



T.C
KOCAELİ SAęLIK VE TEKNOLOJİ ÜNİVERSİTESİ
MÜHENDİSLİK VE DOęA BİLİMLERİ FAKÜLTESİ
BİLGİSAYAR/YAZILIM MÜHENDİSLİęİ

PROJE KONUSU: BAŞARI HESAPLAMA ARAYÜZÜ

HAZIRLAYANLAR

AHMET EREN ŞENGÜL 220502036

BAŞAR BURAK ÜNAL 220501025

YUSUF USTAOęLU 220502003

EFE ÖZTAŞ 220501014

YİęİT SAMET KALKAN 220502042

DERS SORUMLUSU:

Dr. Öğr. Üyesi ELİF PINAR HACİBEYOęLU

TARİH: 22.01.2025

1 GİRİŞ

1.1 Projenin amacı

Bu proje, öğrenci başarı oranlarını belirlemek ve analiz etmek amacıyla bir arayüz oluşturmayı hedefler. Kullanıcılar ders bilgilerini, program ve ders çıktılarını, öğrenci notlarını ekleyerek çeşitli tablolar oluşturabilir.

Projenin Hedefleri:

1. Program ve ders çıktılarını kayıt altına almak.
2. Kullanıcıların ders çıktıları ile program çıktıları arasındaki ilişkileri tanımlamasını sağlamak.
3. Değerlendirme kriterlerine göre başarı hesaplamaları yapmak.
4. Öğrenci notlarını kaydedip analiz etmek.
5. Excel dosyalarından veri yükleme ve dışa aktarım özelliklerini sunmak.

2 GEREKSİNİM ANALİZİ

2.1 Arayüz gereksinimleri

- Kullanıcı dostu ve kolay anlaşılabilir bir menü tasarımı.
- Ders, program çıktıları ve öğrenci listesi yüklemek için Excel entegrasyonu.
- Veri girişleri için hata kontrolleri ve uyarı mesajları.
- Çıktıların kaydedilmesi ve raporlanması için Excel dışa aktarma seçenekleri.

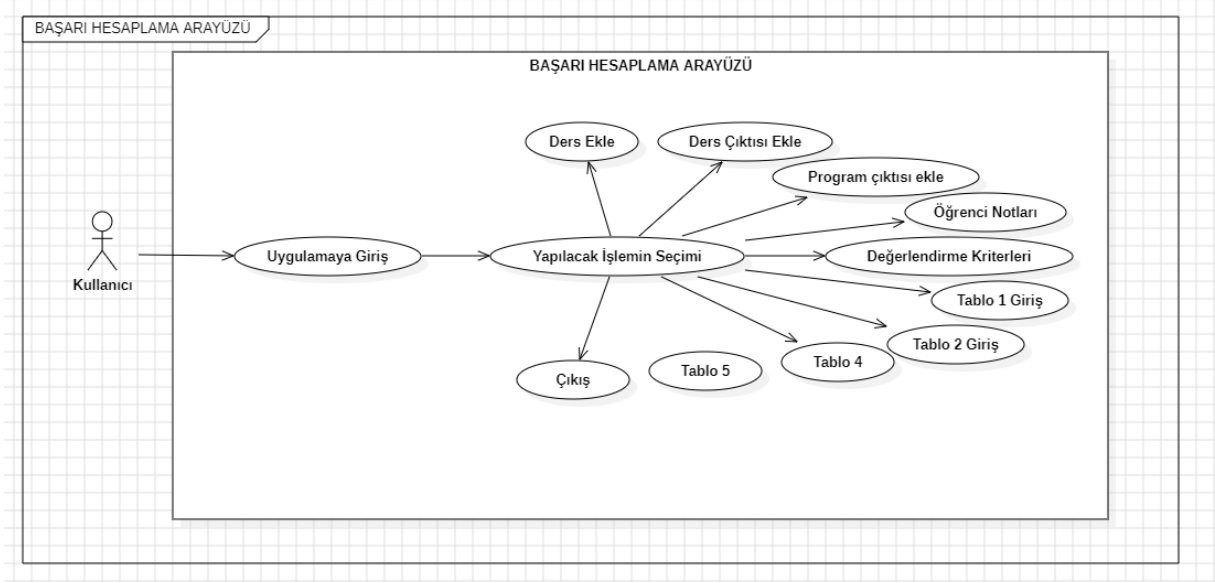
Donanım Gereksinimleri:

- Uygulamanın çalıştırılması için temel bir bilgisayar sistemi.
- Python 3.9 veya daha yeni bir sürüm.

2.2 Fonksiyonel gereksinimler

- Kullanıcının ders bilgilerini ekleyebilmesi ve düzenleyebilmesi.
- Program ve ders çıktılarının yüklenmesi.
- Değerlendirme kriterlerinin ve ağırlıklarının tanımlanması.
- Tablo 1 ve Tablo 2 veri analizlerinin gerçekleştirilmesi.
- Öğrenci notlarının girilmesi ve raporlanması

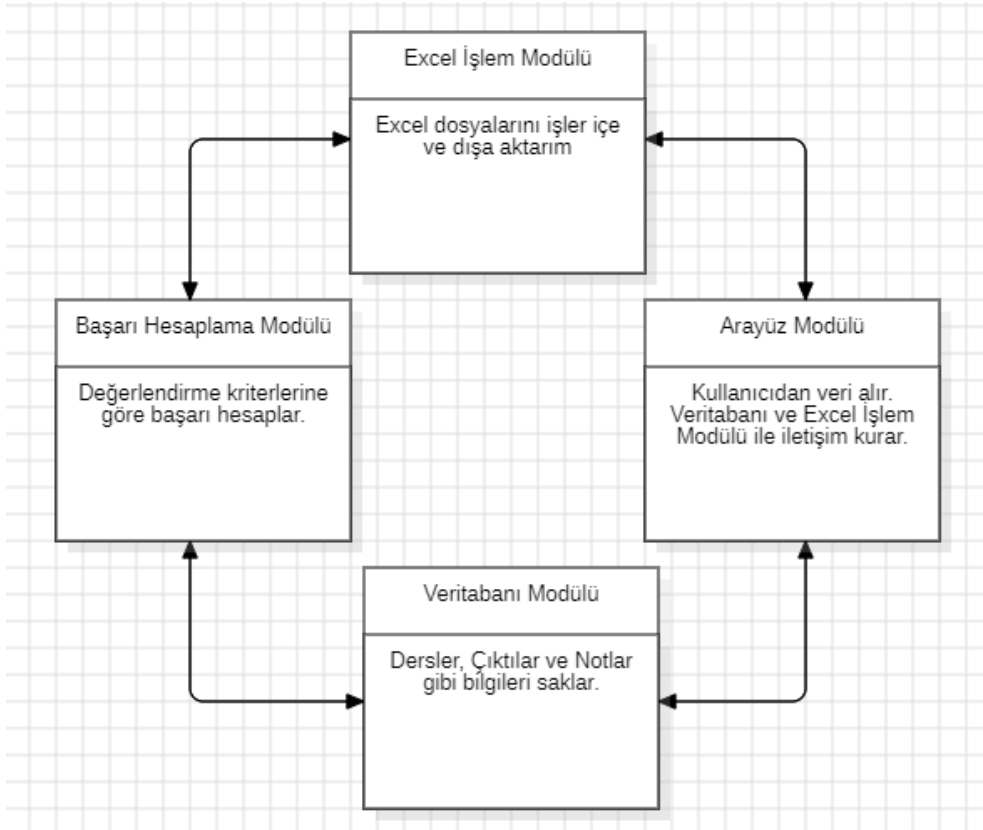
2.3 Use-Case diyagramı



3 TASARIM

3.1 Mimari tasarım

- Uygulama, veritabanı tabanlı bir yapıya sahiptir.
- Veritabanı, SQLite kullanılarak geliştirilmiştir.
- Arayüz, Tkinter kütüphanesi ile oluşturulmuştur.



3.2 Kullanılacak teknolojiler

- **Python** : Yazılım python dilinde yazılmıştır
- **Tkinter kütüphanesi** : Projede kullanılan arayüzü yapmamızı sağlayan kütüphane
- **SQLite** : Projenin veri tabanı için kullanılan kütüphane
- **Pandas ve openpyxl** : Excel ile yapılan dosya ekleme-çıkartma işlemleri için kullanılan kütüphaneler
- **JSON**: Verilerin kolayca saklanması ve işlenmesi için kullanılmıştır.
- **Excel Dosyaları**: Öğrenci listeleri ve not tablolarının giriş-çıkış işlemleri için kullanılmaktadır.

3.3 Veri tabanı tasarımı

SQLite veri tabanı kullanılmıştır. Veri tabanı, dersler, program çıktıları, ders çıktıları, değerlendirme kriterleri ve öğrenci notları gibi bilgileri tutacak şekilde tasarlanmıştır. Örneğin; Ders kodu, Program çıktısı kodu, ders çıktısı kodu, kriter ID ve öğrenci numarası.

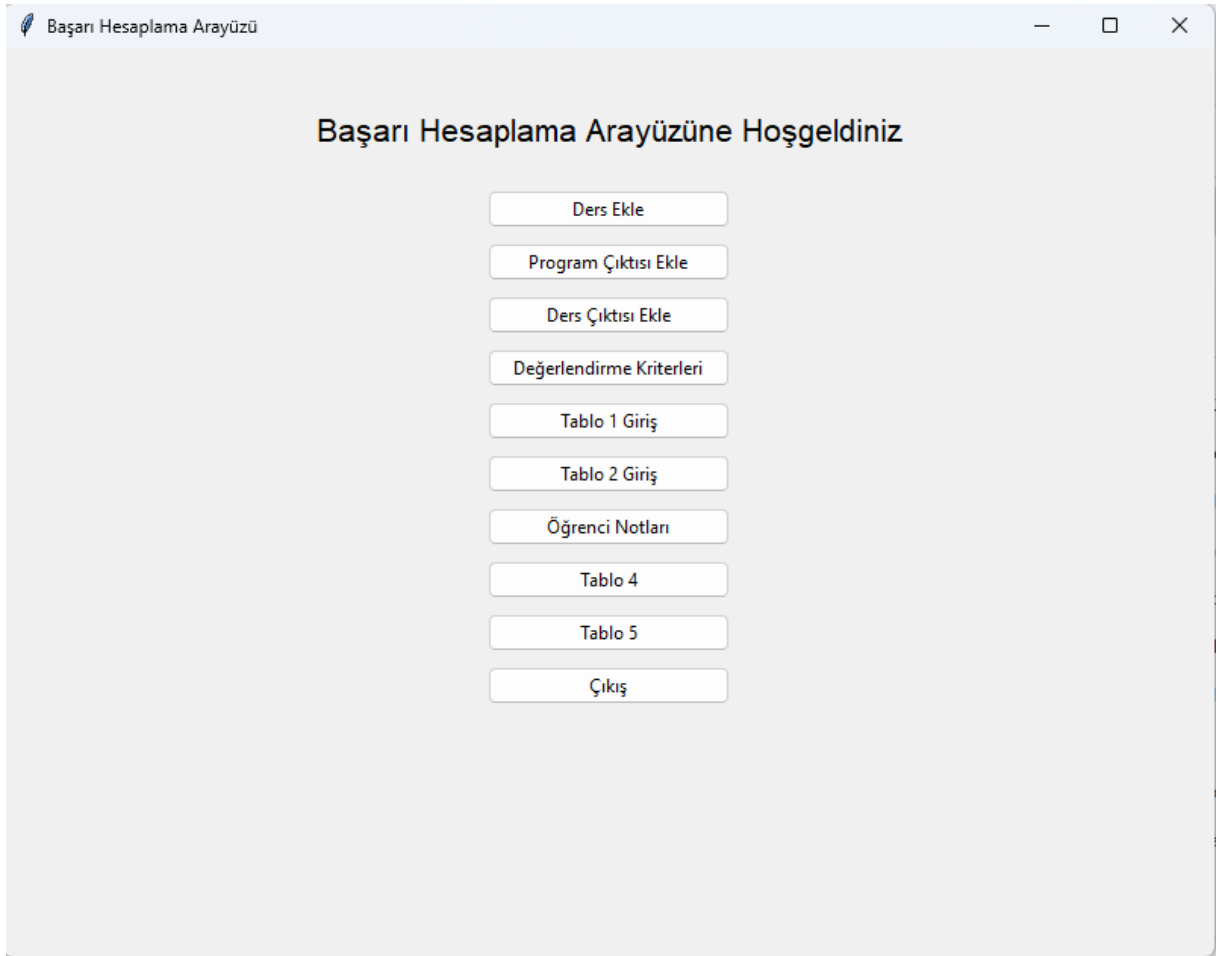
3.4 Kullanıcı arayüzü tasarımı

Uygulama, **Tkinter** üzerinde bir **ana menü** ve bunun altındaki **alt ekranlar** (frame) şeklinde kurgulanmıştır. Her tıklamada, bir önceki ekran `clear_frame()` metodu ile temizlenir ve yeni bir ekran (frame) oluşturulur.

Genel Arayüz Akışı:

1.Ana Menü

- "Ders Ekle", "Program Çıktısı Ekle", "Ders Çıktısı Ekle", "Değerlendirme Kriterleri", "Tablo 1 Giriş", "Tablo 2 Giriş", "Öğrenci Notları", "Tablo 4", "Tablo 5" ve "Çıkış" butonları vardır. Kullanıcı hangi butona tıklarsa o ekrana yönlendirilir.



2.Ders Ekle Ekranı

- Kullanıcıdan “Ders Kodu” ve “Ders Adı” bilgileri istenir. “Kaydet” ile veritabanına eklenir.

Başarı Hesaplama Arayüzü

Ders Ekle

Ders Kodu:

Ders Adı:

Kaydet

Ana Menüye Dön

3.Program Çıktısı Ekle Ekranı

- Kullanıcı, “Kaç tane program çıktısı gireceksiniz?” sorusunu yanıtlar. Sonra girilen sayı kadar metin kutusu oluşturulur ya da “Excel’den Yükle” butonuyla dışarıdan yükleme yapılabilir. Kaydedilen veriler `program_outcomes` tablosuna gider.

Başarı Hesaplama Arayüzü

Program Çıktısı Ekle

Kaç tane program çıktısı ekleyeceksiniz?

Mevcut Program Çıktıları:

PÇ1: 5
PÇ2: 6

Devam

Excel'den Yükle

Ana Menüye Dön

4.Ders Çıktısı Ekle Ekranı

- İlgili “Ders” seçildikten sonra, “Kaç tane ders çıktısı?” sorulur. Her bir çıktı metin halinde girilir veya Excel’den yüklenir. Kayıtlar `course_outcomes` tablosuna eklenir.

Başarı Hesaplama Arayüzü

Ders Çıktısı Ekle

Ders Seçiniz:

Kaç tane ders çıktısı ekleyeceksiniz?

Mevcut Ders Çıktıları:

Devam

Excel'den Yükle

Ana Menüye Dön

5.Değerlendirme Kriterleri Ekranı

- Ders seçimi yapıldıktan sonra, örnek olarak “Vize %30, Proje %20, Final %50” gibi kriterler tanımlanır. Kullanıcı `criteria_screen()` arayüzü üzerinden bu girdileri ekler, listbox’ta görebilir ve en sonunda “Kaydet” ile `criteria` tablosunda tutulur.

Başarı Hesaplama Arayüzü

Değerlendirme Kriterleri ve Ağırlıkları

Ders Seçin:

Kriter Seçin:

Ağırlık (%):

Ekle Seçili Kriteri Düzenle Kaydet Ana Menüye Dön

6.Tablo 1 Giriş (Program Çıktıları - Ders Çıktıları İlişkisi)

- Kullanıcı, seçili derse ait ders çıktılarını ve mevcut program çıktılarını bir matris şeklinde görür. Hücrelerde 0 ile 1 arasında bir değer girilir. Kaydedildiğinde relationship_values tablosu güncellenir.

Başarı Hesaplama Arayüzü

Tablo 1: Program Çıktıları - Ders Çıktıları İlişkisi

Ders:

Kaydet

İçeri Aktar

Dışarı Aktar

Ana Menüye Dön

7.Tablo 2 Giriş (Değerlendirme - Ders Çıktısı İlişkisi)

- Benzer şekilde, seçili derse ait **kriterler** ve **ders çıktıları** bir matris şeklinde görüntülenir. Her hücreye 0-1 arası değer girilir. Kaydedildiğinde `course_outcome_evaluations` tablosu güncellenir.

Başarı Hesaplama Arayüzü

Tablo 2: Değerlendirmeler - Ders Çıktıları İlişkisi

Ders:

Kaydet İçeri Aktar Dışarı Aktar Ana Menüye Dön

8.Öğrenci Notları Ekranı

- Ders ve kriterler seçili olduğunda, satır satır öğrenci isimleri ve not girişleri yapılır. “Ortalama” sütunu, girilen notların toplam ağırlıklarına göre hesaplanarak ekranda gösterilir. Kaydet ile student_grades tablosu güncellenir (JSON formatında not tutar).

Başarı Hesaplama Arayüzü

Öğrenci Notları

Ders Seçin:

Öğrenci Ekle

Excel'den Yükle

Kaydet

Dışarı Aktar

Ana Menüye Dön

9.Tablo 4 Ekranı (Ders Çıktıları Başarı Oranı)

- Sistemde tanımlı **Tablo 2** bilgisi (ÇK-Kriter), kriter ağırlıkları, ve öğrenci notlarını kullanarak her bir **Ders Çıktısı** için öğrencinin başarısını hesaplar. “Kaydet” seçeneğiyle `success_rates` tablosuna kaydedebilir, “Dışarı Aktar” ile Excel’e çıkabilir.

Başarı Hesaplama Arayüzü

— □ ×

Tablo 4: Ders Çıktıları Başarı Oranları

Ders: Öğrenci:

Kaydet Ana Menüye Dön

10.Tablo 5 Ekranı (Program Çıktıları Başarı Oranı)

- Tablo 4'teki ders çıktısı başarı oranları ve Tablo 1'deki PÇ-ÇK ilişkilerini birleştirerek **PÇ'lerin ne kadar başarılı olduğunu** gösterir. Seçili öğrenciye göre veriler ekrana gelir ve istenirse Excel'e aktarılır.

Başarı Hesaplama Arayüzü

— □ ×

Tablo 5: Program Çıktıları Başarı Oranları

Ders: Öğrenci:

Dışarı Aktar

Ana Menüye Dön

4 UYGULAMA

4.1 Kodlanan bileşenlerin açıklamaları

1. Veritabanı Modülü:

- **create_database()**: Veri tabanında gerekli tabloları oluşturur. Tablolar, dersler, program çıktıları, ders çıktıları ve öğrenci notları gibi bilgileri saklar.
- **save_course()**: Kullanıcıdan alınan ders bilgilerini veri tabanına kaydeder. Aynı ders kodunun tekrar eklenmesini engeller.

2. Arayüz Modülü:

- **create_main_menu()**: Kullanıcıların ana menüden ders ekleme, çıktı ekleme, not girişleri ve tablo oluşturma gibi işlemleri yapmasını sağlar.
- **add_course()**: Yeni bir ders eklemek için kullanıcıdan giriş alınmasını sağlar.
- **criteria_screen()**: Değerlendirme kriterlerini yönetmek için kullanıcı arayüzü sunar.

3. Excel Modülü:

- **load_program_outcomes()**: Kullanıcıların Excel dosyalarından program çıktıları yüklemesini sağlar.
- **export_to_excel()**: Tablo verilerini Excel formatında dışa aktarır.

Sınıflar ve Fonksiyonlar:

1- SuccessCalculationApp:

- Uygulamanın temel sınıfıdır. Tkinter tabanlı arayüzün tüm bileşenlerini ve işlevlerini yönetir.
- **Fonksiyonlar:**
 - **__init__()**: Arayüz bileşenlerini ve veritabanı bağlantısını başlatır.
 - **clear_frame()**: Aktif arayüz ekranını temizler.
 - **upload_data()**: Veri yükleme işlemleri için bir ekran sunar.

2- validate_input(P):

- Kullanıcıdan alınan verilerin doğruluğunu kontrol eder. Örneğin, değerlerin 0 ile 1 arasında olup olmadığını denetler.

3- save_grades():

- Öğrenci notlarını JSON formatında veritabanına kaydeder.

4- generate_table_4() ve generate_table_5():

- Tablo 4 ve Tablo 5'i oluşturup Excel formatında kaydeder.

4.2 Görev dağılımı

- Ekip olarak, başarı hesaplama arayüz projesinin her bir aşamasında birlikte ilerledik; kodun ve dosyalarının geliştirilmesi gibi işlemleri birlikte gerçekleştirdik.
- Rapor ekip üyeleri arasında iletişimi koparmadan birlikte hazırlanmıştır.

4.3 Karşılaşılan zorluklar ve çözüm yöntemleri

- **Kullanıcıdan alınan verilerin doğruluğunun sağlanması:** Kullanıcıların girdiği verilerin geçerliliğini kontrol etmek için veri doğrulama yöntemleri geliştirilmiştir. Özellikle, sayısal değerlerin belirli bir aralıkta olup olmadığını kontrol etmek için `validate_input` fonksiyonu kullanılmaktadır.
- **Excel dosyalarının farklı formatlarda olması:** Farklı Excel dosya biçimlerini desteklemek için `pandas` ve `openpyxl` gibi kütüphaneler entegre edilmiştir. Dosya içeriği standart hale getirildikten sonra işleme alınmaktadır.
- **Arayüzü oluşturmak için Tkinter'in kullanılması:** Tkinter'in sınırlı bileşenleri nedeniyle, kullanıcı dostu ve işlevsel bir arayüz oluşturmak zaman almıştır. Bu zorluk, bileşenlerin düzenli bir şekilde organize edilmesi ve grid metodunun etkili kullanımıyla çözülmüştür.
- **Veritabanı bağlantısında performans sorunları:** Çoklu tablo işlemleri sırasında performansı optimize etmek için veritabanı sorguları düzenlenmiş ve indeksler kullanılmıştır.
- **Tablo verilerinin doğru şekilde dışa aktarılması:** Excel'e veri aktarma sırasında formatlama ve veri kaybını önlemek için her sütun tipi manuel olarak tanımlanmıştır.

4.4 Proje isterlerine göre eksik yönler

- Projede eksik herhangi bir yön yoktur.

5 TEST VE DOĞRULAMA

5.1 Yazılımın test süreci:

Yazılımın test sürecinde, program çıktıları, ders çıktıları, ilişki tabloları ve öğrenci notlarının rastgele veri üreten fonksiyonlarla oluşturulması test edilmiştir. Üretilen veriler Excel dosyalarına kaydedilmiş ve her bileşenin doğru çalıştığı manuel olarak doğrulanmıştır. Tüm modüller, beklenen çıktıları başarıyla üretmiştir.

```
6 import pandas as pd
import numpy as np
import random

def create_program_outcomes():
    """Program çıktıları için örnek veri oluştur"""
    # Rastgele sayıda program çıktısı (4-8 arası)
    num_outcomes = random.randint(4, 8)

    base_outcomes = [
        "Matematik, fen bilimleri ve kendi dalları ile ilgili
        mühendislik konularında yeterli bilgi birikimi",
        "Problem tanımlama, modelleme ve çözme becerisi",
        "Bir sistemi, sistem bileşenini ya da süreci analiz etme
        becerisi",
        "Disiplin içi ve çok disiplinli takımlarda etkin biçimde
        çalışabilme becerisi",
        "Mühendislik problemlerini saptama, tanımlama, formüle etme ve
        çözme becerisi",
        "Mesleki ve etik sorumluluk bilinci",
        "Etkin iletişim kurma becerisi",
        "Mühendislik uygulamalarının evrensel ve toplumsal boyutlarda
        etkisini kavrama"
    ]

    # Rastgele seçilen sayıda çıktı al
    outcomes = random.sample(base_outcomes, num_outcomes)

    # DataFrame oluştur
    df = pd.DataFrame(outcomes)
    df.to_excel('program_ciktilari.xlsx', index=False, header=False)
    print(f"{num_outcomes} adet program çıktısı oluşturuldu:
    program_ciktilari.xlsx")
    return num_outcomes

def create_course_outcomes():
    """Ders çıktıları için örnek veri oluştur"""
    # Rastgele sayıda ders çıktısı (3-6 arası)
    num_outcomes = random.randint(3, 6)

    base_outcomes = [
        "Temel programlama kavramlarını anlama ve uygulama",
        "Algoritma geliştirme ve problem çözme becerisi",
        "Nesne yönelimli programlama prensiplerini uygulama",
        "Veri yapılarını anlama ve kullanma",
        "Temel veritabanı işlemlerini gerçekleştirme",
        "Web teknolojilerini kullanabilme becerisi"
    ]
```

```

# Rastgele seçilen sayıda çıktı al
outcomes = random.sample(base_outcomes, num_outcomes)

# DataFrame oluştur
df = pd.DataFrame(outcomes)
df.to_excel('ders ciktilari.xlsx', index=False, header=False)
print(f"{num_outcomes} adet ders çıktısı oluşturuldu:
ders_ciktilari.xlsx")
return num_outcomes

def create_table1(num_program_outcomes, num_course_outcomes):
    """Tablo 1 için örnek veri oluştur (Program Çıktıları - Ders
    Çıktıları İlişkisi)"""
    data = []

    for i in range(num_program_outcomes):
        row = [f"PÇ{i+1}"] # Program çıktısı numarası
        # Ders çıktıları için değerler (0-1 arası)
        values = np.round(np.random.uniform(0, 1, num_course_outcomes),
2)
        row.extend(values)
        data.append(row)

    columns = ['PÇ No'] + [f'ÇK{i+1}' for i in
range(num_course_outcomes)]
    df = pd.DataFrame(data, columns=columns)
    df.to_excel('tablo1.xlsx', index=False)
    print("Tablo 1 oluşturuldu: tablo1.xlsx")

def create_table2(num_course_outcomes):
    """Tablo 2 için örnek veri oluştur (Değerlendirmeler - Ders
    Çıktıları İlişkisi)"""
    # Rastgele sayıda değerlendirme kriteri oluştur (3-5 arası)
    num_criteria = random.randint(3, 5) # Maksimum 5 kriter

    # Temel kriterler
    criteria_groups = {
        "Ödev": ["Ödev", "Ödev 2"],
        "Quiz": ["Quiz"],
        "Proje": ["Proje"],
        "Vize": ["Vize"],
        "Final": ["Final"]
    }

    # Ders çıktısı içerikleri
    course_outcomes = [
        "Temel programlama kavramlarını anlama ve uygulama",
        "Algoritma geliştirme ve problem çözme becerisi",
        "Nesne yönelimli programlama prensiplerini uygulama",
        "Veri yapılarını anlama ve kullanma",
        "Temel veritabanı işlemlerini gerçekleştirme",
        "Web teknolojilerini kullanabilme becerisi"
    ]

    # Rastgele kriter grupları seç
    selected_groups = random.sample(list(criteria_groups.keys()),
min(num_criteria, len(criteria_groups)))
    selected_groups.sort() # Grupları sırala

    # Seçilen gruplardan kriterleri oluştur
    selected_criteria = []

```

```

    for group in selected_groups:
        # Her grup için kaç kriter alınacağını belirle (1 ile o
        # gruptaki maksimum kriter sayısı arasında)
        num_group_criteria = random.randint(1,
        len(criteria_groups[group]))
        # Gruptan sıralı olarak kriter seç
        group_criteria = criteria_groups[group][:num_group_criteria]
        selected_criteria.extend(group_criteria)

        # Toplam kriter sayısı 5'i geçerse, fazla kriterleri kaldır
        if len(selected_criteria) > 5:
            selected_criteria = selected_criteria[:5]
            break

    # Ders çıktılarından rastgele seç
    selected_outcomes = random.sample(course_outcomes,
    num_course_outcomes)

    data = []
    for i in range(num_course_outcomes):
        row = [f"ÇK{i+1}", selected_outcomes[i]] # Ders çıktısı
        numarası ve içeriği
        # Değerlendirme kriterleri için değerler (0-1 arası)
        values = np.round(np.random.uniform(0, 1,
        len(selected_criteria)), 2)
        row.extend(values)
        data.append(row)

    columns = ['ÇK No', 'Ders Çıktısı'] + selected_criteria
    df = pd.DataFrame(data, columns=columns)
    df.to_excel('tablo2.xlsx', index=False)
    print(f"Tablo 2 oluşturuldu ({len(selected_criteria)} kriter):
    tablo2.xlsx")
    return selected_criteria

def create_student_grades(criteria):
    """Öğrenci notları için örnek veri oluştur"""
    # Rastgele sayıda öğrenci (4-15 arası)
    num_students = random.randint(4, 15)

    # Öğrenci numaralarını oluştur (220502000'den başlayarak)
    base_number = 220502000
    # Rastgele numaralar üret (tekrar etmeyen)
    student_numbers = random.sample(range(base_number, base_number +
    1000), num_students)
    student_numbers.sort() # Numaraları sırala

    data = []
    for student_no in student_numbers:
        # Her değerlendirme için 0-100 arası rastgele notlar
        grades = np.random.randint(40, 101, len(criteria))
        row = [str(student_no)] + list(grades)
        data.append(row)

    # Tablo 2'deki kriterlerle aynı sırada olmasını sağla
    columns = ['Öğrenci'] + criteria
    df = pd.DataFrame(data, columns=columns)
    df.to_excel('ogrenci notlari.xlsx', index=False)
    print(f"{num_students} öğrenci için notlar oluşturuldu:
    ogrenci_notlari.xlsx")

```

```
print("Test verileri oluşturuluyor...")
num_program_outcomes = create_program_outcomes()
num_course_outcomes = create_course_outcomes()
create_table1(num_program_outcomes, num_course_outcomes)
criteria = create_table2(num_course_outcomes)
create_student_grades(criteria)
print("\nTüm test verileri oluşturuldu!")
```

6.1 Yazılımın doğrulanması

Program Çıktılarının Doğruluğu:

- Kod, rastgele sayıda (4-8 arası) program çıktısı üretmekte ve bunları bir Excel dosyasına (program_ciktilari.xlsx) kaydetmektedir.
- Üretilen çıktılar, kodun sonunda doğrulama amacıyla ekrana yazdırılır.
- **Test sonucu:** Program çıktıların doğru bir şekilde oluşturulduğu ve dosyaya kaydedildiği gözlemlendi.

Ders Çıktılarının Doğruluğu:

- Ders çıktıları rastgele seçilen bir listeden (3-6 arası) alınarak bir Excel dosyasına (ders_ciktilari.xlsx) kaydedilmektedir.
- Dosya, her çıktının içeriğini doğru şekilde içermektedir.
- **Test sonucu:** Ders çıktıları eksiksiz bir şekilde oluşturulmuş ve dosyaya kaydedilmiştir.

Tablo 1: Program Çıktıları - Ders Çıktıları İlişkisi:

- Program çıktıları ve ders çıktıları arasındaki rastgele ilişki değerleri (0-1 arası) hesaplanmış ve bir Excel dosyasına (tablo1.xlsx) kaydedilmiştir.
- Dosya, tüm ilişki değerlerini ve satır/sütun başlıklarını eksiksiz içermektedir.
- **Test sonucu:** Tablo 1'in oluşturulmasında bir sorun gözlemlenmedi.

Tablo 2: Değerlendirmeler - Ders Çıktıları İlişkisi:

- Rastgele seçilen değerlendirme kriterleri ile ders çıktıları arasındaki ilişki değerleri (0-1 arası) hesaplanmış ve bir Excel dosyasına (tablo2.xlsx) kaydedilmiştir.
- Kriterlerin sayısı 5'i geçmemekte ve sıralama doğru şekilde yapılmaktadır.
- **Test sonucu:** Tablo 2'nin oluşturulması başarılı şekilde tamamlanmıştır.

Öğrenci Notlarının Doğruluğu:

- Rastgele öğrenci numaraları ve değerlere dayalı notlar oluşturulmuş, kriterlere uygun şekilde bir Excel dosyasına (ogrenci_notlari.xlsx) kaydedilmiştir.
- Dosyada tüm öğrenciler ve notlar doğru şekilde sıralanmıştır.
- **Test sonucu:** Öğrenci notlarının doğru şekilde ürettiği anlaşılmıştır.

KAYNAKÇA

<https://www.geeksforgeeks.org/python-gui-tkinter/>

<https://www.geeksforgeeks.org/json-load-in-python/>

<https://docs.python.org/3/library/sqlite3.html>

<https://www.geeksforgeeks.org/working-with-excel-files-using-pandas/>

GITHUB LINKLERİ

Ahmet Eren Şengül <https://github.com/Eren1213>

Yusuf Ustaoglu <https://github.com/katlicia>

Efe Öztaş <https://github.com/EOztas>

Başar Burak Ünal <https://github.com/BasarBurakUnal>

Yiğit Samet Kalkan <https://github.com/yigitkalkan>