adındaki koleksiyona kullanıcının id si ile kayıttım.
- setupAntrenör sayfası : Bu sayfada kullanıcının uzmanlık alanlarını ve deneyimlerini aldım ve bu verileri kullanıcıları tuttuğum Users koleksiyonuna kullanıcının id si ile kayıttığım verilere ekledim.

Bu iki sınıfı yaptıktan ve kullanıcı iki sınıftan birine girip verileri alınıp kayıt olduktan sonra ana sayfalarına yönlendirmek için bu ana sınıfları oluşturmaya başladım bunlar admin , antrenör ve danisman sayfaları oldu.

Adım 4

Bu adımda danisman sayfasını oluşturmaya başladım öncelikle useEffect içine locale kaydettiğim kullanıcı id si ile kullanıcıya ait verileri cloudadan çektim ve depoladım. Daha sonra 6 adet grafik koydum. Bu grafikler danismanın ilerlemelerini göstermek için kullandım. Bunlar kilo, boy, yağ, kalori, kas kütlesi ve kütle indeksi verilerini yansıttım. Daha sonra sayfanın tepesine navigatör bar oluşturdum. Buraya belli sayfalara ve işlemleri gerçekleştirmek için butonlar koydum. Bunlar:

- home butonu : Danisman bu butona basınca direkt giriş sayfasına gönderdim.
- sign out butonu: Burada kullanıcı current kullanıcı durumunda olduğu için onu firebase de mevcut olan signOut metodu ile current kullanıcı olmak çıkardım ayrıca kullanıcı artık current kullanıcı olmadığı için local verileri temizleyip giriş sayfasına yönlendirdim.
- profil butonu: Bu butona basılınca danisman kendi profilini görmesi ve değiştirmesi için bir danismanProfil sayfası oluşturdum. Bu sayfada butona bastığında yönlendirme yapılacağından danismanın verilerini yollayıp bu sayfada kullandım ve kullanıcı istediği gibi değişiklik yapmasını sağladım.
- ilerleme butonu: Bu butona basılınca bir kapanabilir modal yapısı açılıyor ve bende güncel boy , kilo , yağ ve yakılan kalorileri istiyor ve bu verileri setupDanisman sınıfında kullandığım formüller ile kas kütlesi ve kütle indeksi hesaplanıp firebase de danismanın ilerlemelerine ekleniyor ve danismanın ana sayfasındaki grafiklerde dynamic olarak değişiyor.
- Antrenör butonu: Bu butonda setupDanisman sayfasında danisman antrenör eşleşmesi sonucunda danismanın antrenörün bilgilerini göstermek için danisman-Antrenör sayfası oluşturdum. Bu sayfada giriste danismanın antrenörünün bilgilerini çekip bir card yapısında belli kısmını gösterdim.
- programlar butonu: Bu butonda danismanın antrenörünün eklediği programları görmesi için yaptım öncelikle bir modal açılıyor ve hangi programı görmek istediğini soruyor beslenme programını görmek istiyorsa danismanBeslenme sayfasına , hangi egzersiz programını görmek istiyorsa onu seçip danismanEgzersiz sayfasına gitip verileri görebiliyor fakat bu verileri değiştirme yetkiğine sahip değildir.

Navigatör bar yapısını da yaptıktan sonra mevcut antrenör ile mesajlaşmak için sayfanın sağ alt köşesine bir icon buton koydum. Bu buton ile antrenör ile anlık olarak mesajlaşmasını sağladım. Burada firebase clouda da bulunan Message koleksiyonunu useEffect içinde bir kere dinleyerek sağladım. Bu koleksiyonu dinleyerek bu koleksiyonda her bir değişimde bu yapı çalışır ve içine yazdığım kod ile verileri ekrana yansıtarak

bir chat sayfası oluşturdum.Boylelikle danismanlar için olan sayfamda bitmiş oldu.

Adım 5

Bu adımda antrenor sayfasını oluşturmaya başladım öncelikle useEffect içine locale kaydettığım kullanıcı id si ile kullanıcıya ait verileri clouddan cektim ve depoladım.Daha sonra 1 adet grafik koydum.Bu grafikler antrenorun uzman olduğu alanlarda kayıtlı olan kişilerin dağılımını göstermek için kullandım.Daha sonra sayfanın tepesine navigaton bar oluşturdum.Buraya belli sayfalara ve işlemleri gerçekleştirmek için butonlar koydum.Bunlar:

- home butonu :Antrenor bu butona basınca direkt giriş sayfasına gönderdim.
- sign out butonu: Burada kullanıcı current kullanıcı durumunda olduğu için onu firebase de mevcut olan signOut metodu ile current kullanıcı olmak çıkardım ayrıca kullanıcı artık current kullanıcı olmadığı için local verileri temizleyip giriş sayfasına yönlendirdim.
- profil butonu: Bu butona basılınca antrenor kendi profilini görmesi ve değiştirmesi için bir antrenorProfil sayfası oluşturdum.Bu sayfada butona bastığında yönlendirme yapılacağından antrenor verilerini yollayıp bu sayfada kullandım ve kullanıcı istediği gibi değişiklik yapmasını sağladım.
- Danisman butonu: Bu butonda setupDanisman sayfasında danisman antrenor eşleşmesi sonucunda antrenorun uzmanlık alanına göre eklenen danismanların bilgilerini göstermek için antrenorDanismanları sayfası oluşturdum.Bu sayfada giriste firebase cloudda bulunan AntrenorDanismanları koleksiyonundan antrenorun id sine göre belli doküman çekilip bir card yapısında danismanların belli kısmını gösterdim.Bu card yapısında daha danismanın daha detaylı bilgileri görmek için bilgi , programlarını görmek ve düzenlemek için programlar ve program eklemek için program ekle butonu koydum.Bu butonlar:
 - bilgi butonu : Bu butona basılınca danismanInfo adında bir sayfa oluşturdum ve oraya yönlendirdim.Bu sayfada öncelikle yönlendirme işleminde card daki danismanın id sini yollayıp bu id yi danismanInfo sayfasında danismanın verilerini ve ilerleme bilgilerini almak için kullandım.Veriler aldıktan sonra sayfayı iki bölerek yarısında temel bilgilerini gösterdim yarısında ilerleme bilgilerini kullanarak danisman sayfasında bulunan grafikleri kullanarak danismanın ilerlemelerini gösterdim.
 - programlar butonu : Bu butona tıklandığında öncelikle bir modal açılıyor ve beslenmemi yoksa egzersiz mi olduğu seçimi yapılıyor ve ona göre danismanın mevcut program bilgilerini görüp onları düzenleyebiliriz.
 - yeni program ekle butonu ile beslenme ve egzersiz programı eklemesi yapabilmemiz için bir modal açılıyor ve burada yaptığımız tercihe göre

danismanProgramEkle sayfasında gerekli işlemleri gerçekleştirmek için bir yapı çıkarttım.

Navigaton bar yapısını da yaptıktan sonra mevcut danismanlar ile mesajlaşmak için sayfanın sağ alt köşüne bir icon button koydum.Bu buton ile danismanlar ile anlık olarak mesajlaşmasını sağlamak için koydum.Öncelikle bir kapanabilir modal ile antrenorun sahip olduğu kullanıcılardan hangisiyle mesajlaşmak istediğini seçtirdim.Secimden sonra message sayfasında firebase cloudda bulunan Messages koleksiyonunu useEffect içinde bir kere dinleyerek anlık verileri ekrana yansıttım.Bu koleksiyonu dinleyerek bu koleksiyonda her bir değişimde bu yapı çalışır ve içine yazdığım kod ile verileri ekrana yansıtarak bir chat sayfası oluşturdum.Boylelikle antrenorlar için olan sayfamda bitmiş oldu.

Adım 6

Bu adımda admin sayfasını oluşturmaya başladım öncelikle handleInit fonksiyonu içinde kullanıcılara ait tüm verileri clouddan cektim ve depoladım.Daha sonra 2 adet grafik koydum.Bu grafikler sistemdeki kullanıcı type ına göre dağılımını ve zaman göre kişi sayısındaki azalış artışı göstermek için kullandım.Daha sonra sayfanın tepesine navigaton bar oluşturdum.Buraya belli sayfalara ve işlemleri gerçekleştirmek için butonlar koydum.Bunlar:

- home butonu :Admin bu butona basınca direkt giriş sayfasına gönderdim.
- sign out butonu: Burada kullanıcı current kullanıcı durumunda olduğu için onu firebase de mevcut olan signOut metodu ile current kullanıcı olmak çıkardım ayrıca kullanıcı artık current kullanıcı olmadığı için local verileri temizleyip giriş sayfasına yönlendirdim.
- kullanıcı ekle butonu: Bu butona basılınca basta sign in sayfasına çok bezer bir sayfa yaptım ve adına kullanıcıEklemePage koydum.Burada basta kayıt olmak için gereken şeylerin aynısı yaparak firebase auth ve clouddaki User koleksiyonuna eklem işlemini gerçekleştiriyoruz.Bu arada kullanıcı login den giriş yaparken boy , kilo veya antrenor ise eneyim gibi bilgileri girmediği için ana sayfalara gitmek yerine setupDanisman veya setupAntrenor sayfasına giderek burada eksik bilgileri tamamlayıp oyle ana sayfalarına geçiyorlar.
- Antrenor butonu: Bu butonda admin sayfasının başında sistemKullanıcıları verilerinde antrenorların id leri bulunuyordu.Bu id leri kullanarak ve table yapısı kullanarak bir tabloda kullanıcıları sıraladım.Bu tabloda her satırdaki kullanıcılar için 4 adet buton koydum bilgi,disable,unDisable ve sil butonu bunlar :
 - disable butonu : Kullanıcının hesabını devre dışı bırakıp giriş kısmında eposta ve şifresi doğru olsa bile girişine izin verilmemesini sağladım
 - unDisable butonu: Kullanıcının hesabı devre dışı ise bunu tersine çevirmek için kullandım.
 - bilgi buton : Bu buton ile antrenorun bilgilerini düzenleyebilir biçimde gösteriyorum.
 - sil buton : Bu buton ile kullanıcıyı basta Userstan olmak üzere gerekli yerlerden siliyorum.

- Danisman butonu: Bu butonda admin sayfasının basında sistemKullanıcıları verilerinde danismanların id leri bulunuyordu.Bu id leri kullanarak ve table yapısı kullanarak bir tabloda kullanıcıları sıraladım.Bu tabloda her satırdaki kullanıcılar için 5 adet buton koyudm bilgi,programlar,disable,unDisable ve sil butonu bunlar :
 - disable butonu : Kullanıcının hesabını devre dışı bırakıp giriş kısmında eposta ve şifresi doğru olsa bile girişine izin verilmemesini sağladım
 - unDisable butonu: Kullanıcının hesabı devre dışı ise bunu tersine çevirmek için kullandım.
 - bilgi buton : Bu buton ile danismanların bilgilerini düzenleyebilir biçimde gösteriyorum.
 - programlar buton : Bu buton ile danismanın programlarına erişip düzenleyebiliyorum.
 - sil buton : Bu buton ile kullanıcıyı basta Userstan olmak üzere gerekli yerlerden siliyorum.

Navigaton bar yapısını da yaptıktan sonra admin sayfam ve projem bitmiş oldu.

III. ÖZET

Bu projenin amacı,cloud veritabanı kullanarak bir Online Fitness Koçluğu Platformu olusturmaktı. Bu proje yapımında cloud veritabanı ile ilgili işlemleri uygulayarak ve test ederek daha iyi öğrendim ve benimsedim.Ayrıca react,html, css ve javaScripti daha detaylı öğrenme ve kullanam fırsatı yaşadım.

IV. SONUÇ

Tüm kontroller yapıldıktan sonra artık log in sayfasından 3 farklı kullanıcı tipi için giriş yapabiliyorum.Admin ile danismanların ve antrenorlerin bilgilerini gorup degistirebildim.Ayrıca sistemdeki kullanıcı sayılarını vb belgerini grafik olarak gosterebildim.Antrenor ile eşleşen danismanla beslenme ve egzersiz palnlarını ekleyip düzenleyebiliyorum.Ayrıca mesajlaşıp ve danismanların bilgilerini gosterebildim.Danisman ile antrenor bilgilerini gosterebildim ve danismanın yazdığı beslenme ve egzersiz programlarında gosterebilim.Fakat projeyi yaparken bazı problemlerle karşılaştım bunlar:

- Öncelikle sorgu sonuçlarının yanlış ya da hata verip programın durdurulması gibi sorunlarla karşılaştım.Bu sorunlar için, yazdığım sorguları kontrol ederek ve veritabanını kontrol ederek bu sorunları çözdüm.
- kullanıcı silme işlemlerinde bazı koleksiyonların silinmesi ve düzenlenmesinde çok sıkıntı yasadım fakat bu sıkıntıyı deneye deneye cozdum.

Genel olarak projenin sonuçları böyleydi.

V. KAYNAKÇA

- console.firebase.google.com/project/yaz-lab-2-46ab8/firestore/data/2FUsers
- <https://getbootstrap.com/docs/5.3/getting-started/introduction/>
- <https://mui.com/joy-ui/getting-started/tutorial/>
- <https://blog.webdevsimplified.com/2022-07/react-router/>



