

PROJE 1

Egemen akır

Kocaeli Bilgisayar Mühendisliđi
Email: cakiregemen0@gmail.com

I. GİRİŞ

Bu projenin amacı bir üniversitede açılan proje dersleri için talep kriter temelinde öğrenci-hoca eşleştirmesi yapacak bir sistem geliştirmeniz istendi. Bu sayede öğrenci ve hocaların kriterlerine en uygun seçimlerin yapılacaktır. Proje sayesinde veritabanını daha iyi kullanmayı, dart dilini ve flutterı daha iyi öğrendim ve kullandım. Ayrıca projenin kuralları:

- Masaüstü uygulamasında yönetici paneli, öğrenci paneli ve hoca paneli olmak üzere 3 adet panel bulunmalıdır. Yönetici panelinde, öğrenci ve hoca bilgilerinin sisteme kaydedilmesi, güncellenmesi ve sistem yönetimi ile ilgili işlemler, Hoca panelinde, hocaların kendi dersleri için uygun öğrencilerin seçilmesi ile ilgili işlemler, Öğrenci panelinde ise öğrencilerin ilgi alanlarına uygun olarak ve istedikleri hocalardan ders talep işlemleri yapılacaktır. Ders alacak öğrencilere hoca eşleştirme süreci iki ana aşamadan oluşacaktır:

- 1. aşama, dersi almak isteyen öğrencilerin istedikleri hocalarla anlaşarak ders secebilecekleri anlaşmalı seçim sürecini içermektedir. Bu aşamanın süresi, ne zaman başlayıp biteceği yönetici tarafından belirlenecektir. Bu süreçte sırasıyla aşağıdaki işlemler yapılabilir:
- • Öğrenci, hoca ve yönetici ilk başta ilgili panel üzerinden kendi kullanıcı adı ve şifresiyle giriş yapmalıdır.
- • Öğrenci, sisteme giriş yaptıktan sonra kendi transkript belgesinin pdf dosyasını dışarıdan sisteme yükleyecektir. Yuklenen transkript içeriğindeki öğrencinin önceden almış olduğu ders bilgilerin sistemde OCR aracılığıyla okunarak metne dönüştürülmesi daha sonrasında ise bu verilerin veritabanında ilgili alanlara kaydedilmesi sağlanacaktır.
- • Transkriptini sisteme yükleyen öğrenciler almış olduğu derslerle ilgili notları ve bilgileri kendi panelinden görebilecektir.
- • Sistem transkriptini yüklemi, solan öğrencilere almak istediği derse göre dersi veren hocaları listelemeli ayrıca öğrenci ilgi alanlarına göre hoca listesinde filtreleme yapabilir.
- • Öğrenci alacağı her bir ders için sistem yöneticisinin başta belirlediği sayıda farklı "hocalardan" (varsayılan değer: 1) talep oluşturabilecek ve herhangi bir hoca tarafından talebi onaylanmadığı sürece talebini geri çekebilecektir.

- • Öğrenci bir hocadan ders talebi oluştururken hocaya iletmek istediği bir mesajı yöneticinin belirlediği karakter sayısı sınırıyla sistemden gönderebilmelidir. Benzer şekilde hoca da öğrenciyle mesajlaşabilmelidir.
- • Öğrenciye talebini ilk onaylayan hocadan ders eklenecek ve öğrencinin panelinde aldığı ders ve hoca bilgileri gösterilecektir.
- • Hoca sisteme giriş yaptıktan sonra kendi ilgi alanlarını seçerek sisteme kaydedebilmeli böylece öğrenciler bu sayede hocaların ilgi alanlarını görebilmelidir.
- • Hoca sistemden kendi derslerine talep oluşturan öğrencilerin listesini görüntüleyebilmelidir.
- • Hoca ayrıca kendisinin de verdiği bir ders için henüz bir hoca tarafından onaylanmamış talebi olan veya talep oluşturmamış öğrencileri de görüntüleyebilmelidir. Ancak bu öğrencilerin başka hocalardan yaptıkları talep durumlarını diğer hocalar görmemelidir. Dersi başka bir hocadan onaylanmış öğrenciler diğer hocalarda hiç gösterilmemelidir.
- • Hoca listelediği öğrencilerden birine tıklayıp almış olduğu ders bilgilerine erişerek görüntüleyebilmelidir.
- • Hoca bölümde açılan tüm dersler içerisinden önemli gördüklerini seçip her seçtiği ders için de katsayı ataması yaparak kendisine özgü bir öğrenci puanlama formülü oluşturabilmelidir. Oluşturduğu formüle göre listelediği öğrencilerin puanları da gösterilmelidir. (Örnek "formül = (Ders A notu * 3 + Ders B notu * 2 + Ders C notu * 1) / 6). Hoca oluşturduğu bir formül puanına veya seçtiği bir ders notuna göre öğrencileri filtreleyebilmeli ve sıralayabilmelidir.
- • Hoca listelediği öğrencilerden herhangi birine dersini alması için sistemden talepte bulunabilmelidir. Öğrencinin talebi kabul etmesi halinde ders öğrenciye eklenmelidir " ancak öğrenciye red seçeneği verilmemelidir.
- • Hoca yöneticinin belirlediği kontenjan sayısına öğrenci talebini onaylayabilmelidir. (Kontenjan miktarı bir hocadan ders alabilecek toplam öğrenci sayısına karşılık gelmektedir. Bir hocanın farklı iki dersini alan aynı öğrenciler 2 öğrenci gibi düşünülerek kontenjandan düşülmelidir. Her ders için ayrı kontenjan olmamalıdır.)
- • Yöneticinin belirlediği anlaşmalı süre dolduğunda

sistem hoca ve ogrencilerin talepte bulunma, kabul ve onay islemlerini kapatmalıdır. Bu asamadan sonra random atama surecine gecilmelidir.

- İkinci asama, anlasma yapmayan veya talebi hicbir hoca tarafından onaylanmamis ogrencilerin hocalara dagıtılacağı surectir. Bu asamada asagıdaki islemler yapılabilmelidir:
 - • Yonetici panelinden otomatik olarak ogrencilerin atanabilmesi saglanmalıdır.
 - • Bu asama 3 farklı yontemle gerceklesebilmedir:
 - * – Rastgele atama: Tum ogrenciler herhangi bir „sart aranmaksızın sırasıyla farklı hocalara atanacaktır.
 - * – Not ortalamasına gore ortalama: Ortalaması en yuksek ogrenciden baslamak uzere sırasıyla hocalara atama yapılacaktır. (Bu atama sırasında hocaların sırası yonetici tarafından belirlenmelidir.)
 - * – Belirli derslere gore atama: Yonetici tarafından bazı derslerinsecilerek ilgili dersler icin belirlenecek ortak formulasıyona gore atama yapılacaktır. Bu asamada 5 derse kadar secim yapılabilmedir.(Bu atama sırasında hocaların sırası yonetici tarafından belirlenmelidir.)

Bu proje veritabanı ve veritabanı normalleştirmeyi öğrenme acısından oldukça önemlidir. Bu projeyi java dilinde yazdım ve arayüzü kodlamak için swingi kullandım.

II. ADIMLAR

Adım 1

İlk adım olarak main classını actım ve 3 kullanıcı için 3 buton koydum ve daha sonra 3 buton için 3 farklı adda klasor olusturup her bir kullanıcı type için giriş sayfası yaptım.Bu sayfalarda kullanıcıdan kullanıcı adını ve şifresini alıyorum fakat eger kullanıcı ilk defa giriş yapıyorsa yeni şifre alması için ekstra buton ekledim ve eger kullanıcı ilk defa giriyorsa ve girdiği butondaki kullanıcı type ında ise yeni şifre alıp giriş yapmasını sağladım.Bu ilk defa giriş olayını kontrol etmek için databaseden kullanıcının adını ve şifresini öğrenci tablosunda aratıp çıkan sonucu değerlendiren bir method yazdım eger method sonucunda yeni şifre alırsa yeni şifresini database update işlemi için yazdığım metotlardan biri ile mevcut kullanıcının şifresini guncelliyorum.

Adım 2

Bu adımda hoca ve ogrenciler için onların ana ekranlarına gecmeden hoca için ilgi alanlarını ve vereceği derslerini secmesi , öğrencinin ise alacağı dersleri , ilgi alanını ve transkript belgesini yuklemesi için öğrenciSetupPage ve hocaSetupPage sayfaları yaptım.Hoca setup page sınıfında SqlSelectService sınıfında yazdığım getAllDers ve getAllIlgiAlani methodları ile databasede bulunan dersleri ve onların bilgilerini List<Map<String,dynamic>> sekinde adım ve setup sayfasında hocanın kısa bilgilerini ve aldığım verileri iki farklı şekilde listeleyp hocaya secim yapma islemlerini

yapmasını sağladım.Aynı methodları kullanarak aynı tabloları ve kısa bilgi yapısını gosterdim ayrıca öğrencinin transkript belgisini yuklemesi için bir button koydum.Bu butona basıldığında flutterda ki filePicker ile dosyayı alıp syncfusionflutterpdf paketi ile pdf deki verileri tek tek okudum ve okudugum verileri bir dizide tutum.Databasede öğrencinin gecmis verileri adında tablom bu verileri eklemek için gecmisVeriEkle(int öğrenciId) yapısında bir method yazıp tutugum verileri forEach yapısında tek tek ekleme işlemi yaptım.Son olarak her iki setup sayfasında yer alan bilgileri kaydet butonu ile secilen dersleri ve ilgi alanlarını öğrencidestable,öğrenciilgiAlanitable,hocaderstable ve hocailgiAlanitable tablolarına ekleme işlemi yaptım.Bu adım sonunda hoca ve öğrencinin ilk giriş sonucunda karşılasacağı setup sayfalarını yaptım ve hoca ve öğrencilerin verileri ekledim.

Adım 3

Bu bolumde öğrenci main sayfasını yapmaya başladım.Bu sayfada gecmis bilgiler , dersler ve ilgi alanları, talep gecmisi, talep olusturma ve gelen talep adında butonlar koydum.Bu butonlar:

- talep olustur butonu: Bu butonda öğrencinin aldığı dersleri databaseden cekiyorum ve bir showDialog ile ders sayısı kadar buton olusturup bu butonlara her tıklanıldığında getFilterHocaData methodu ile butonda hangi ders varsa o dersi veren hocaların bilgilerini gosterip aynı zamanda bilgilerin yanında talep olustur butonu ile hocadan talep olusturma işlemini yapabiliyorum.Ayrıca talep olustururken hocaya mesaj da atabiliyorum.
- gelen talep butonu : Bu butona basıldığında öğrencinin id sini kullanarak öğrenci talep tablosundan talepleri cekip tabloda gosterebiliyorum.Ayrıca gelen talebi kabul et butonu ile kosullar uygun ise öğrenciyi ,dersi ve hocayı onaylanmisderslertable ına ekliyorum ayrıca hocanın kontenjanı gibi verileri SqlUpdateService sınıfında bulunan methodlar ile guncelledim.
- talep gecmisi butonu: Bu butona tıklanıldığında öğrencinin tüm taleplerini cekiyorum ve öğrenci talep sınıfının ozelligi olan durum bilgisi , ders adı , hoca adı gibi verileri tabloda listeledim.
- gecmis bilgiler butonu: Bu butona basıldığında SqlSelectService sınıfında yer alan getGecmisVeri(int id) methodu ile girilen öğrenci id ile öğrencinin gecmisveritable tablosundan verileri cekip tabloda gosterdim.
- dersler ve ilgi alanları butonu : Burada ise öğrencinin id si ile öğrenciderslertable ile öğrenciilgiAlanitable ında ki öğrenciye ait verileri alıp tabloda gosterdim.

Bu butonların action ozelliklerini yaptıktan sonra artık öğrenci ile ilgili istenen isterleri yapmış oldum.

Adım 4

Öğrenci main sayfasını ve öğrenci ile ilgili yapılması gereken tüm isterleri yaptıktan sonra Hoca main sayfasını yapmaya başladım.Bu sayfaya dersi alanlar ve onaylananlar

, dersi alanlar fakat dersi onaylanmayanlar ve talepler adında 3 buton koydum.Bu butonlar:

- talepler buton:Bu butonda hoca id si ile SqlSelectServiceteki bir methodan hocataleptable tablosundan verileri ceptim ve tabloda verileri listeledim.Ayrıca hoca talebi onaylaması , reddetmesi için butonlar koydum.Onayla butonu ile talebi alan ogrenci onaylananderslertable tablosuna eklenecek ve hocanın kontenjan bilgileri guncellencek.Reddet butonu ile ogrencinin ogrenciderstable tablosunda yer alan ders almak için talepte bulunulan dersin kalan talep sayısı ve mevcut talebin durum ozelligi guncelliyorum.
- dersi alanlar ve onaylananlar butonu : Bu butonda ogrenci hoca id sini kullanarak onaylananderslertable tablosunda hocaid = id kosulunun saglayan ogrencileri ceptim ve tabloda bu ogrencilerin adını , soyadını ve onaylanan dersini gosterdim ayrıca ogrencinin aldığı tüm dersleri ve ilgi alanlarını ogrencinin bilgilerinin üzerine tıklayıp hocaya toplolar seklinde gosterdim.
- dersi alanlar fakat onaylanmamış ogrenciler butonu:Burada ogrenci main sayfasında talep olusturma butonunda yaptığım işlem gibi hocanın verdiği dersleri buton seklinde listeledim.Hoca verdiği dersi tıkladığında bir filtreleme methodu yazdım.Bu methodda öncelikle ogrenciderstable tablosunda dersi id sine gore verileri ceptim.Daha sonra ceptigim verli List<Map<String,dynamic>> tipinde olduğu için foreach ile her bir ogrencinin id si ve ders id si ile onaylanmışdeslertable tablosunda filtreleme işlemi ile dersi alan fakat dersi hic bir hocadan almamış ogrencilerin verilerini depoladım ve tabloda verileri yazdırdım ayrıca hoca isterse ogrenciye dersi alması için talepte bulunabilir.

Böylelikle hoca main sayfasında bitirdim ve hoca ile ilgili isterleride yapmış oldum.

Adım 5

Hoca tipindeki kullanıcılar için main sayfası ve isterler bittikten sonra yönetici main sayfasını yapmaya başladım.Bu sayfaya 9 adet buton koydum ve yönetici ile ilgili isterleri yapmak için butonların actionlarını yazdım.Butonlar:

- dersler ve ilgi Alanları buton:Bu butonda tüm dersleri ve ilgi alanlarını databaseden cekip iki farklı tablo da verileri listeledim.Ayrıca yeni bir ders ya da ilgi alanı eklemek için iki tablonun başlığının yanına ekle butonu ve ders bilgisine tıkladığında ders adını guncelleme ya da dersi silmek için butonlar koydum.Ders ya da ilgi alanı ekleme butonuna tıkladığında ise bir pencere açılıyor ve ders ya da ilgi alanı adı girerek ekle butonuna basarak ilgili tabloya ekleme yapabiliyoruz.
- geçmiş işlemler buton : Bu butonda ogrencitaleptable ve hocataleptable tablolarındaki verileri ceptim. Ve bir tabloda listeledim her talep bilgisini yazdırdığım satırda 3 adet buton ekledim ve bu butonlar :
 - onayla buton : Bu butonda yazdığım onaylaButon methodunu kullandım.Bu butonda eger talep

zaten onaylanmış ise hiçbir işlem yapılmıyor aksi durumda hocanın kontenjan ve ogrencinin aynı hocadan ders alabilme gibi durumları kontrol ediliyor eger sıkıntı çıkmaz ise talep onaylanıp ders onaylananderslertable tablosuna ekleniyor ayrıca hocanın kontenjan bilgileri güncelleniyor.

- reddet butonu : Bu butonda yazdığım reddetButon methodunu kullandım.Bu methodda öncelikle talep durumu kontrol ediliyor ve durum != "red edilmiş" ifadesi doğru ise if bloguna girer ve orada durum == "onaylanmış" mı diye kontrol edilir eger onaylanmış bir talep ise onaylı derslerden silinir ve hoca kontenjan ve ogrencinin dersinin kalantalep sayıları guncellenir ayrıca durum beklemede ise sadece talep tablosunda talebin durum özelligi guncellenir.
- sil butonu : Bu butonda yazdığım siButon methodunu kullandım.Bu methodda öncelikle talep durumu kontrol ediliyor ve durum != "onaylanmış" ifadesi doğru ise if bloguna girer ve onaylı derslerden silinir ve hoca kontenjan sayıları guncellenir ayrıca bloga girmez ise sadece talep tablosunda talep silinir ve ogrencinin dersinin kalan talep sayısı guncellenir.

- sistemdeki kullanıcılar buton:Bu butonda tüm kullanıcıları verileri ile bir listede topladım ve tabloda listeledim.Herhangi bir kullanıcının üzerine tıkladığında kullanıcının verilerini gorme ve düzenleme işlemleri gerçekleştirdim.
- kullanıcı ekle buton : Bu butonda showDialog ile bir pencere açıyorum ve kullanıcı adını , soyadını ve hangi type kullanıcı olduğunu drop down button yaparak alıyorum daha sonra ekle butonu ile kullanıcı type ına(Hoca , Ogrenci) gore ogrencitable ya da hocatable tablolarından birine atıyorum.
- ders secimi baslat buton: yönetici main sayfasına giriste tüm sistem verilerini cekmistim ve sistem verilerinde yer alan derssecimdurumuna gore bu butona tıkladığında ders secimi baslatma işlemi gerçekleştiriyorum.Eger ders secimi devam ediyorsa buton tıkladığında bir şey olmaz fakat diğer durumlarda bu butona basıldığında kullanıcıdan bir sayı girmesini istiyorum ve o anki tarihten girdiği gün sayısı kadar sonraki tarihe kadar olan süreçte ders secimi işlemi aktifleştiriyorum ayrıca sistemayarlarstable tablosunda derssecimdurum , baslangictarih ve bitistarih gibi degerleri guncelliyorum.
- 2. asamaya gec buton : Bu butona basıldığında karşımıza iki seçenek çıkıyor.Biri demo diğeri 2. asamaya gec butonları.Aslında ikiside aynı fakat 2. asama gec butonuna tıkladığında sonuc gormek istiyorsak dersSeciminin bitmiş olması lazım diğer türlü girilemez.Fakat bu durumu beklemek için demo butonu var.Bu iki butonun içerisinde 3 buton var.Bunlar:

- rastgele atama buton: Burada tüm ogrencileri bilgileri ve almış olduğu derleri ceptim ve almadığı dersleri dersi veren hocalara rastgele atama işlemi yaptım.

- hoca onceligine gore atama button:Burada oncelikle yoneticiye tüm hocaları ilk tıklanan en oncelikli olacak sekilde tüm hocaları sectiriyorum.Daha sonra ogrencileri ve onaylanmamıs derslerini dersi veren hocalara oncelik sırasına gore ekliyorum.
- hoca onceligi ve belirli derleri atama button : Burada oncelikle yoneticiye tüm derslerden sadece atama yapılacak dersleri sectiriyorum daha sonra tüm hocaları ilk tıklanan en oncelikli olacak sekilde tüm hocaları sectiriyorum.Daha sonra secilen dersleri alan fakat onaylı olmayan ogrencileri dersi veren hocalara oncelik sırasına gore ekliyorum.

- ders secim tarihlerini ayarı button : Bu butonda ise ders secimi devam ediyor ise islem yapılabiliniyor.Bu butonda yapılan islem ise ders secim bitim tarihini ayarlama işlemi yapılır.
- ayarlar button: Bu butonda SqlSelectService sınıfında bulunan getSistemAyarlari methodu ile ugulamdaki kriterleri liste olarak cekiyorum.(maxmesajuzunlugu,baslangıcgrencitalepsayisi,ogrencifarklıhocadandersalabilir,...)
Bu ozellikleri tabloda listeliyorum ve eger degiştirmek istedigı veriler varsa degiştirip kayded guncelle butonu ile degişiklikleri kaydedebilir.
- otomatik kullanıcı ekle button :Burada otomatik olarak tüm degerleri rastgele olusturulmus 50 ogrenci olusturuyorum ve ogrencitable tablosuna ekliyorum.

Bu yonetici main sayfasını ve butonlarıda doldurarak yonetıcının isterlerini ve projeyi tamamlamıs oldum.

III. ÖZET

Bu projenin amacı, veritabanı kullanarak bir universitede acılan proje dersleri için talep kriter temelinde ogrenci-hoca eşlestirmesi yapacak bir sistem geliştirmemizdi.Bu proje yapımında veritabanını ve veritabanı ile ilgili işlemleri uygulayarak ve test ederek daha iyi öğrendim ve benimsedim.Ayrıca projenin arayüzünü dart flutter ile yaptım ve flutterı daha detaylı öğrenme ve kullanam fırsatı yaşadım.

IV. SONUÇ

Tüm kontroller yapıldıktan sonra artık ana menüde 3 farklı kullanıcı tipi için giriş yapabiliyorum.Yonetici ile genel sistem ayarlarını,dersleri ,ilgi alanlarını , kullanıcıları , talepleri ve diğer özellikleri görüp ve bilgileri degistirebiliyorum.Hoca ile ogrenciden gelen taleplere ve mesajlara cevap verebiliyorum.Ogrecileri belli degere gore filtreleyebiliyorum.Ogrenci ile aldıgı derse gore o dersi veren hocaları listeleyip dersi almak için talep ya da mesaj atabiliyorum.Ayrıca transkript belgesini yüklemisse daha önce aldıgı dersleri ve derslerin bilgilerini gösterebiliyorum.Fakat projeyi yaparken bazı problemlerle karşılaştım bunlar:

- Öncelikle sorgu sonuclarının yanlış ya da hata verip programın durdurulması gibi sorunlarla karşılaştım.Bu sorunlar için, yazdıgım sorguları kontrol ederek ve veritabanını kontrol ederek bu sorunları çözdüm.
- Karşılaştığım diğer sorun ise otomatik ogrenci olusturma ya da 2. asamada gerceklestirilecek isterleri yapmak için

yazdıgım methodların yanlış sonuc vermesi.Bu sorunları teker teker kontrol ve bazı yerlere kontrol amaçlı konsol çıktısı atayarak hata tespiti ve düzenlemesi yaptım.

Genel olarak projenin sonuçları böyleydi.

V. KAYNAKÇA

- <https://youtu.be/URskGHtvFIE?si=ZanZ33rZAxZXt16i>
- <https://flutter.dev>
- <https://help.syncfusion.com/flutter/pdf/working-with-text-extraction>

