

2023 Yılı 1. Dönem 1. Yazlab Projesi

Tayyib Okur

200202060

Kocaeli Üniversitesi Bilgisayar Mühendisliği İÖ
ultratayyib@gmail.com

Fatma Nur Kurt

210202003

Kocaeli Üniversitesi Bilgisayar Mühendisliği İÖ
kurtfatmanur8@gmail.com

Index Terms—Java, Java Swing, GUI,PostgreSql, SQL, Sorgu, SFW, Class, Method, ArrayList, Layout .

I. ÖZET

Bu rapor belgesi 2023-2024 Güz dönemi Yazılım Laboratuvarı - I 1. projeyi açıklamaya yönelik bilgiler içermektedir. Bu proje kapsamında bir öğrenci bilgi sistemi tarzında bir uygulama yapılması hedeflenmiştir. Bunu yaparken veritabanı becerilerimizin geliştirilmesi hedeflenmiştir

Programlama Dili: herhangi bir programlama dilini kullanarak Masaüstü uygulaması yapılması hedeflenmiştir Veri tabanı yönetim sistemi için PostgreSQL kullanılması zorunludur. Tasarlanacak veritabanı, için çeşitli tablolar oluşturulmalıdır. Bu tablolar temel olarak öğrenci, öğretmen ve admin bilgilerini tutmalıdır. Öğrenci için Öğrenci numarası, ad, soyad, ilgi alanları, anlaşma talep sayısı, anlaşma durumu, öğrenciye ait not durum belgesi, Derslerle ilgili bilgiler ve harf notları, genel not ortalaması bilgileri. Öğretmen için Sicil numarası, Ad, Soyad, ilgi alanları, Kontenjan bilgisi, verdiği dersler, kriter dersler bilgileri veritabanında tutulması zorunlu olan bilgilerdir. Bunların yanında ihtiyaca yönelik gerekli bilgiler de tutulmalıdır. Uygulama çeşitli işlevleri gerçekleştirebilmelidir. Bunlardan bazıları; Öğrenci transkriptini yükledikten sonra sistem transkripti okunmalı ve dersleri harf notlarıyla birlikte kayıt edebilmeli. Öğrenci kendi aldığı dersleri notlarıyla birlikte görüntüleyebilmeli. Öğrenci proje dersleri için hocalardan ders talebinde bulunabilmeli aynı zamanda hoca eğer talebi onaylamadıysa öğrenci talebini iptal edebilmeli. Ders talebi esnasında hocaya mesaj da gönderebilmeli. Öğrenci ilgi alanına göre öğretmeni filtreleyebilmeli. Öğretmen öğrenciden gelen ders talebini görüntüleyebilmeli, kabul ve ret seçenekleri olmalı ve isterse öğrenciye mesaj atabilmeli. Öğretmen isterse öğrenciye kendi dersini seçmesi için istekte bulunabilmeli Öğretmen öğrencileri seçmek için istediği derslerden bir formül puanı oluşturabilmeli ve oluşturduğu o puana göre öğrenciyi filtreleyebilmeli. Admin tüm öğretmen ve öğrenci bilgilerini yönetebilmeli, silme, ekleme ve güncelleme işlemlerini yapabilmelidir.

II. GİRİŞ

Proje tanımı ve kuralları

Bu uygulamada öğrenci, hoca ve yönetici girişleri olmalıdır. Her kullanıcının kendisine ait şifre ve id bilgileri vardır bu bilgileri kullanarak giriş yapmalıdır.

Öğrenci panelinde öğrenci transkript belgesini yükleyebilmeli ve o belgede bulunan dersler harf notlarıyla birlikte veritabanına kaydedilmelidir. Öğrenci açılmış olan proje derslerinden ve bu dersleri veren hocalardan ders almak için talepte bulunabilir. Ders talebinde bulunurken aynı zamanda hocaya yöneticinin belirlediği karakter sınırı kadarıyla mesaj da gönderebilir. Aynı zamanda kendi ilgi alanına göre hocaları filtreleme yapabilmeli. Eğer öğrenci talepte bulunursa ve bu talebi öğretmen henüz kabul etmemişse bu talebi iptal edebilmelidir. Öğrenci aldığı dersleri harf notları ile istediği zaman görüntüleyebilmelidir. Öğrenci panelinde gelen hocalar tarafından gelen talepleri de görüntüleyebilmelidir. Bu talepler için sadece kabul seçeneği olmalıdır.

Öğretmen panelinde hoca ilgi alanlarını seçebilmeli ve sisteme kaydedebilmeli. Öğrenciler tarafından gelen ders taleplerini görüntüleyebilmeli ve hocaların gelen talebi reddetme ve kabul etme seçenekleri olmalı. Eğer hoca kabul ederse o ders öğrenciye eklenmeli. Hoca isterse talep eden öğrencinin ders bilgilerini görüntüleyebilmelidir. Öğretmen kendi verdiği bir ders için öğrencilere dersi alması için talepte bulunabilmelidir. Bunu yaparken talepte bulunduğu dersi öğrencinin almamış olması gereklidir. Öğretmen bir formül puanı oluşturabilmelidir ve bu oluşturduğu puana göre öğrencileri filtreleyebilmelidir. Öğretmen eğer öğrenciden gelen bir mesaj varsa onu görüntüleyebilmelidir ve öğrenciye karşı bir mesaj gönderebilmelidir.

Yönetici panelinde tüm öğrenci ve hocaların bilgilerine erişim güncelleme ve silme işlemlerini gerçekleştirebilmelidir. İlgi alanlarının sisteme kaydedilmesini, güncellenmesi ve silinmesi sağlanmalıdır. Yönetici aynı zamanda 1. ve 2. aşamaya geçişi yönetmelidir. 1. aşamada öğrenciler ve hocalar aralarında anlaşarak ders seçimi yaparlar. 2. aşamada ise yönetici henüz bir ders seçimi yapmamış öğrencileri hocalara atayabilmelidir, bunu da 3 şekilde yapabilmelidir. 1-) Öğrencileri rastgele bir şekilde hocalara atama 2-) Öğrencileri ortalama puanlarına göre sıralayarak ve yine yöneticinin belirlediği sırada hocalara atayabilmeli 3-) Yöneticinin belli derslere göre oluşturduğu bir formüle göre öğrencileri sıralayarak yine yöneticinin belirlediği sırada hocalara atayabilmeli

III. YÖNTEM

İlk olarak 1 tane giriş paneli oluşturduk. Bu giriş panelinde öğrenci, hoca ve admin girişlerini ayrı kısımlarda oluşturduk.

Giriş yaparken kullanıcılardan id ve şifre aldık. Veritabanında eşleşen kullanıcı olduğunda girişlerini yaptık.

Öğrenci panelinde Öğrenci transkript belgesini pdf şeklinde yükleyerek almış olduğu dersleri harf notlarıyla birlikte sisteme yükleyebilir. Biz bu dersleri ve notları veritabanına kayıt ediyoruz. Pdf okumayı java da bulunan pdfBox kütüphanesi ile yapıyoruz. Öğrenci isterse almış olduğu dersleri harf notlarıyla birlikte görüntüleyebilir. Bu bilgileri veritabanında tutuyoruz ve bir butona basıldığında veritabanından çekip ekranda listeliyoruz. Öğrenci sistemde açılmış proje derslerini seçebilir isteği hocadan talepte bulunabilir. Ders seçimi yaparken hocaya mesaj da gönderebilir ve seçtiği hocanın ilgi alanını da görebilir, aynı zamanda öğrencinin ilgi alanına göre bir filtreleme de yapılabilir. Öğrenci bir hocadan talepte bulunduğu bu talebi veritabanında tutuyoruz ve anlaşılmadurumu özelliğini beklemede yapıyoruz. Eğer talep oluştururken mesaj gönderilmek istenirse gönderilen mesajı da veritabanında tutuyoruz. Eğer hocayı öğrenci ilgi alanına göre filtreleme yaparsak veritabanından hocaların ilgi alanlarını çekiyoruz ve öğrencinin ilgi alanı ile eşleşen hocaların ilgi alanı varsa sadece o hocaları listeliyoruz. Talep ettiğimiz bir dersi iptal de edebiliriz. Oluşturduğumuz taleplerden beklemede olanları veritabanından çekiyoruz ve listeliyoruz. Bu listenen taleplerden birini seçip iptal butonuna basarsak veritabanından o talep durumunu iptal edildi şeklinde değiştiriyoruz ve listeden siliyoruz. Öğrenciye hoca tarafından da talep gelebilir. Öğrenci bu talepleri görebilir ve öğrenci sadece kabul et seçeneği ile kanul edebilir. Eğer öğrenci kabul ederse veritabanında bu dersi öğrenciye ekliyoruz. Öğrenci hocadan gelen mesajları da görebiliyor.

Öğretmen panelinde; Öğretmene ilgi alanı seçmesi için bir buton koyduk. Bu butona tıklandığında ilgi alanları listeliyoruz ve öğretmen bu ilgi alanlarından istediğini seçip kendi ilgi alanlarına ekleyebiliyor. Biz bu ilgi alanlarını veritabanında tutuyoruz. Öğretmene öğrencilereden kendisine gelen ders taleplerini de gösteriyoruz. Bunu veritabanında tuttuğumuz taleplerden çekiyoruz ve öğretmene gösteriyoruz. Öğretmen gelen taleplerden öğrencileri kabul edebilir veya reddedebilir. Eğer kabul ederse öğrenciye o dersi ekliyoruz ve beklemede olan talep durumunu kabul edildi diye değiştiriyoruz. Eğer öğretmen reddederse talep durumu ret edildi diye değiştiriyoruz. Öğretmen aynı zamanda kendisine talepte bulunan öğrencilerin ders bilgilerini görüntüleyebilir. Öğretmen isterse kendi dersi için öğrencilere talepte bulunabilir. Biz burada veritabanından ders seçmemiş öğrencileri listeliyoruz ve öğretmene bunları gösteriyoruz. Aynı zamanda bu listede öğretmenin oluşturduğu formül puanlarını da gösteriyoruz. Eğer öğretmen formül puanına göre bir filtreleme yapmak isterse öğretmenden minimum bir formül puanı istiyoruz ve bu puanın üstünde olan öğrencileri listeliyoruz. Aynı zamanda öğretmene sıralama seçeneği de sunduk öğretmen eğer öğretmen sıralaya tıklarsa belirlediği formül puanına göre büyükten küçüğe bir sıralama gerçekleşir. Öğretmen öğrenciler için bir filtreleme yapabilmek için bir formül oluşturabilir. Bunun için bir buton koyduk ve bu butona tıklandığında bölümde verilen dersleri listeledik. Öğretmen bu derslerden

en fazla 5 tane olmak üzere istediği dersleri seçebiliyor ve bu seçtiği dersler için hocadan seçtiği her ders için bir katsayı belirlemesini istedik ve bu dersleri ve derslerin katsayılarını veritabanında tuttuk. Öğretmen panelinde bir de mesaj kutusu koyduk. Bu mesaj kutusunda öğrencilerden gelen mesajları veritabanından çekip öğretmene gösteriyoruz. Öğretmen isterse bu mesajlara cevap verebiliyor.

Admin panelinde ise ilk başta otomatik olarak öğrenci üretmek için bir buton koyduk. Bu butona tıklandığında 20 tane rastgele isim ve soyisim oluşturuyoruz. Bunlar bizim öğrencilerimiz oluyor. Bu öğrenciler için belirlenen ilgi alanları rastgele atanıyor. Aynı zamanda bölümde verilen derslerden 20 tane rastgele seçilerek bu öğrencilerin derslerine atanıyor. Harf notları yine rastgele olarak belirleniyor. Bu harf notlarından yola çıkarak öğrencilerin ortalamasını da hesaplayıp veritabanına kayıt ediyoruz. Yönetici tüm öğrencilerin ve hocaların bilgilerine erişebiliyor ve bu bilgileri güncelleyip silebiliyor. Yönetici aynı zamanda ders seçimi için 2. aşamaya geçişi sağlayabiliyor. Bunun için bir buton koyduk ve bu butona tıklandığında 2. aşamaya geçiliyor. Bu aşamada proje derslerini öğrencilere belirli şartlara göre atamamamız isteniyor. 1. şart rastgele atama - bu şarta göre rastgele bir hoca seçilerek ve bu hocanın verdiği proje derslerinden biri rastgele seçilerek öğrenciye ekleniyor. Veritabanında ilgili ders de öğrenciye ekleniyor. 2. şart öğrencinin ortalamasına göre atama - bu şarta göre öğrencilerin ortalamalarını veritabanından teker teker çekiyoruz ve ortalamalar büyükten küçüğe doğru olacak şekilde sıralıyoruz daha sonra yöneticiden öğretmenlerin sırası belirlemesini istiyoruz. Ve en yüksek ortalaması olan öğrenci yöneticinin belirlediği öğretmen sırasına göre 1. olan öğretmene atanıyor ve atama bu şekilde devam ediyor. 3. şart ise yöneticinin belirlediği formül puanına göre atama - bu şart için yöneticiden bölümde olan derslerden en fazla 5 tane olacak şekilde seçip seçtiği her ders için de bir katsayı ataması istiyoruz ve bu şekilde bir formül puanı oluşuyor. bu formüle göre öğrencilerin ortalamalarını küçükten büyüğe sıralıyoruz. Ve yine yöneticiden öğretmenlerin sırasını da istiyoruz. Bu şekilde öğretmenin 1. sırada belirlediği öğretmene formül puanına göre en yüksek ortalama sahip öğrenci atanmış oluyor.

IV. YALANCI KOD

-Giriş sayfası oluştur ve bu giriş sayfasında öğrenci, öğretmen ve admin için giriş kısımları oluştur. her bir giriş kısmı için id ve şifre bilgilerini al. Bu bilgileri veritabanında karşılaştırdı eğer eşleşme varsa eşleşen kişinin sayfasını aç.

Öğrenci sayfası için bir textfield belirle ve bu textfield için transkript pdf yolunu iste. daha sonra bu pdf'i metine çevir ve öğrencinin derslerini veritabanına kayıt et.

Öğrenci sayfasında öğrenci bilgilerini gör diye bir buton koy. Bu butona basıldığında öğrenci ilgili öğrencinin id bilgisini kullanarak veritabanından almış olduğu dersleri çek ve listeleyerek öğrenciye göster.

Öğrenci sayfasında ders seçimi adında bir buton koy ve butona tıklandığında bölümde verilen proje derslerini liste. bu listeleri butonlar halinde yap. Eğer bu derslerden birine

tıklanınca bu dersleri veren hocaları listele ve bu listenin altına talep et, öğrenci ilgi alanına göre filtrele, talep et ve mesaj gönder, hoca ilgi alanını gönder adında butonlar koy. Eğer bir hoca seçilip, talep et butonuna tıklanırsa seçilen ders seçilen hocadan talep edilmiş olsun ve veritabanına bu talep kayıt et. ilgi alanına göre filtrele butonuna tıklayınca öğrencinin ilgi alanıyla aynı ilgi alanına sahip öğretmenleri listele. talep et ve mesaj gönder butonunda talep etme işlemi yanında hocaya mesaj gönder ve bu mesajı veritabanında tut. Aynı zamanda mesajın karakter sınırını yöneticinin belirlediği sınır doğrultusunda kontrol et. Hoca ilgi alanını gör butonunda veritabanından hocanın ilgi alanlarını çek ve öğrenciye göster.

Öğrenci sayfasında talep iptal butonu koy. Bu butona tıkladığında öğrencinin talep etmiş olduğu ve hocanın onaylamadığı dersleri listele. Öğrenci bu derslerden birini seçip iptal butonuna tıklarsa bu dersi veritabanından talep iptal edildi olarak güncelle.

Öğrenci sayfasında gelen talepler adında bir buton koy ve hoca tarafından gelen talepleri listele. Öğrencinin bu talepleri kabul edebilmesi için kabul et butonu koy.

Öğrenci sayfasına gelen kutusunu koy. Öğrenciye gelen mesajları kimden geldiğini listele ve mesajları görüntüle.

Öğretmen sayfasında ilgi alanı seçimi butonu koy ve öğretmene seçmesi için bazı ilgi alanları sun öğretmen bu ilgi alanlarını seçtiğinde veritabanına kayıt et.

Öğretmen sayfasına gelen talepler adında bir buton koy ve bu butona tıkladığında öğrencilerden gelen ders taleplerini listele. Bu listenin sonuna kabul et, reddet ve bilgi görüntüle butonları koy. Kabul et butonuna tıklanırsa öğrenciye dersi ekle. Reddet butonuna tıklanırsa dersi ekleme. Bilgi görüntüle butonuna tıklanırsa seçilen öğrencinin almış olduğu dersler ile bunların notlarını görüntüle.

Öğretmen sayfasına öğrenci görüntüle butonu koy ve bu butona basıldığında öğretmenin vermiş olduğu dersleri listele bu derslerden birine tıkladığında o dersi almamış öğrencileri listele.

Öğretmen sayfasına öğrenci formül puanı oluştur adında buton koy ve bu butona basıldığında bölümde verilen dersleri listele bu dersler seçilerek bir formül puanı oluştur.

Öğretmen sayfasında gelen kutusu oluştur. ve öğrencilerden gelen mesajları burada listele. Mesaj gönder butonu da koy ve ilgili öğrenciye mesaj gönder.

Admin sayfasında öğrenci üret butonu koy. Bu butona basıldığında isimleri, soy isimleri, dersleri, harf notları ve ilgi alanları rastgele atanan öğrenciler üret ve veritabanına ekle

Admin sayfasında hocalar için kontenjan belirleyen bir textfield ve buton koy. ve bu şekilde veritabanında hocaların

kontenjan verisini güncelle.

Admin sayfasında aynı şekilde öğrencinin seçebileceği hoca sayısını ayarla.

Admin sayfasına 2. aşama butonu koy. Bu butona tıkladığında 2. aşamaya geçilsin. 3 seçenek sun 1. seçenek ders seçmeyen öğrencileri random şekilde random hocalara ata. 2. seçenek ortalamaya göre atama ders seçimi yapmayan öğrencilerin ortalamalarını sırala ve sırasını belirttiğin hocalara sırayla ata. 3. seçenek 2. seçenekteki aynı şeyi formül puanı oluşturarak yap.

V. DENEYSEL SONUÇLAR VE SONUÇ

Proje verildiğinde geçen sene aldığımız veritabanı dersinde öğrendiklerimizi geliştirmeye yönelik bir çalışma olduğunu anladık. Zaten veritabanı kullanarak bir proje yapmıştık. Bu proje de veritabanı bilgimizi daha da geliştirdi. Bu proje için daha önceki projelerde kullandığımız ve hakim olduğumuz java dilini kullanmaya karar verdik. Veritabanı uygulaması olarak postgresql kullanılması zorunlu idi biz de bunu kullandık. Zaten geçen seneki projeden tablo oluşturmaya ve gerekli sql sorgularını yazmaya alışık olduğumuz için pek bir zorluk çekmedik ve hızlıca gerekli işlemleri yaptık. Java ve postgresql arasındaki bağlantıyı kurmak için gerekli connection jdbc kütüphanesini internetten bulup projemize yükledik. Projemizde hem önceden kullandığımız Java Swing kütüphaneleri, hem de arayüz hakkındaki tecrübelerimiz işimizi kolaylaştırdı. Projeyi tamamladığımızda bu geçen süreçte Java, arayüzler, SQL sorguları ve veritabanı hakkında yeni bilgiler öğrenip tecrübeler kazanmış olduk. Yaptığımız her projenin bizi bir adım öne taşıdığının farkına vardık.

VI. NORMALİZASYON İŞLEMLERİ

Normalizasyon, veri fazlalığını azaltan ve Ekleme, Güncelleştirme ve Silme Anormallikleri gibi istenmeyen özellikleri ortadan kaldıran bir veritabanı tasarım tekniğidir. Normalleştirme kuralları, büyük tabloları daha küçük tablolara böler ve ilişkileri kullanarak bunları birbirine bağlar. SQL’de Normalleştirmenin amacı, gereksiz (tekrarlayan) verileri ortadan kaldırmak ve verilerin mantıksal olarak depolanmasını sağlamaktır. Projemizde ilk olarak 1.NF kurallarına göre tablolarımızı oluşturduk.

Birinci Normal Form (1NF): Bir tablo, 1NF’ye uygun olduğunda her bir hücrede yalnızca bir değer bulunur ve her bir sütun tek bir veri türünü temsil eder. Bu, verilerin atomik (bölünemez) olduğu anlamına gelir. Ayrıca, her bir satır benzersiz bir tanımlayıcıya sahip olmalıdır, böylece her bir satırı tek bir kaydı temsil eder. 1NF ayrıca, verilerin tekrarlılığını önlemek için her sütunda birincil anahtar (primary key) belirlenmesini gerektirir.

öğrenci, öğretmen,admin olmak üzere 1NF kurallarına uygun 3 adet tablo oluşturduk. Öğrenci tablomuz studentid,

ad, soyad, talep sayısı, ortalama, ilgi alanı, şifre, pdf,dersler, harf notları, seçilen hocalar olarak özellikleri ekledik.

Öğretmen tablomuz: id,isim, soyisim, kota, öğrenci sayısı, verdiği dersler, ilgi alanları, şifre

Admin tablomuz: id, şifre, karakter sayısı, talep miktar

İkinci Normal Form (2NF):

Bir tablo, 2NF'ye uygun olduğunda önceki 1NF koşullarını karşılar ve ek olarak kısmi bağımlılıkları ortadan kaldırır. Kısmi bağımlılık, bir tabloda birincil anahtarın yalnızca bir kısmını kullanarak başka bir sütuna bağımlılık olması durumudur. 2NF'ye uygun bir tabloda, her sütun, yalnızca tablonun tamamlayıcı bir kısmına bağımlı olmalıdır. Bunun için, gerektiğinde tablonun parçalanması ve ilgili sütunların bağımsız tablolara taşınması gerekebilir.

tablolarımız 2NF uyaması için bazı özellikler için yeni tablolar oluşturduk. Mesela öğrencinin aldığı dersler için ayrı bir tablo oluşturduk ve bu tabloda öğrencinin aldığı her ders ve aldığı harf notlarını tuttuk. Aynı şekilde Öğretmenin verdiği her dersi de ayrı bir tabloda tuttuk. Talep oluşturma işlemlerini tutabilmek için ayrı bir tablo oluşturmamız gerekiyordu. Bunun içinde öğretmen ve öğrenci arasındaki talep geçmişini ve o anki talep durumlarını tutabilmek için talep adında bir tablo oluşturduk. Bu tabloda talebi yapan kişinin id'si ve talep edilen kişinin id bilgisi tutuldu ve hangi ders için talep edildiği bilgisine de yer verildi. Talebin durumu için yine bir özellik bu veriyi tuttuk. Mesajlaşma için de bir tablo yapamamız gerekti bunun içinde bir tablo oluşturduk. Öğrencinin öğretmene veya öğretmenin öğrenciye gönderdiği mesajları bu tabloda tuttuk. Öğretmenler kendileri için bir formül oluşturabilir. Bu formülde yer alan dersler ve bu derslerin formül için belirlenen katsayılarını tutmak için yine bir tablo oluşturduk. Öğretmenlerin ilgi alanlarını tutmak için de ayrı bir tablomuz var. Bu tabloda ise öğretmen id bilgisi ve ilgi alanını tuttuk.

Üçüncü Normal Form (3NF):

Bir tablo, 3NF'ye uygun olduğunda önceki 2NF koşullarını karşılar ve ek olarak geçişli bağımlılıkları ortadan kaldırır. Geçişli bağımlılık, bir tabloda bir sütunun diğer bir sütuna dolaylı olarak bağımlı olduğu durumu ifade eder. 3NF'ye uygun bir tabloda, her sütun, yalnızca birincil anahtara doğrudan bağımlı olmalıdır. Bu, tablonun gerektiğinde parçalanması ve ilişkili sütunların farklı tablolara taşınması gerektiği anlamına gelebilir.

Tüm tablolarımız için aynı zamanda primary keylerini belirledik

GELİŞTİRME ORTAMI

Projemizi java dilinde gerçekleştirdik. Veritabanı yönetim sistemi için postgresql kullandık. Arayüz için ise Java Swing kullandık. Oyun alanımızı Layout mantığındaki gibi Butonlar kullanarak oluşturduk.

Kullandığımız kütüphaneler:

- javax.swing.*;
- javax.swing.border.LineBorder;

- java.awt.*;
- java.io.IOException;
- java.sql.PreparedStatement;
- java.util.ArrayList;
- java.io.File;
- java.awt.event.ActionListener;
- java.awt.event.ActionEvent
- javax.swing.text.DocumentFilter;
- javax.swing.text.BadLocationException;
- javax.swing.text.AttributeSet;
- javax.swing.text.AbstractDocument;
- javax.swing.event.ListSelectionListener;
- javax.swing.event.ListSelectionEvent;
- org.apache.pdfbox.text.PDFTextStripper;
- org.apache.pdfbox.pdmodel.PDDocument;
- java.util.Random;
- java.util.Collections;
- javax.swing.text.AttributeSet;

KAYNAKÇA

[1] postgresql kurulumu için ekte verilen Youtube videosu kullanılmıştır. [2] java ile Mysql bağlantısını sağlamak için ekte verilen kitap kullanılmıştır [3] SQL sorgularını javada kullanmak için [4] postgresql için gerekli kütüphaneyi yüklemek için ekte verilen link kullanılmıştır. [5] javada pdf okumak için gerekli olan pdfBox yapısını kullanabilmek için gerekli bileşenleri ekteki linkten elde ettik.

REFERENCES

- [1] <https://www.youtube.com/watch?v=hRnXI1qo0G0>
- [2] 2022 kodlab-java yeni başlayanlar için 27.baskı kodlab.
- [3] 2022 kodlab-java yeni başlayanlar için 27.baskı kodlab.
- [4] <https://jdbc.postgresql.org/download/>
- [5] <https://pdfbox.apache.org/>