Adı Soyadı: Ahmed Said Kılıç Adı Soyadı: Yunus Alp Turan

Numara:170421015 Numara:171421001

T.C. MARMARA ÜNİVERSİTESİ TEKNOLOJİ FAKÜLTESİ 2023-2024 EĞİTİM ÖĞRETİM YILI GÜZ DÖNEMİ BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ VERİTABANI YÖNETİM SİSTEMLERİ DERSİ PROJE DÖKÜMANI

Açıklamalar

- Ödevinizi gönderirken tek bir dosya şeklinde **OğrenciNo_AdSoyad_VTYS_FinalÖdevi.docx,** olmasına dikkat ediniz.
- Google Classroom platformlarına 27 Aralık 2023 tarihine kadar yükleyiniz.
- Tüm sorular soru metninin altındaki alanlara cevaplandırılacaktır. Bu belgenin formatını bozmayınız, sadece gerekli alanları doldurunuz.

1. **Gerçekleştirdiğiniz veri tabanı projesi için grup arkadaşlarınız var ise isimlerini yazınız ve projenize ait veri tabanı/diğer yazılım bileşenleri hakkında bilgi veriniz. (10 p)**

Ahmed Said Kılıç
170421015
kilicsaud1221@gmail.com

Yunus Alp Turan
171421001
yunus.alp@marun.edu.tr

2. **Gerçekleştirdiğiniz projenin amacını detaylı bir şekilde açıklayınız. (Proje konularınızın yer aldığı classroom excel dosyası içerisindeki yazdığınız açıklama kabul edilmeyecektir.) (10 p)**

Bu proje, scooter alıcılarına yönelik kapsamlı bir platform oluşturmayı hedefler. Bu platform, scooter almak isteyen kullanıcılara geniş bir ürün yelpazesi sunar ve kullanıcıların ihtiyaçlarına uygun scooterları bulmalarını kolaylaştırır.

1. **Ürün Karşılaştırma ve Özelliklerin İzlenmesi:** Kullanıcılar, scooterların teknik özelliklerini, markalarını ve modellerini kolayca karşılaştırabilir. Bu özellikler arasında menzil, maksimum hız, taşıma kapasitesi, motor gücü gibi detaylar yer alır. Böylece kullanıcılar, ihtiyaçlarına en uygun scooterı seçebilir.
2. **Fiyat Karşılaştırması ve En İyi Fiyatı Bulma:** Platform, farklı satıcıların sunduğu fiyatları izler ve kullanıcılara farklı sitelerdeki fiyatları karşılaştırma imkanı sunar. Kullanıcılar, aynı scooterın farklı satıcılar arasındaki fiyat farklarını görerek en uygun fiyatı bulabilirler.
3. **Kullanıcı Dostu Arayüz ve Favori Ürünler:** Kullanıcılar, kendi hesapları üzerinden favori scooter modellerini kaydedebilir, bu scooterların fiyatlarını takip edebilir ve istedikleri zaman bu favori ürünlerin güncel fiyatlarını görebilirler. Böylece alışveriş yapmadan önce karar vermeleri kolaylaşır.
4. **Kullanıcı Geri Bildirimi ve Değerlendirmeler:** Kullanıcılar, satın aldıkları scooterları değerlendirerek diğer kullanıcılara yardımcı olabilirler. Bu sayede platform, kullanıcı deneyimlerini paylaşma ve değerlendirmeleri görüntüleme imkanı sunarak daha bilinçli alışveriş yapılmasını sağlar.

5. **Kapsamlı Kullanıcı Profili ve Öneriler:** Kullanıcılar, tercihlerine ve favori ürünlerine dayalı olarak önerilen scooterları görebilirler. Ayrıca, kişiselleştirilmiş kullanıcı profilleri üzerinden alışveriş deneyimlerini daha da zenginleştirebilirler.

3. Tasarladığınız veri tabanı mimarisinde hangi tablo ve ilişkileri kullanıldığınızı açıklayınız. (10 p)

Scooter Tablosu: Bu tablo, scooterların genel bilgilerini saklar. ScooterID ile her scooter kaydı benzersiz bir kimlik alır. MarkaID ve ModelID, diğer tablolarla bağlantı kurar. Örneğin, bu tablo üzerinden bir scooterın markası ve modeli diğer tablolar aracılığıyla detaylarıyla görülebilir.

GenelOzellik Tablosu: Scooterların genel özelliklerini içerir. Her bir scooterın en yüksek motor gücü, menzil, hız, taşıma kapasitesi, tekerlek tipi, sayısı, lastik ebatı ve katlanabilirlik gibi özellikleri bu tabloda tutulur. ScooterID ile Scooter tablosuyla ilişkilendirilir.

TeknikOzellik Tablosu: Scooterların teknik detaylarını içerir. Önemli teknik özellikler, örneğin batarya voltajı, akımı, tırmanma açısı, fren teknolojisi, koruma sertifikasyonu, renk ve ağırlık gibi detaylar bu tabloda yer alır. Yine ScooterID ile Scooter tablosuyla ilişkilendirilir.

Model ve Marka Tabloları: Scooter modellerinin ve markalarının detaylarını içerir. Her bir marka birden fazla modele sahip olabilir, bu nedenle MarkaID ile Model tablosu arasında bir ilişki bulunur. Böylece bir markaya ait tüm modeller Model tablosunda saklanır.

Diğer Özellik Tabloları: Tekerlek tipi, fren teknolojisi, koruma sertifikasyonu ve renk gibi özellikler ayrı tablolarda saklanır. Bu tablolar Scooter, GenelOzellik ve TeknikOzellik tablolarıyla ilişkilendirilir.

Fiyatlar Tablosu: Scooterların farklı satıcılardan ve farklı sitelerden fiyatlarını içerir. Her bir scooter için birden fazla fiyat kaydı bulunabilir. ScooterID ile Fiyat tablosu arasında ilişki vardır ve bu tablo Site ve Satıcı tablolarıyla ilişkilendirilir.

Urun Tablosu: Bu tablo, scooterlarla ilgili genel, teknik ve fiyat bilgilerini bir araya getirir. Bu sayede her bir scooter için tüm detaylar tek bir tabloda toplanır.

Kullanıcı, Favoriler ve FavorilerDetay Tabloları: Kullanıcıların kaydını tutar, favori ürünlerini saklar ve favori ürünlerin detaylarını içerir. KullanıcıID ile Favoriler tablosu arasında ilişki bulunur ve bu tablo FavorilerDetay tablosu ile bağlantılıdır.

Bu yapı, scooterların genel, teknik ve fiyat bilgilerini ayrı ayrı tutarak, bu bilgiler arasındaki ilişkileri sağlayarak kullanıcıların ihtiyaç duydukları bilgilere erişmelerini kolaylaştırır. Kullanıcılar, istedikleri scooter modellerini bulabilir, özelliklerini karşılaştırabilir ve fiyatlarını görüntüleyebilirler. Aynı zamanda favori ürünlerini kaydedebilir ve yönetebilirler. Bu da kullanıcıların daha bilinçli tercihler yapmalarını sağlayabilir.

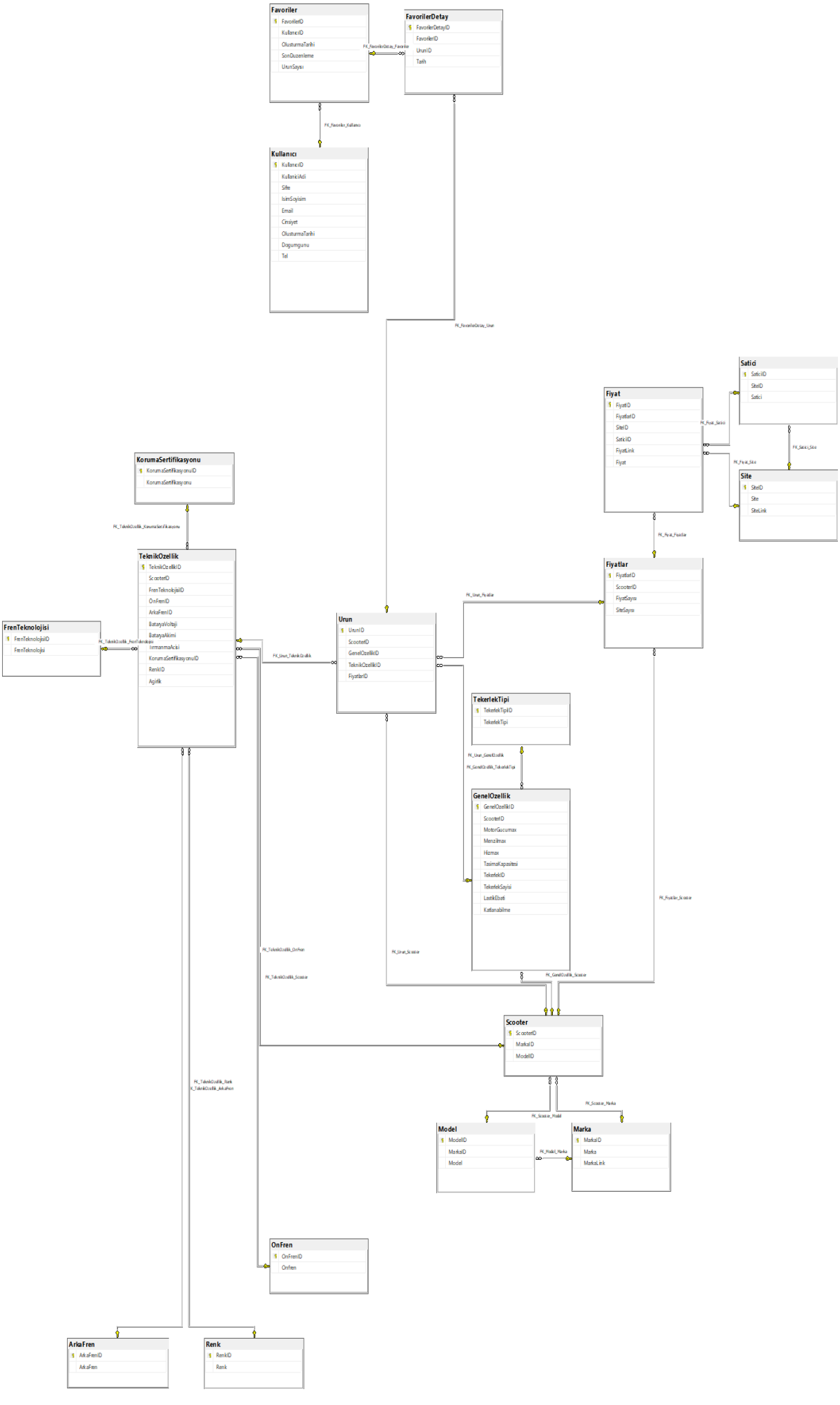
4. Veri tabanı ER (Entity Relationship) diagramının bilgisayar ortamında çizilmiş halini paylaşınız. (Ara raporda eksik kısımlar bu raporda giderilmelidir ve ER çizme programlarından faydalanıbilir. Elle çizim, çizip fotoğrafını çekme vb. kabul edilmeyecektir.) (10 p)

Adı Soyadı: Ahmed Said Kılıç

Numara:170421015

Adı Soyadı: Yunus Alp Turan

Numara:171421001



Adı Soyadı: Ahmed Said Kılıç
Numara:170421015

Adı Soyadı: Yunus Alp Turan
Numara:171421001

5. Herhangi iki tablonuz için DDL (create) kodları yazılmalıdır. (10 p)

```
CREATE TABLE GenelOzellik(
GenelOzellikID INT IDENTITY(1,1),
ScooterID INT,
MotorGucumax INT,
Menzilmax INT,
Hizmax INT,
TasimaKapasitesi  INT,
TekerlekID INT,
TekerlekSayisi INT,
LastikEbati INT,
Katlanabilme   VARCHAR(5),
CONSTRAINT [PK_GenelOzellik] PRIMARY KEY CLUSTERED (GenelOzellikID ASC))

CREATE TABLE TeknikOzellik(
TeknikOzellikID INT IDENTITY(1,1),
ScooterID INT,
FrenTeknolojisiID INT,
ÖnFrenID  INT,
ArkaFrenID INT,
BataryaVoltaji INT,
BataryaAkimi INT,
TirmanmaAcisi  INT,
KorumaSertifikasyonuID INT,
RenkID INT,
Agirlik FLOAT,
CONSTRAINT [PK_TeknikOzellik] PRIMARY KEY CLUSTERED (TeknikOzellikID ASC))
```

6. 5 adet DML (update, insert, delete) içeren kodları yazılmalıdır. (10 p)

```
# GenelOzellik tablosuna veri ekleme
        cursor.execute("""
            INSERT INTO GenelOzellik (ScooterID, MotorGucumax, Menzilmax,
Hizmax, TasimaKapasitesi, TekerlekID, TekerlekSayisi, LastikEbati,
Katlanabilme)
            VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?)
        """, (scooter_id, motor_gucu_max, menzil_max, hiz_max,
tasima_kapasitesi, tekerlek_id, tekerlek_sayisi, lastik_ebati, katlanabilme))
        conn.commit()
```

```
# TeknikOzellik tablosuna veri ekleme
        cursor.execute("""
            INSERT INTO TeknikOzellik (ScooterID, FrenTeknolojisiID, ÖnFrenID,
ArkaFrenID, BataryaVoltaji, BataryaAkimi, TirmanmaAcisi,
KorumaSertifikasyonuID, RenkID, Agirlik)
            VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?)
        """, (scooter_id, fren_teknolojisi_id, on_fren_id, arka_fren_id,
batarya_voltaji, batarya_akimi, tirmanma_acisi, koruma_sertifikasyonu_id,
renk_id, agirlik))
        conn.commit()
```

```
# İlgili FiyatlarID'leri bul
        cursor.execute("""
            SELECT FiyatlarID
            FROM Fiyatlar
            WHERE ScooterID = ?
        """, scooter_id)
        fiyatlar_ids = cursor.fetchall()

        # Her bir FiyatlarID için Fiyat tablosunu güncelle
        for fiyatlar_id in fiyatlar_ids:
            cursor.execute("""
                UPDATE Fiyat
                SET Fiyat = ?
                WHERE FiyatlarID = ?
            """, (new_price, fiyatlar_id[0]))
```

```
# İlişkili tablolardaki verileri silin
        cursor.execute("DELETE FROM Fiyatlar WHERE ScooterID = ?", scooter_id)
        cursor.execute("DELETE FROM TeknikOzellik WHERE ScooterID = ?",
scooter_id)
        cursor.execute("DELETE FROM GenelOzellik WHERE ScooterID = ?",
scooter_id)

        # Ana Scooter tablosundan kaydı silin
        cursor.execute("DELETE FROM Scooter WHERE ScooterID = ?", scooter_id)

        conn.commit()
```

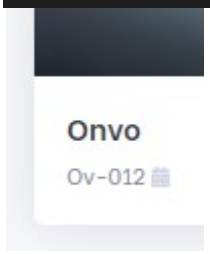
```
cursor.execute("""
            INSERT INTO Scooter (MarkaID, ModelID)
            VALUES (?, ?)
        """, (marka_id, model_id))
        conn.commit()
```

7. **Projenize ait kendi belirlediğiniz 10 adet SQL sorgusu yazınız, sorguların amacını ve sonuç çıktısını da lütfen ekleyiniz. (Açıklama: Sorgular Select deyimleri ve gruplama fonksiyonlarını HAVING deyimini (min, max, avg, count gibi) ve join deyimlerini (en az iki tablo ile birleştirme sorgusu) içerecek şekilde basitten karmaşığa doğru gitmelidir. Proje sunum anında veri tabanınıza ait sorular SQL ortamında gösterilecek ve açıklanacaktır. Raporunuzda ise sorgular, sorguların cevap ve sonuçlarının ekran görüntüsü olarak paylaşılması beklenmektedir. (20 p)**

Adı Soyadı: Ahmed Said Kılıç
Numara:170421015

Adı Soyadı: Yunus Alp Turan
Numara:171421001

```
query = """
    SELECT s.ScooterID, m.Marka, mo.Model
    FROM Scooter s
    JOIN Marka m ON s.MarkaID = m.MarkaID
    JOIN Model mo ON s.ModelID = mo.ModelID
    ORDER BY s.ScooterID
    """
```




```
query = """
    SELECT s.ScooterID,
       g.MotorGucumax, g.Menzilmax, g.Hizmax, g.TasimaKapasitesi,
tt.TekerlekTipi, g.TekerlekSayisi, g.LastikEbati, g.Katlanabilme,
       ft.FrenTeknolojisi, o.Onfren, af.ArkaFren, t.BataryaVoltaji,
t.BataryaAkimi, t.TirmanmaAcisi, kst.KorumaSertifikasyonu, r.Renk, t.Agirlik
    FROM Scooter s
    JOIN GenelOzellik g ON s.ScooterID = g.ScooterID
    JOIN TeknikOzellik t ON s.ScooterID = t.ScooterID
    JOIN TekerlekTipi tt ON g.TekerlekTipiID = tt.TekerlekTipiID
    JOIN FrenTeknolojisi ft ON t.FrenTeknolojisiID = ft.FrenTeknolojisiID
    JOIN OnFren o ON t.ÖnFrenID = o.OnfrenID
    JOIN ArkaFren af ON t.ArkaFrenID = af.ArkaFrenID
    JOIN KorumaSertifikasyonu kst ON t.KorumaSertifikasyonuID =
kst.KorumaSertifikasyonuID
    JOIN Renk r ON t.RenkID = r.RenkID
    """
```

{'Ozellikler': {'MotorGucumax': 800, 'Menzilmax': 50, 'Hizmax': 45,
'TasimaKapasitesi': 150, 'TekerlekID': 'Şişme (Pnömatik)', 'TekerlekSayisi': 2,
'LastikEbati': 10, 'Katlanabilme': 'Var', 'FrenTeknolojisiID': 'Mekanik', 'ÖnFrenID':
'Disk', 'ArkaFrenID': 'Disk', 'BataryaVoltaji': 48, 'BataryaAkimi': 10, 'TirmanmaAcisi':
30, 'KorumaSertifikasyonuID': 'IPX4 Koruma', 'RenkID': 'Siyah', 'Agirlik': 24.0}}

```
query = """
    SELECT fl.ScooterID, f.Fiyat, s.Site, f.FiyatLink
    FROM Fiyatlar fl
    JOIN Fiyat f ON fl.FiyatlarID = f.FiyatlarID
    JOIN Site s ON f.SiteID = s.SiteID
    ORDER BY fl.ScooterID
    """
```

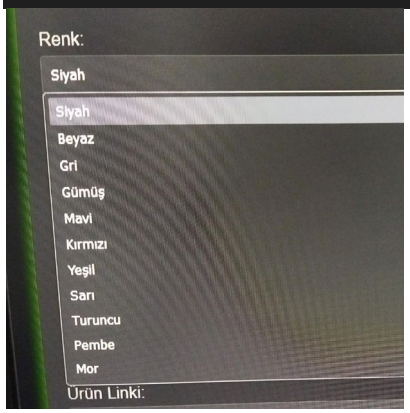
Adı Soyadı: Ahmed Said Kılıç

Numara:170421015

Adı Soyadı: Yunus Alp Turan

Numara:171421001


```
query = """
    SELECT RenkID, Renk FROM Renk
    """
```



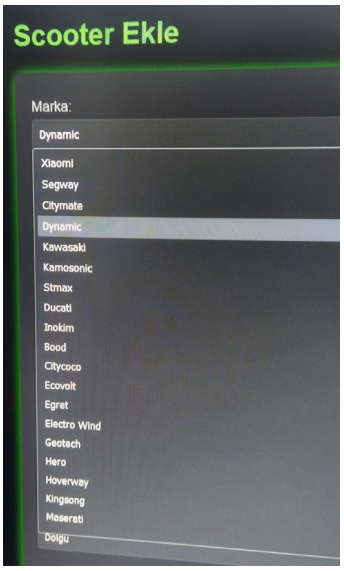

```
query = """
    SELECT MarkaID, Marka FROM Marka
    """
```

Adı Soyadı: Ahmed Said Kılıç

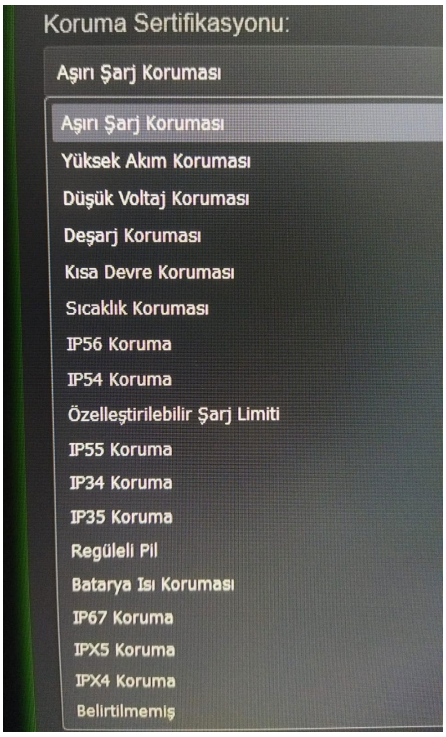
Numara:170421015

Adı Soyadı: Yunus Alp Turan

Numara:171421001




```
query = """
    SELECT KorumaSertifikasyonuID, KorumaSertifikasyonu FROM
KorumaSertifikasyonu
    """
```

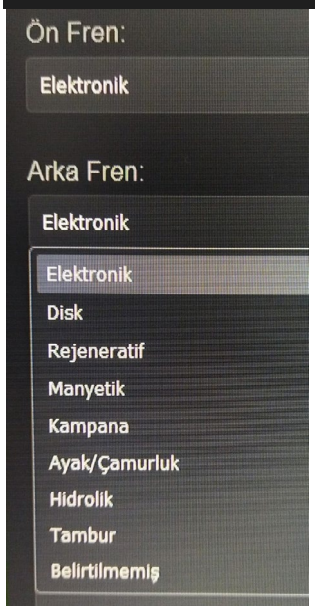



```
query = """
    SELECT TekerlekTipiID, TekerlekTipi FROM TekerlekTipi
    """
```

Adı Soyadı: Ahmed Said Kılıç
Numara:170421015

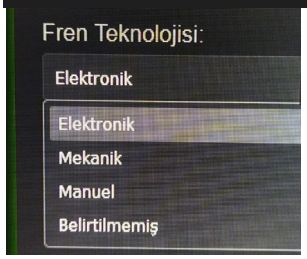
Adı Soyadı: Yunus Alp Turan
Numara:171421001

```
query = """
    SELECT OnFrenID, OnFren FROM OnFren
    """
```




```
cursor.execute("""
            SELECT MAX(Fiyat)
            FROM Fiyatlar
            INNER JOIN Fiyat ON Fiyatlar.FiyatlarID = Fiyat.FiyatlarID
            WHERE Fiyatlar.ScooterID = ?
        """, scooter_id)
```

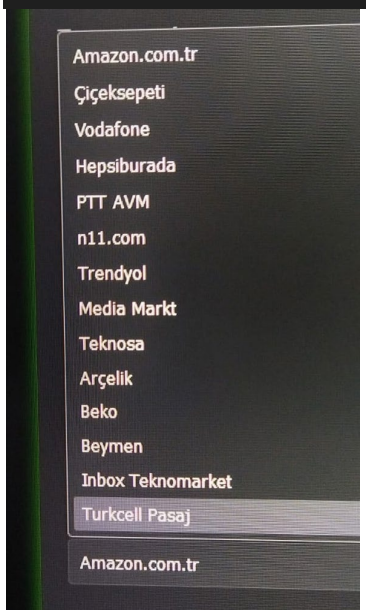

```
query = """
    SELECT FrenTeknolojisiID, FrenTeknolojisi FROM FrenTeknolojisi
    """
```



Adı Soyadı: Ahmed Said Kılıç
Numara:170421015

Adı Soyadı: Yunus Alp Turan
Numara:171421001

```
query = """
    SELECT SiteID, Site FROM Site
    """
```



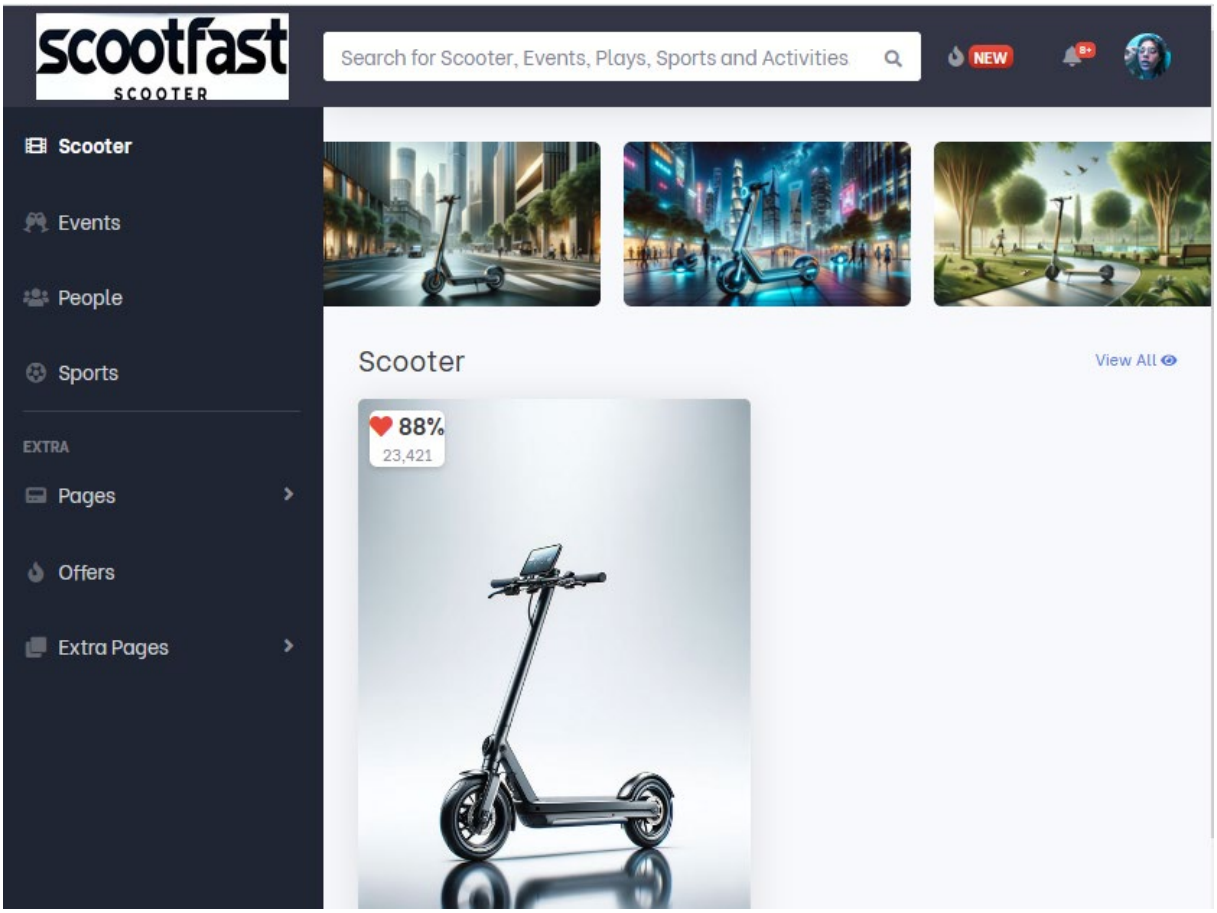
8. Eğer gerçekleştirmiş iseniz, veri tabanı bağlama ve uygulama geliştirme aşamalarınızı kısaca açıklayarak, kullanıcı ara yüz ekranından bir örnek veriniz. Ve geliştirdiğiniz ara yüzü anlatınız. (10 p)

Veri tabanını MSSQL'de oluşturduk. Veri tabanımız Scooter verilerini içeriyor. Ardından projemizin backend kısmını Django kullanarak geliştirmeye karar verdik. Django, web uygulamaları oluşturmak için kullanılan bir Python çerçevesidir ve güçlü veri tabanı entegrasyonuyla bilinir. Veri tabanı bağlantısı için Python ile MSSQL veritabanına bağlanarak SQL sorgularını doğrudan Python kodu içinde çalıştırdık.

Frontend tarafında HTML ve CSS kullanarak kullanıcı arayüzünü oluşturduk. Ana sayfa, kullanıcıların scooterların temel bilgilerini görmelerini sağlar, fiyatlarını ve özelliklerini inceleyebilirler. Detaylı sayfada ise scooterların teknik özellikleri, görselleri ve hatta videoları gibi ayrıntılı bilgiler sunuldu. Kullanıcılar ayrıca yorumlar ve incelemeleri görüntüleyebilir ve kendi yorumlarını ekleyebilirler.

Projemizin önemli bir parçası olan admin paneli, yönetici kullanıcıları için özel olarak oluşturuldu. Bu panel üzerinden yeni scooter ekleyebilir, mevcut scooterları silebilir ve fiyat güncellemeleri yapabiliriz. Böylece verilerin kolayca yönetilebilmesini sağladık.

Bu şekilde projemizi geliştirerek, kullanıcıların scooterlar hakkında bilgi edinmelerini ve yöneticilerin veritabanı işlemlerini kolayca gerçekleştirmelerini sağladık.



9. Eğer veri tabanı bağlama işlemini gerçekleştirmemiş iseniz VTYS sistemlerinde Transaction nedir açıklayınız ve çalışmanızdan bir Transaction örneği veriniz. (10 p)

10. View nedir açıklayınız ve bir adet view, bir adet saklı yordam (Stored Procedute) ifadesine ait SQL deyimlerinin sorgusunu ve cevabını yazınız. (10 p)

Bir "view" , veri tabanında saklanmayan, ancak bir veya daha fazla tablodan verileri sorguları daha basit hale getirmek ve belirli bir amaç için daha anlamlı bir şekilde sunmak için kullanılan sanal bir tablodur. View'lar, karmaşık sorguları daha okunabilir ve yönetilebilir hale getirmenin yanı sıra veritabanı güvenliğini artırmak için de kullanılabilir.

Adı Soyadı: Ahmed Said Kılıç
Numara:170421015
Adı Soyadı: Yunus Alp Turan
Numara:171421001

Bir adet view ve bir adet saklı yordam örneği :

1. **View Örneği:** Diyelim ki bir e-ticaret veritabanınız var ve "Ürünler" ve "Kategoriler" adlı iki tablonuz bulunuyor. Kullanıcılar, her kategoriye ait ürünleri görmek istiyor. Bu durumda bir view oluşturabiliriz:

-
```
CREATE VIEW KategoriUrunleri AS
SELECT Kategori.Ad AS KategoriAd, Urun.UrunAd AS UrunAd
FROM Kategoriler AS Kategori
JOIN Urunler AS Urun ON Kategori.KategoriID = Urun.KategoriID;
```

Bu view, "KategoriUrunleri" adı altında, her kategoriye ait ürünleri içeren bir sanal tablo oluşturur. Kullanıcılar, bu view'i sorgulayarak kategoriye göre ürünleri daha kolay bir şekilde elde edebilirler.

- **Saklı Yordam (Stored Procedure) Örneği:** Şimdi de bir saklı yordam (stored procedure) örneği verelim. Diyelim ki müşteri siparişlerini veritabanınıza eklemek için bir saklı yordam oluşturmak istiyorsunuz:

```
sql
CREATE PROCEDURE SiparisEkle
@MusteriID INT,
@UrunID INT,
@Adet INT
AS
BEGIN
   INSERT INTO Siparisler (MusteriID, UrunID, Adet, SiparisTarihi)
   VALUES (@MusteriID, @UrunID, @Adet, GETDATE());
END;
```

Bu saklı yordam, "SiparisEkle" adıyla oluşturuldu. Müşteri ID, Ürün ID ve Adet parametreleri alır ve bu bilgileri "Siparisler" tablosuna ekler. Ayrıca siparişin verildiği tarih de otomatik olarak kaydedilir.