

YAZILIM LAB.I PROJE 1

Proje Dersi Kayıt Sistemi

1st Sibel özlem Gül
Bilgisayar Mühendisliği 3. sınıf
sibelozlemgul811@gmail.com
210201034

2th Emircan Duman
Bilgisayar Mühendisliği 3. sınıf
emircanduman10@gmail.com
210201043

Bu rapor,yazılım laboratuvarı-1 dersinin birinci projesini açıklamak ve sunumunu gerçekleştirmek amacıyla hazırlanmıştır.Verilen projede Bir "universitede açılan proje dersleri için talep-kriter temelinde "öğrenci hoca eşleştirmesi yapacak bir sistem geliştirmemiz beklenmektedir. Bu sayede "öğrenci ve hocaların kriterlerine en uygun seçimlerin yapılması hedeflenmiştir. Proje sayesinde öğrencilerin veri tabanı dersinde edindikleri bilgilerin pekiştirilmesi, veritabanı tasarımı ve sorgu yazma becerilerinin gelişimiyle birlikte pratik sağlamamız amaçlanmıştır.Bu amaç doğrultusunda Rapor özet, giriş, yöntem, deneysel sonuçlar, sonuç ve kaynakça bölümünden oluşmaktadır.

I. GİRİŞ

Proje Dersi Kayıt Sistemi projesi genel olarak Öğrenci ve Öğretmen classlarından oluşmaktadır. öğretmen ve öğrenci classları sisteme giriş yapan öğrenci ve öğretmeni belirlemek amacıyla oluşturulmuştur. aynı zamanda GUIAtama(),ÖğrenciEkrani(), ÖğrenciİlgilAlanıGörüntüle(),ÖğrenciDersleriGörüntüle(), GirişEkrani(),YöneticiGirişEkrani(),ÖğretmenGirişEkrani(), ÖğrenciGirişEkrani(),YöneticiEkrani(),DersleriListeleEkrani(), İlgilAlanlarıListeleEkrani(),ÖğretmenEkrani(Öğretmen öğretmen), ÖğretmenEkraniTalepListele(Öğretmen öğretmen),YöneticiTalepleriListele(), ÖğretmenÖğrencileriListele(),ÖğretmenDersleriGörüntüle(Öğretmen öğretmen) ÖğretmenİlgilAlanlarıListele(), App(Connection connection) , metodlarından oluşmaktadır. metodların içerikleri yöntem kısmında açıklanmaktadır.

II. ÖZET

Masaüstü uygulamasında yönetici paneli, öğrenci paneli ve hoca paneli olmak üzere 3 adet panel bulunmaktadır. 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 yapılmıştır.dersi almak isteyen öğrencilerin istedikleri hocalarla anlaşarak ders seçebilecekleri anlaşmalı seçim sürecini içermektedir.bu aşamanın süresi ne zaman başlayıp biteceği yönetici tarafından belirlenmiştir öğrenci,hoca ve yönetici ilgili panel üzerinden giriş yapmaktadır.transkript

sisteme yüklenmiş ve kendilerine ait bilgileri kendi panellerinde görebilmektedirler.öğrenciler kendi ilgi alanlarına göre dersi veren hocaları listeleyebilmektedir.öğrenci yöneticinin belirlediği sayıda talep oluşturabilmektedir ve talebini geri çekebilmektedir.hoca sisteme giriş yaptığında kendi alanlarını seçerek sisteme kaydebilmektedir sonuç olarak öğrenciler öğretmenlerin ilgi alanlarını görebilmektedir.aynı zamanda hoca sistemden kendi derslerle talep oluşturan öğrencilerin listesini görüntüleyebilmektedir.Yöneticinin belirlediği anlaşmalı süre olduğunda sistem hoca ve öğrencilerin talepte bulunma,kabul ve onay işlemlerini kapatabilmelidir.Tüm öğrenci hoca taleplerinin işlem geçmesiyle birlikte görüntülenmesi yapılmaktadır aynı zamanda ders seçiminin 1. ve 2. aşama süreçleriyle ilgili gerekli tüm parametrelerin ve sistem ayarlarının yönetimi yapılmaktadır.İlgil alanlarının sisteme kaydedilmesi , güncellenmesi ve silinmesi öğrencinin birden fazla dersi sadece tek bir hocadan alabileceği kısıtlamasının aktif , pasif durumunun belirlenmesi işlemleri yapılmaktadır.

III. YÖNTEM

Veri bağlantısı oluşturabilmek amacıyla connection URL değişkeni oluşturulmuştur, aynı amaç doğrultusunda connection nesnesi oluşturulmuştur.Projenin arayüzünü oluşturabilmek için panel nesnesi oluşturulmuş ve arayüze eklemek için ve istekleri yerine getirebilmek amacıyla ve formlar arası geçiş sağlayabilmek için buton nesneleri oluşturulmuştur.Yöneticinin durum belirleyebilmesi için combobox nesnesi tanımlanmış aynı dersten maksimum talep sayısı, taleplerin maksimum karakter sayısını ve rastgele öğrenci oluşturmak için ve belirli değer aralıklarında seçim yapabilmek amacıyla JSpinner nesneleri tanımlanmıştır.Kullanıcıdan girdi alabilmek amacıyla JTextField nesneleri oluşturulmuş ve aynı şekilde kullanıcıya bilgi iletmek için JLabel nesneleri oluşturulmuştur.Bir renk nesnesini temsil etmek için JLabel nesneleri oluşturulmuştur.Bir renk nesnesini temsil etmek için Color primary nesnesi tanımlanmış ve metin biçimlendirmesi ve yazı tiplerinin ayarlanması için Font nesnesi oluşturulmuştur.Sisteme giriş yapan öğretmen ve öğrenci bilgilerine veritabanından erişebilmek amacıyla öğretmen , öğrenci nesneleri oluşturulmuştur.Yöneticinin belirleyeceği Talep durumunu ayarlayabilmek için TalepDurumu enum'ı ve Ders talep durumunu belirleyebilmek için DersTalepDurumu enum'ı oluşturulmuştur.Sisteme giriş yapan öğrenci veya

öğretmenin veritabanındaki bilgilerini tutabilmek için Öğrenci ve Öğretmen classları oluşturulmuştur. Öğrenci classına özellik olarak no, ad, soyad, sifre, ilgi alanları, genelNot özellikleri verilirken Öğretmen classına nitelik olarak sicilNo, ad, soyad, sifre, ilgiAlanlari, kontenjanSayisi, acilanDersler özellikleri verilmiştir. Yönetici , öğrenci ve öğretmen girişinde bulunan TextField'lerin temizlenebilmesi için TextFieldTemizle metodu oluşturulmuş ve setText metodu ile temizleme işlemleri yapılmıştır. Giriş paneline eklenmek amacıyla StandartGirisPaneliButonu() metodu ile buton StandartGirisPaneliButonu() metodu ile buton StandartGirisPaneliTextField() metodu ile textField, StandartGirisPaneliLabel() metodu ile label döndürülmüştür. GUIAtama() metodu ile Arayüzde bulunan buton, comboBox, spinnerModel, TextField, Label nesneleri oluşturulmuştur. Öğrenci giriş yaptığında öğrenci formunun oluşturulması için ÖğrenciEkranı(Öğrenci öğrenci) metodu oluşturulmuş ve parametre olarak giriş yapan öğrenci bilgilerine erişebilmek için öğrenci nesnesi gönderilmiştir. ÖğrenciİlgiAlaniGoruntule() metodu ile öğrencilerin ilgi alanlarının gösterilmesi amaçlanmıştır. İçerik olarak öğrenciler tablosundan belirli bir öğrencinin ilgi alanlarını sorgulamak için ÖğrenciPreParedStatement nesnesi oluşturulmuş aynı şekilde parametreler tablosunda bulunan ilgi alanlarına erişebilmek için parametrelerResulset oluşturulmuştur. Arayüzde verilerin gösterilebilmesi için DefaultTableModel oluşturabilmek için model nesnesi oluşturulmuştur. model nesnesine ilgi Alani ve ilgi Alani Durumu bilgileri eklenmiştir. Öğrenci ilgi alanları ve parametre ilgi Alanları bilgilerini arrayListte tutabilmek için öğrenciİlgiAlanlari ve ParametreİlgiAlanlari arraylist'i oluşturulmuştur. for döngüsü kullanılarak parametreİlgiAlanlari arrayListinde dolaşma işlemi ile contains metodu kullanılarak ÖğrenciİlgiAlanlari arrayListinde ilgiAlani olup olmadığı kontrol edilmiş var ise model nesnesinin ilgili satırına Dahil Stringi eklenirken yok ise Dahil değil stringi eklenmiştir. Veritabanı sorgusunda alınan verileri göstermek için jTable nesnesi kullanılmıştır. Aynı metod içerisinde öğrenci ders kaydı var ise dersleri görüntüle yoksa pdf yükle butonları panel'e eklenmiş aynı şekilde öğrenci ilgi alanları Butonu, öğrenci ilgi alanı ekle butonu ve öğrenci ilgi alanı sil butonu panel'e eklenmiştir. Giriş Ekranı için GirisEkranı() metodu, Yönetici giriş ekranı için YoneticiGirisEkranı() öğretmen giriş ekranı için ÖğretmenGirisEkranı(), öğrenci giriş ekranı için ÖğrenciGirisEkranı() metodları oluşturulmuştur. YöneticiEkranı() metodu içerisinde arayüze yöneticiDurumComboBox, aynıHocaMultCheckBox, aynıDersmaksTalepLabel, talepMaksKarakterSpinner, talepmaksKarakterLabel, yöneticiParametreleri, kaydetmeButonu, rastgeleÖğrenciOlusturSpinner, raasgeleÖğrenciOlusturLabel, rastgeleÖğrenciOlusturButonu, yöneticiTalepleriListeleButonu, yöneticiÖğretmenListeleButonu, yöneticiÖğrenciListeleButonu, yöneticiİlgiAlanlariListeleButonu, yöneticiDersleriListeleButonu, anaGirisEkranıDon nesneleri eklenmiştir. PreParedStatement ve Resulset kullanılarak parametreler tablosuna erişilmişif blokları

kullanılarak veri alındıktan sonra if(resulSet.next()) ile bir sonraki satıra geçilir ve bu satırın var olup olmadığı kontrol edilmiştir. Veri işleme kısmı , bu satırdaki değerlerin alınması ve switch case yapısının kullanılmasıyla gerçekleştirilmiştir. TalepDurumu adındaki bir enum yapısı resulSet.getString(2) ile elde edilen sütun değerleriyle karşılaştırılmıştır. Şöyle ki; resulSet.getString(2) ifadesiyle ikinci sütundaki değer alınmış ve TalepDurumu enum yapısındaki değerle karşılaştırılmıştır. TalepDurumu.valuof(resulset.getString(2)) ifadesi kullanılarak veri tabanından alınan değeri TalepDurumu enum türüne dönüştürülmüştür. Eğer TalepDurumu enum yapısındaki değer anlaşma ise yöneticiDurumComboBox'ın seçili indeksi 0 olarak ayarlanır. Eğer TalepDurumu enum yapısındaki değer rastgele ise yöneticiDurumComboBox'ın seçili indeksi 1 olarak ayarlanır. Eğer Talep Durumu enum yapısındaki değer durdur ise yöneticiDurum Combobox'ın seçili indeksi 2 olarak ayarlanır. Bu yapı ile özellikle veritabanından alınan enum tipindeki verinin switch-case yapısıyla kontrol edilip, bu değere göre belirli bir ComboBox bileşenlerinin seçilen indeksinin ayarlanması sağlanmıştır. Bu durum kullanıcı arayüzünde belirli bir durum veya seçenekle ilgili önceden tanımlanmış bir seçeneğin gösterilmesi veya işaretlenmesi için kullanılır. Veritabanından alınan verilerin kullanıcı arayüzünde gösterilmesi veya kullanıcıların bu veriler üzerinde işlem yapmasına olanak tanınması amacıyla checkBox ve Spinner bileşenleri kullanılmıştır. Böylece program içinde kullanılan veritabanı verilerinin kullanıcı arayüzünde görsel bir şekilde temsil edilmesi sağlanmıştır. Şöyle ki; Resulset üzerinden alınan verilerin kullanıcı arayüzünde belirli bileşenlere atanması için aynıHocaMultCheckBox.setSelected(resulset.getBoolean(3)); ifadesi ile Resulset ' ten alınan üçüncü sütun değerini bir checkBox bileşenine atılmış , getBoolean(3) ile bu sütundaki değer boolean türüne alınmıştır ve aynıHocaMultCheckBox adındaki bir CheckBox bileşenine atanmıştır. Eğer sütundaki değer true ise aynıHocaMultCheckBox seçili (checked) olur. aynıDersMaksTalepSpinner.setValue(resulSet.getInt(4)); ve talepMaksKarakterSpinner.setValue(resulSet.getInt(5)); ifadeleri ile Resulset'ten alınan dördüncü ve beşinci sütun değerlerini sırasıyla bir Spinner bileşenine atanmış getInt(4) ve getInt(5) ile bu sütunlardaki değerler integer olarak alınmıştır ve ilgi Spinner bileşenlerine atanmıştır. Bu sayısal bir değeri temsil eden Spinner bileşenlerin değerlerini bekirler. DersleriListeleEkranı() metodu ile belirli veritabanı bilgilerini almak ve bu bilgileri bir tablo şeklinde kullanıcı arayüzünde göstermektir. İşlevi adım adım açıklamak gerekirse panel.removeAll() metodu ile mevcut panele ait tüm bileşenler kaldırılmıştır. Yani eğer önceden panele eklenmiş başka bileşenler varsa , bu metodun çağırılmasıyla o bileşenler temizlenmiştir. Veritabanından parametreler tablosundan id değeri 1 olan satırın dersler sütunundaki veriyi almak üzere bir sorgu hazırlanmıştır. Resulset resulset = statement.executeQuery("SELECT dersler From parametreler Where id=1") Veritabanında bu sorgu

çalıştırılarak sonuç resultSet adlı ResultSet nesnesinde saklanmıştır.if(resultSet.next())... bloğu kullanılarak , sorgudan dönen sonuçlar var ise işlem yapılmasını kontrol etmek için yazılmıştır.Eğer veri varsa ResuSet'in ilk satırına geçmesi sağlanmıştır.DefaultTableModel model = new DefaultTableModel(); ile boş bir tablo modeli oluşturulmuştur.model.addColumn("Dersler") ile Tablo modeline dersler sütunu eklenmiştir.for döngüsü içerisinde resultSet.toArray(1).toArray() kullanılarak sorgudan alınan dersler sütununun içeriği alınmış ve bir String[] dizisine dönüştürülmüş ve for döngüsü ile dizi içerisinde dolaşarak her eleman model ' e eklenmiştir.table = new JTable(model) kullanılarak oluşturulan model (DefaultTableModel) kullanılarak bir jTable oluşturulmuştur.Oluşturulan tablo için satır yüksekliği ve yazı tipi ayarları yapılmış panele eklenmiş ve statement.close(), resultSetClose() kullanılarak Panelin yeniden çizilmesi ve değişikliklerin ekranda görülmesi sağlanmıştır.ilgiAlanlariListeleEkran() metodunun amacı , bir grafiksel kullanıcı arayüzü ekranı oluşturmak ve veritabanından belirli verileri alarak bu bilgileri listeyen bir işlevi yerine getirmektir.işlev şu şekilde açıklanabilir;PanelremoveAll() metodu ile mevcut panele ait tüm bileşenler kaldırılmış ve Veritabanından parametreler tablosundan id değeri 1 olan satırın ilgi-alankarı sütundaki veriyi almak üzere bir sorgu hazırlanmıştır.ResultSet resultSet=Statement.executeQuery("Select ilgi-alanlari From parametreler WHERE id=1 veri tabanından bu sorgu çalıştırılmış ve sonuç resultSet adlı ResuSet nesnesinde saklanmıştır.model tablosuna ilgialanlari bilgileri eklenmiştir.OgretmenEkran() metodu Ogretmen sınıfından bir nesne alır ve bu öğretmenin bilgilerini temsil eden bir kullanıcı arayüzü ekranı oluşturmak amacıyla tasarlanmıştır.this.ogretmen = ogretmen; oluşturularak parametre olarak gelen ogretmen nesnesi, bu sınıfın ogretmen değişkenine atanmıştır. OgretmenEkran() TalepListele(ogretmen); ile OgretmenEkran()TalepListele adında başka bir metodu çağırılmış Bu metot ile mevcut öğretmenin taleplerini listelenmiş ve bu ekrana eklenmiştir.OgretmenEkran()TalepListele() metodu ile belirli bir öğretmenin taleplerini bir tablo şeklinde listelemek için tasarlanmıştır.Bu taleper, bir veritabanı tablosundan alınır ve öğretmenin taleplerinin kullanıcı arayüzünde listelenmesi sağlanır.PreparedStatement preparedStatement = connection.prepareStatement("Select * from anasmalar where ogretmenno=?") Bu satır ile anasmalar tablosundan , parametre olarak verilen öğretmen numarasına (ogretmenno) göre sorgu oluşturulmuştur.preparedStatement.setInt(1,ogretmen.sicilNo);Bu satır ile öğretmenin sicil numarasını sorgunun parametresine eklemiştir.ResultSet resultSet = preparedStatement.executeQuery();Bu sorgu çalıştırılarak ResultSet içinde öğretmenin taleplerini içeren sonuçları alınmıştır. while(resultSet.next()) - ResultSet içinde dönerek her bir satırın verilerini alınmış ve bu verileri tabloya eklenmiştir.YöneticiTalepleriListele() metod yöneticinin talepleri listelemesi için oluşturulmuştur. Yönetici, verita-

banındaki anasmalar tablosundaki talepleri çeker ve bunları bir tablo şeklinde kullanıcı arayüzünde gösterir.ResultSet resultSet = statement.executeQuery("SELECT * FROM anasmalar"); sorgusu kullanılarak , anasmalar tablosundaki tüm sütunları seçilmiş ve sonuçlar resultSet içinde saklanmıştır.while(resultSet.next()) ... - ResultSet içinde dönerek, her bir satırın verileri alınmış ve bu veriler tabloya eklenmiştir. OgretmenOgrencileriListele() metodu Bu metod, bir öğretmenin ilgi alanlarına göre öğrencileri listelemek için tasarlanmıştır. Metodun işleyişi adım adım açıklanabilir: Statement statement = connection.createStatement(); - Veritabanı sorguları çalıştırmak için bir Statement nesnesi oluşturulur. ResultSet resultSet = statement.executeQuery("SELECT * FROM ogrenciler"); - Bu sorgu, ogrenciler tablosundaki tüm verileri çeker ve resultSet içine yerleştirir.PreparedStatement preparedStatement = connection.prepareStatement("SELECT ilgialanlari FROM hocalar WHERE sicilno = ?"); Öğretmenin ilgi alanlarını almak için önceden hazırlanmış bir başka sorgu oluşturulur. preparedStatement.setInt(1, ogretmen.sicilNo); - Öğretmenin sicil numarası bu sorgunun parametresine eklenir. ResultSet resultSet1 = preparedStatement.executeQuery(); - Bu sorgu çalıştırılır ve sonuçlar resultSet1 içine yerleştirilir. DefaultTableModel model = new DefaultTableModel(); - Boş bir tablo modeli oluşturulur. model.addColumn("Ogrenci No");, model.addColumn("Ad");, vb. - Tablo modeline sütunlar eklenir. resultSet1.next(); - İlk satıra geçilir. Bu sorgu yalnızca bir sonuç döndüreceği için next() çağrılabilir. while (resultSet.next()) ... - resultSet içinde dönerek, her bir öğrencinin ilgi alanlarını kontrol eder. Öğrencinin ilgi alanları ile öğretmenin ilgi alanları arasındaki kesişimi (ortak ilgi alanları) hesaplar. Ortak ilgi alanları olan öğrenciler tabloya eklenir. Son olarak, tüm kullanılan ResultSet'ler ve Statement'lar kapatılır ve tablo arayüzü için gerekli bileşenler eklenir ve ekranda gösterilir. OgretmenIlgiAlanlariListele() metodu ile bir öğretmenin veritabanında bulunan dersler ile ilişkisini inceleyerek, hangi dersleri verdiğini veya vermediğini belirlemek için kullanılır. Öğretmenin dersleri almak için önceden hazırlanmış bir sorgu oluşturulur. Öğretmenin sicil numarası bu sorgunun parametresine eklenir. Genel parametreleri almak için bir başka sorgu oluşturulur. İlgili sorgular çalıştırılır ve sonuçlar ResultSet içine yerleştirilir. Boş bir tablo modeli oluşturulur. Tablo modeline sütunlar eklenir. Öğretmenin ve genel parametrelerin derslerinin bulunduğu diziler oluşturulur. Döngü içinde, genel parametrelerdeki her ders için öğretmenin derslerinde var olup olmadığı kontrol edilir. Her ders için, öğretmenin o dersi verip vermediği durumu belirleyerek tabloya eklenir. Son olarak, kullanılan tüm ResultSet'ler ve Statement'lar kapatılır, tablo arayüzü için gerekli bileşenler eklenir ve ekranda gösterilir. App(Connection connection) metodu ile Genel olarak, bir Swing tabanlı bir kullanıcı arayüzü oluşturmak için bir başlangıç noktası sağlanmış ve bu nesne oluşturulduğunda pencerenin özelliklerini ve görünürliğini yapılandırılmıştır. public static void main(String[] args) Bu metod , Java programlarının başlangıç noktası olarak tanımlanan ve

uygulamanın yürütülmesine başlanan metoddur. metod içerisinde App nesnesi oluşturularak veri tabanı bağlantısı parametre olarak verilmiştir. actionPerformed(ActionEvent e) metodu ile Bu metod, actionPerformed metodunu içerir. Bu metod genellikle bir kullanıcı arayüzü (GUI) olaylarını (genellikle buton tıklamaları gibi) dinleyen ve bu olaylara tepki gösteren bir ActionListener tarafından uygulanır. Bu actionPerformed metodundaki yapı, bir dizi koşul içerir. Her bir koşul, e.getSource() ile alınan olayın kaynağını kontrol eder. Eğer olay belirli bir düğmeye tıklama gibi bir kullanıcı etkileşimiyle ilgiliyse, ilgili işlevi gerçekleştirir. Örneğin: Belirli butonların tıklanması durumunda, ilgili metodlar çağrılır (YoneticiGirisEkrani(), OgretmenGirisEkrani(), OgrenciGirisEkrani(), vs.). Yönetici, öğretmen ve öğrenci girişleri sağlandığında, veritabanına sorgu gönderip giriş işlemleri gerçekleştirilir. Bu işlemler veritabanından veri alıp karşılaştırma yaparak, doğru kullanıcı girişlerini kontrol eder. Yönetici panelinde parametrelerin kaydedilmesi, ilgi alanlarının düzenlenmesi, derslerin eklenmesi veya silinmesi, taleplerin listelenmesi ve onaylanması reddedilmesi gibi işlemler yapılmıştır. Öğretmen panelinde ders alma, ders bırakma, ilgi alanlarının düzenlenmesi gibi işlemler gerçekleştirilmiştir. Öğrenci panelinde ilgi alanlarının düzenlenmesi ve derslerin görüntülenmesi gibi işlemler gerçekleştirilmiştir. Bu metod, kullanıcı arayüzünden gelen olaylara yanıt vermek üzere tasarlanmıştır. Örneğin, bir buton tıklandığında belirli işlevleri gerçekleştirmek gibi. Bu yapı, genellikle Swing veya JavaFX gibi GUI kütüphanelerinde kullanılır ve kullanıcı etkileşimine yanıt vermek için olay dinleme ve işleme yetenekleri sunar. if(e.getSource() == yoneticiLoginButonu) kod bulaogu kullanılarak bir kullanıcının girdiği yönetici şifresini, veritabanındaki kayıtlı şifreyle karşılaştırarak doğrulamaya yönelik yapılmıştır. Eğer giriş doğrulanırsa, yönetici ekranına geçişi sağlar. Eğer doğrulama başarısız olursa veya ilgili kayıt bulunamazsa, kullanıcıya uygun mesajlarla bildirim yapılmıştır. ifadesi, veritabanından alınan şifrenin, giriş yapılan şifre ile karşılaştırılmasını sağlamıştır. Eğer şifreler eşleşirse, YoneticiEkrani(); fonksiyonu çalıştırılmıştır. Her bir if bloğu, kullanıcının farklı bir etkinlik gerçekleştirmesine tepki verir. Örneğin, bir butona tıklama, veritabanı sorgusu yapma, kullanıcı girişi kontrolü veya başka bir pencereyi açma gibi işlevler gerçekleştirebilir. Bu kod parçaları genellikle, kullanıcı arayüzü işlevselliğini dinamik olarak yöneten ve kullanıcının etkileşimde bulunduğu uygulamalarda kullanılır. Yapılan işlemler, kullanıcıların butonlara tıklaması veya belirli etkileşimlerde bulunması üzerine gerçekleştirilir. OgrenciNotlariGoruntule() metodu ile öğrenci kendi notlarını görebilmektedir. OgrenciNotlariGoruntule butonu arayüzde öğrenci kendi notlarını görüntüleyebiliyorken ogrenciNotlariSil butonu tüm notları silebilmektedir.

IV. DENEYSEL SONUÇLAR

PostgreSQL Nedir? Altta yatan güçlü teknolojisi sayesinde karmaşık ve yüksek hacimli veri işlemleri gerçekleştiren işletmelerin ilk tercihi haline gelmiştir. Birkaç yazar ve

okuyucunun aynı anda sistem üzerinde çalışmasına izin veren çok sürümlü eşzamanlılık denetimi (MVCC) kullanır. PostgreSQL, aynı anda ve verimli bir şekilde birden fazla görevi yerine getirme konusunda son derece yeteneklidir. Sonuç olarak, Apple, Yahoo!, Facebook ve Instagram gibi şirketlerin yanı sıra telekom firmaları, finans ve devlet kurumları tarafından da kullanılır. PostgreSQL, standartlara uyumlu, yüksek düzeyde programlanabilir ve genişletilebilir olma hedefiyle çerçevelenmiştir. Bu DBMS'nin zorlayıcı doğasına rağmen birçok üçüncü taraf araç ve kitaplık, PostgreSQL ile çalışmayı basitleştirir. Diğer özellikler arasında yüksek eşzamanlılık, ACID uyumluluğu (Atomicity, Consistency, Isolation, Durability), nesne ilişkisi, SQL uyumluluğu (çoğunlukla) ve aktif bir geliştirici topluluğundan destek bulunur. PostgreSQL'in Avantajları Nelerdir? 1. Açık kaynak 2. Güçlü bir topluluk 3. Güçlü üçüncü taraf desteği 4. Genişletilebilirlik

PostgreSQL, ilişkisel bir veritabanı yönetim sistemidir ve standart SQL sorgularını destekler. PostgreSQL'de kullanılacak bazı temel SQL sorgu türleri şunlardır: SELECT: Veri okumak için kullanılır. SELECT sorgusu, belirli sütunlardan veri almayı sağlar.

INSERT: Yeni veri eklemek için kullanılır. INSERT sorgusu, bir tabloya yeni bir kayıt ekler. UPDATE: Varolan veriyi güncellemek için kullanılır. UPDATE sorgusu, belirli koşullara uyan kayıtların sütun değerlerini günceller. DELETE: Veri silmek için kullanılır. DELETE sorgusu, belirli koşullara uyan kayıtları siler.

ALTER TABLE: Bir tablonun yapısını değiştirmek için kullanılır. ALTER TABLE sorgusu, tabloya yeni bir sütun eklemek, mevcut bir sütunu silmek veya sütunların veri türlerini değiştirmek gibi işlemleri yapmayı sağlar. CREATE TABLE: Yeni bir tablo oluşturmak için kullanılır. CREATE TABLE sorgusu, yeni bir tablonun adını ve sütunlarının özelliklerini belirtir. DROP TABLE: Bir tabloyu silmek için kullanılır. DROP TABLE sorgusu, belirtilen tabloyu veritabanından kalıcı olarak kaldırır. JOIN: İki veya daha fazla tabloyu birleştirmek için kullanılır. JOIN sorguları, ilişkili tablolardan veri almayı sağlar. GROUP BY ve HAVING: Verileri gruplamak ve gruplar üzerinde işlem yapmak için kullanılır. GROUP BY ve HAVING sorguları genellikle toplama işlemleriyle birlikte kullanılır. ORDER BY: Sonuçları belirli bir sıraya göre sıralamak için kullanılır. ORDER BY sorgusu, sonuçları artan veya azalan sırada sıralar. Bu temel sorguların yanı sıra PostgreSQL, özel veri türleri, işlevler, prosedürler ve çok daha fazlasını destekler. PostgreSQL'in sunduğu özellikler, karmaşık sorguları ve veritabanı işlemlerini gerçekleştirmek için geniş bir araç seti sunar.

V. SONUÇ

Projenin genelinde Bir üniversitede açılan proje derleri için talep kriter temelinde öğrenci hoca eşleştirilmesi yapacak bir sitem geliştirilerek öğrencilerin veritabanı derslerinde edindikleri bilgilerin pekiştirilmesi , veri tabanı tasarımı ve sorgu yazma becerilerinin geliştirilmesi , pratik sağlanması amaçlanmaktadır. Bu amaç doğrultusunda nesneye yönelik

programlamanınve postgreSQL in kural ve medotları kullanılarak genel bir öğrenme gerçekleştiği ve amaca ulaşıldığı söylenebilir.

VI. KAYNAKÇA

- A. <https://www.udemy.com/course/postgresql-sifirdan-ileri-seviyeye-tum-yonleriyle-ogrenin/learn/lecture/36916534?start=15overview>
- B. <https://www.udemy.com/course/sifirdan-ileri-seviyeye-komple-java-gelistirici-kursu/learn/lecture/8595366?start=90overview>
- C. https://www.youtube.com/watch?v=lC8xh60CIG4list=PLKnjBH2xXNOoo9pcx5Dw-6zOyk1rwyOab_c_hannel = Murat
- D. https://www.youtube.com/watch?v=o9dcSS82gwab_c_hannel = NamasteCoding
- E. <https://bulutistan.com/blog/postgresql-nedir/>
- F. <https://www.hosting.com.tr/blog/postgresql/>
- G. https://tubitak-bilgem-yte.github.io/pg-gelistirici/docs/02-sorgu-yapisi-davranislari/sorgu_tipleri/
- H. <https://medium.com/sql-thttps://blog.gunduzdanismanlik.com/index.php/?archives/6-PostgreSQL-de-Veritaban-Yaratmak-Ve-Basit-Sorgular.html>
- I. <https://www.postgresqltutorial.com/postgresql-jdbc/connecting-to-postgresql-database/>
- J. https://www.tutorialspoint.com/postgresql/postgresql_java.htm
- K. https://www.youtube.com/watch?v=OLmAZmBSwMoab_c_hannel = CodeJava
- L. https://www.youtube.com/watch?v=dkn9NOChFdQab_c_hannel = CodeCube
- M. <https://www.geeksforgeeks.org/postgresql-crud-operations-using-java/>
- N. <https://www.javaguides.net/2020/02/java-crud-operations-with-postgresql.html>
- O. <https://pdfbox.apache.org/>
- P. <https://www.tutorialspoint.com/pdfbox/index.htm>
- Q. <https://pdfbox.apache.org/2.0/examples.html>

Parametreler

id	integer
durum	asama E NN
ayni_hoca_mult	boolean
ayni_ders_mult	integer
talep_maks_karakter	integer NN
ilgi_alanlari	String[]
admin_sifresi	String NN
dersler	String[]

Anlasmalar

anlasma_no	integer
ogrenci_no	integer NN
ogretmen_no	integer NN
ders	String NN
gonderen	gonderen E NN
durum	talepDurumu E
ogrenci_mesaj	char[talep_maks_karakter]
hoca_mesaj	char[talep_maks_karakter]

Ogrenciler

ogrenci_no	integer
sifre	String NN
ad	String NN
soyad	String NN
ilgi_alanlari	String[]
not_belgesi	pdf
genel_not	float NN

Hocalar

sicil_numarasi	integer
sifre	String NN
ad	String
soyad	String
ilgi_alanlari	String[]
kontenjan_sayisi	integer
acilan_dersler	String[]

Notlar

ogrenci_no	integer
ders_no	String
harf_notu	char

kriter_formulleri_puanlari

kriter_no	integer
hoca_no	integer
ders	String
katsayi	integer

OrnekOgrenci

id	integer
ad	String NN
soyad	String NN
sifre	String NN

