

**Gazi Üniversitesi**  
**BM102 Dönem Sonu Projesi**

**Devamsızlık Otomasyon Sistemi**

Ahmet Emir Coşkun - 22118080049

Baran Asar - 22118080072

İshak Yorgancı - 23118080033

Mustafa Temur Turan - 22118080021

Necmettin Sönmez - 22118080037

Yusuf Duman – 22118080056

# 1) Giriş

**Tanım:** Devamsızlık Otomasyon Sistemi Projesi; derse katılan öğrencilerin yüzlerini taratarak anlık olarak OBS sisteminde işlenmesini ve işlenen verilerin kullanıcılar tarafından takip edilebilmesini sağlamaktadır.

**Amaç:** Projede; yoklama alınırken ve işlenirken gerçekleşen vakit kaybını önlemek, yoklamanın güvenilirliğini arttırmak, öğretmen ve öğrencilerin devamsızlık durumlarını güncel olarak kontrol edebilmesini sağlamak amaçlanmıştır.

Sistemimiz sade ve anlaşılır bir arayüz ile tasarlanmış olup çeşitli eğitim kurumlarında rahatça kullanılabilecek şekilde tasarlanmıştır, bu sayede yoklamalar dijitalleştirilip kontrol edilebilirlik ve güvenliğin en üst seviyede sağlanması hedeflenmiştir.

## **Kapsam:**

- 1) Giriş Ekranı: İlgili kullanıcı yetkisinin seçilip *Kullanıcı ID* ve *Şifre* ile OBS sistemine giriş yapılan kısım.
- 2) Yönetici Ekranı: *Öğretmen*, *Öğrenci* ve *Ders Programı* bilgilerinin görüntülediği ve düzenlendiği kısım.
- 3) Öğretmen Ekranı: Giriş yapan öğretmenin kendi dersindeki devamsızlık durumlarını görüntülediği kısım.
- 4) Öğrenci Ekranı: Giriş yapan öğrencinin derslerdeki devamsızlık durumlarını görüntülediği kısım.
- 5) Yüz Tanıma Sistemi: Sisteme öğrencilerin yüzlerinin kaydedildiği ve yüzlerin taratılarak derse katılım durumunu veri tabanına gönderen kısım.
- 6) Veri Tabanı: *Yüz Tanıma Sistemi* ile *OBS* sistemi arasında köprü görevi gören ve verilerin tutulup işlendiği kısım.

## 2) Proje Ekibi ve Görev Dağılımı

### • Ahmet Emir Coşkun

- SQL Tablolarının tasarlanması ve oluşturulması
- Tablolar arasındaki bağlantıların sağlanması ve *Diyagram* yapısının oluşturulması
- *Yoklama* tablosundaki verilerin kontrolünün sağlanıp anlamlı şekilde işlenmesi için *DevamsizlikGuncelle* ve *ToplamDersGuncelle* Triggerlarının oluşturulması
- Oluşturulan Triggerlar yardımıyla verilerin *Devamsızlık* ve *İşlenen\_Toplam\_Ders* tablolarına yazılması

### • Mustafa Temur Turan

- *Yüz Tanıma Sistemi*'nin oluşturulması
- Sisteme kullanıcıların tanıtılması
- Kamera ile kullanıcıların doğru biçimde taratılması
- Taratılan kullanıcıların öğrenci numaralarının ve taratılan tarihin veri tabanındaki *Yoklama* tablosuna yazılması

### • Baran Asar

- OBS sisteminin *Login* panelinin oluşturulması ve veri tabanı bağlantısının yapılması
- *Login* panelindeki giriş ve şifre bilgilerinin doğrulanması
- *Login* panelindeki giriş bilgileri doğrultusunda ilgili ekrana yönlendirilmesi

### • İshak Yorgancı

- *Azure SQL Server* sisteminin kurulması, kullanıcı ve erişim sisteminin yönetilmesi
- SQL üzerindeki temel *Diyagram* yapısının oluşturulması
- OBS sistemindeki *Ogrenci* ve *Ogretmen* panellerinin oluşturulması ve veri tabanı bağlantısının yapılması
- SQL ve OBS sisteminde oluşan hataların çözümünde ve algoritmaların tasarlanmasında rol oynama

• **Yusuf Duman**

- OBS sistemindeki *Yonetici* panelin oluşturulması ve veri tabanı bağlantısının yapılması
- *Yonetici* panelindeki ekle, sil ve düzenle gibi butonların oluşturulup işlevselliğinin sağlanması
- *Yonetici* panelindeki veri işlemlerinin ve hata kontrolünün sağlanması
- OBS sistemindeki algoritmaların geliştirilmesi ve veri tabanı bağlantılarının sağlanmasında rol oynama

• **Necmettin Sönmez**

- OBS sistemindeki panellerin tasarımlarının yapılması ve gerekli kaynakların kullanılması
- OBS sistemindeki paneller arasındaki bağlantıların sağlanması
- OBS sistemindeki algoritmaların geliştirilmesi ve hataların çözümünde rol oynama

### 3) Kullanılan Teknoloji ve Araçlar

#### •Programlama Dilleri

- **C#:** Windows Forms ile OBS sisteminin oluşturulması
- **SQL:** Veri Tabanının oluşturulması
- **Python:** Yüz Tanıma Sistemi'nin OpenCV kütüphanesi kullanılarak oluşturulması

#### •IDE'ler

- **Visual Studio:** Windows Forms - OBS sisteminin kodların yazılıp derlenmesi ve arayüz oluşturulması
- **PyCharm:** Yüz Tanıma Sistemi'ndeki kodların yazılıp derlenmesi ve sistemin çalıştırılması
- **SQL Server Management Studio:** Veri Tabanının oluşturulması ve SQL Server işlemlerinin yapılması

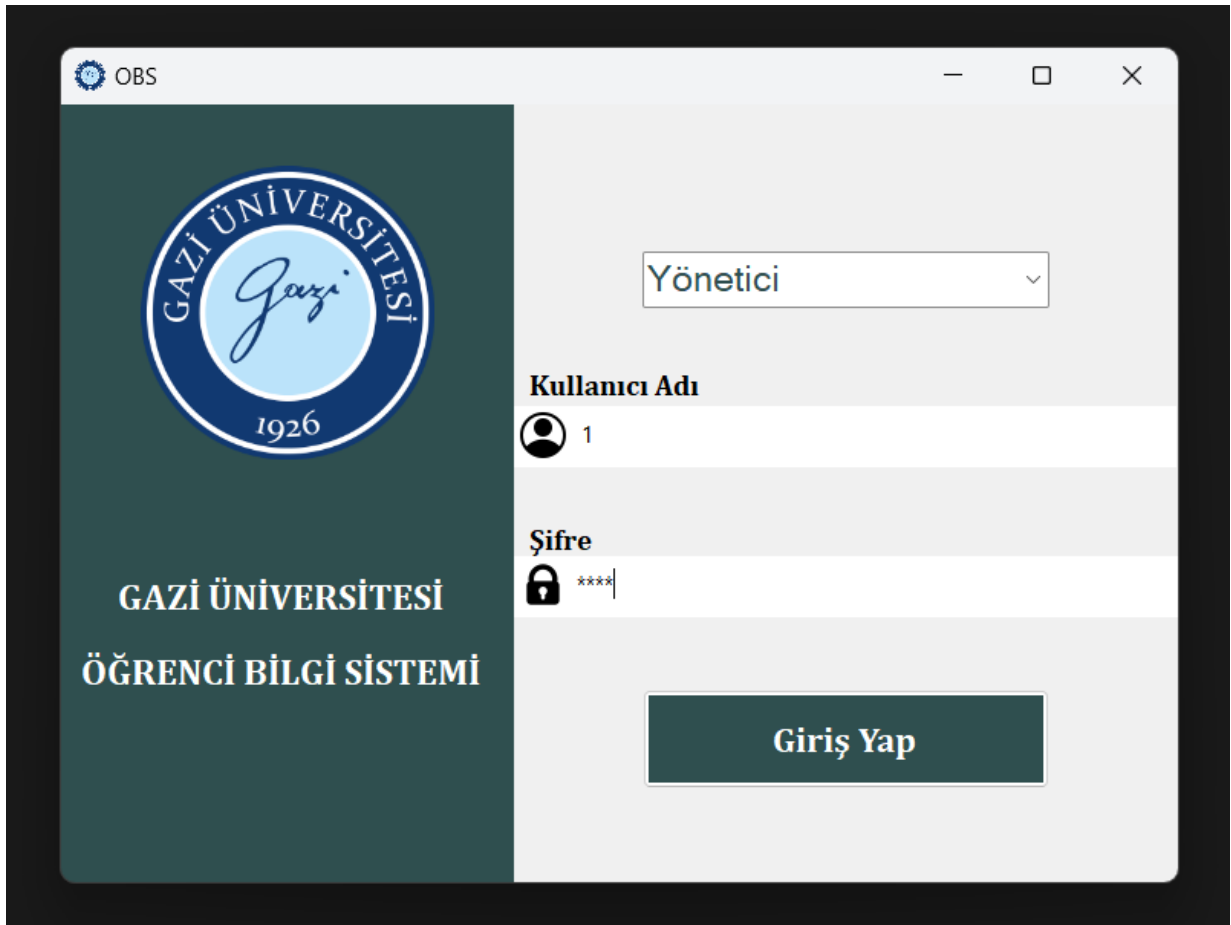
#### •Teknolojiler ve Araçlar

- **Microsoft Azure Data Studio:** Veri tabanının erişiminin ve kontrollerinin ayarlanması
- **Microsoft Azure SQL Server:** Herkesin erişebileceği şekilde online bir SQL veri tabanının oluşturulması
- **Git & GitHub:** Aynı proje üzerinde çalışmanın sağlanması ve proje düzenlemelerinin yapılması

## 4) Gerçekleştirilen Çalışma

### •OBS Sisteminin Arayüzü ve Paneller

#### Giriş Ekranı



OBS

GAZİ ÜNİVERSİTESİ  
1926

GAZİ ÜNİVERSİTESİ  
ÖĞRENCİ BİLGİ SİSTEMİ

Yönetici

Kullanıcı Adı


1

Şifre

\*\*\*\*

Giriş Yap

## Yönetim Ekranı



Öğrenci Yönetimi

Öğretmen Yönetimi

Ders Yönetimi

Yönetim

Öğrencileri Yönet

Ad:

Mustafa Temur

Soyad:

Turan

Şifre:

1

Öğrenci No:

221821

Ekle


Düzenle

Yoklama

Sil

Veriler:

Öğrenci No	Öğrenci Adı	Öğrenci Soyadı	Öğrenci Şifresi
221821	Mustafa Temur	Turan	1
221837	Necmettin	Sönmez	1
221849	Ahmet Emir	Coşkun	1
221856	Yusuf	Duman	1
221872	Baran	Asar	1



Öğrenci Yönetimi

Öğretmen Yönetimi

Ders Yönetimi

Yönetim

Öğretmenleri Yönet

Ad:

Soyad:

Öğretmen Ve Ders No:

Ders Adı:

Şifre:


Ekle

Düzenle

Sil

Veriler:

Öğretmen No	Öğretmen Adı	Öğretmen Soyadı	Ders Adı	Öğretmen Şifresi
1	Oktay	Yıldız	BM102	1
2	Aybike	Şimşek	BM102 LAB	1
*				



Oğrenci Yönetimi

Oğretmen Yönetimi

Ders Yönetimi

## Yönetim

Öğretmen ID:

1

Ders Günü:

Monday

Başlangıç Saati:

13:30:00

Bitiş Saati:

16:20:00


Ekle

Sil

### Ders Düzenleme

	Öğretmen No	Ders Günü	Başlangıç Saati	Bitiş Saati
▶	1	Monday	13:30:00	16:20:00
	2	Monday	10:30:00	12:20:00
	1	Tuesday	10:30:00	12:20:00
	1	Wednesday	16:30:00	20:00:00
*				

### Öğrenci Ekranı



## Öğrenci

Öğrenci Adı:

Ishak Bedir

Öğrenci Soyadı:

Yorgancı

Öğrenci Kimlik:


231833

### Veriler:

	Ders Adı	Öğretmen Adı	Öğretmen Soyadı	Mevcut	Toplam
▶	BM102 LAB	Aybike	Şimşek	1	1
	BM102	Oktay	Yıldız	2	3
*					



## Öğretmen Ekranı



GAZİ ÜNİVERSİTESİ  
1925

Öğretmen

Öğretmen Adı:

Öğretmen Soyadı:








Öğretmen ID:

Veriler:

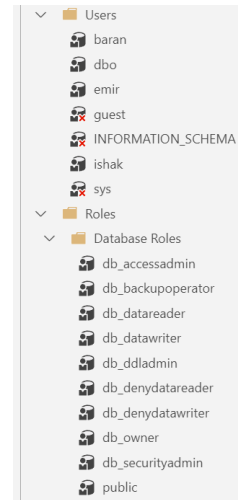
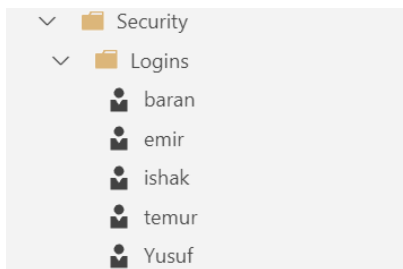
Öğrenci No	Adı	Soyadı	Var	Toplam
231833	Ishak Bedir	Yorgancı	2	3
221837	Necmettin	Sönmez	1	3
221872	Baran	Asar	2	3
221821	Mustafa Temur	Turan	2	3
221849	Ahmet Emir	Coşkun	1	3

## •Azure SQL Server

Azure SQL Server'a erişimin sağlanması kullanıcıların oluşturulması; serverda oluşacak herhangi bir sorunun çözülmesi, yedeklemelerin yapılması ve kontrolü server sağlığının kontrolü burada sağlanır.

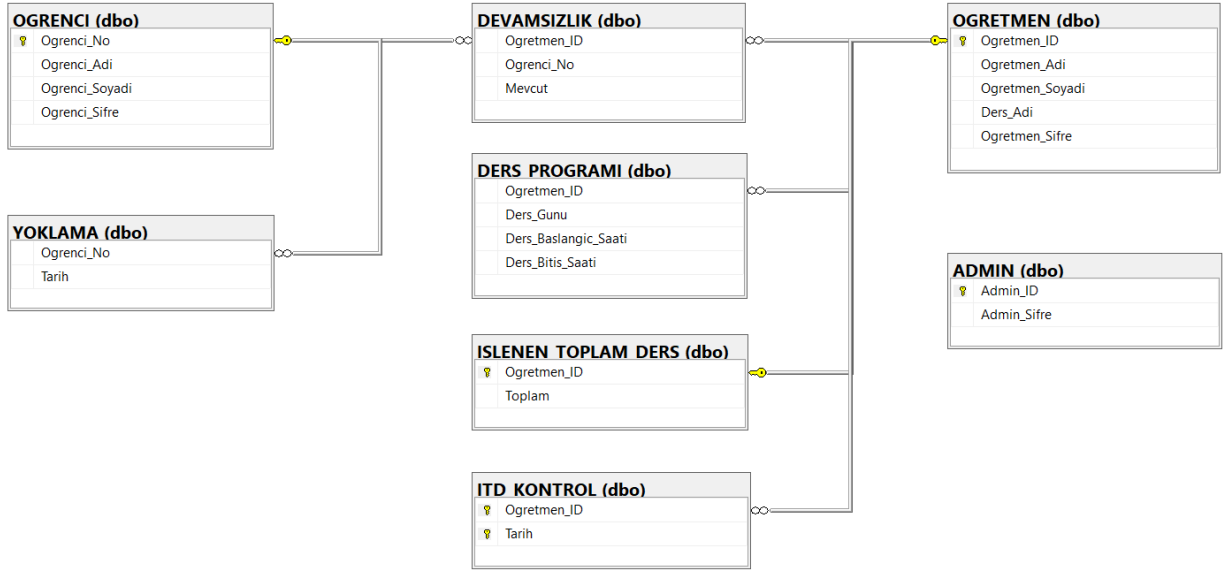
<input type="checkbox"/>	 aecoskuno (Konuk) aecoskuno@outlook.com	Kullanıcı	Sahip	Bu kaynak	Ekle
<input type="checkbox"/>	 ishakBedir Yorganci ishakbedir06_hotmail.com#EXT#...	Kullanıcı	Sahip	Abonelik (Devralınmış)	Yok
<input type="checkbox"/>	 ishakBedir Yorganci ishakbedir06_hotmail.com#EXT#...	Kullanıcı	Sahip	Abonelik (Devralınmış)	Yok
<input type="checkbox"/>	 ishakBedir Yorganci ishakbedir06_hotmail.com#EXT#...	Kullanıcı	Sahip	Bu kaynak	Ekle
<input type="checkbox"/>	 ishakbediryorganci (Konuk) ishakbediryorganci@gmail.com	Kullanıcı	Sahip	Bu kaynak	Ekle
<input type="checkbox"/>	 yusuf78duman (Konuk) yusuf78duman@gmail.com	Kullanıcı	Sahip	Bu kaynak	Ekle
<input type="checkbox"/>	 yusuf78duman (Konuk) yusuf78duman@gmail.com	Kullanıcı	Sahip	Kaynak grubu (Devralınmış)	Yok

baran	5.24.215.162	5.24.215.162	
ClientIPAddress_2024-6-1_21-38-45	95.65.212.176	95.65.212.176	
ClientIPAddress_2024-6-1_21-40-9	217.131.115.103	217.131.115.103	
ClientIPAddress_2024-6-1_22-8-44	217.131.115.103	217.131.115.103	
ClientIPAddress_2024-6-3_12-58-36	194.27.19.111	194.27.19.111	
ClientIPAddress_2024-6-7_17-22-49	95.65.218.30	95.65.218.30	
ClientIPAddress_2024-6-8_14-37-33	217.131.99.191	217.131.99.191	
emir2	31.233.50.64	31.233.50.64	
emir3	31.223.50.0	31.223.50.255	
emirr	31.223.43.214	31.223.43.214	
okulbaran	194.27.19.0	194.27.19.255	
temur	78.175.54.44	78.175.54.44	
temurmobil	176.216.210.0	176.216.210.255	



## •SQL Server

### Database Diyagramı



### **DevamsizlikGuncelle Tetikleyicisi**

DevamsizlikGuncelle tetikleyicisi, YOKLAMA tablosuna yeni bir kayıt eklendiğinde çalışır. Bu tetikleyici, eklenen kayıtların öğrenci numarası ve tarih bilgilerini alır, bu bilgilere göre hangi gün ve saat aralığında ders işlendiğini belirler. Belirlenen zaman diliminde ders veren öğretmenin ID'sini DERS\_PROGRAMI tablosundan bulur ve bu öğretmen için öğrencinin devamsızlık bilgisini günceller. Burada eğer ilgili öğretmen ve öğrenci kombinasyonu daha önce DEVAMSIZLIK tablosunda mevcutsa, mevcut devamsızlık sayısını bir artırır; eğer bu kombinasyon mevcut değilse, yeni bir kayıt ekler. Bu şekilde, her yeni yoklama kaydı sonrası ilgili öğretmen için devamsızlık kayıtları güncellenmiş olur, böylece öğrencilerin devamsızlık bilgileri düzenli olarak takip edilir.

### **ToplamDersGuncelle Tetikleyicisi**

ToplamDersGuncelle tetikleyicisi, YOKLAMA tablosuna yeni bir kayıt eklendiğinde çalışır. Bu tetikleyici, eklenen kayıtların öğrenci numarası ve tarih bilgilerini alarak hangi gün ve saat aralığında ders işlendiğini belirler. Bu bilgilere dayanarak dersin öğretmen ID'sini DERS\_PROGRAMI tablosundan bulur ve öğretmenin toplam işlenen ders sayısını ISLENEN\_TOPLAM\_DERS tablosunda günceller. Ayrıca, ITD\_KONTROL tablosuna öğretmen ve tarih kombinasyonunu ekleyerek, aynı gün içinde aynı öğretmen için birden fazla kez toplam ders güncellemesi yapılmasını engeller. Eğer öğretmenin ders verdiği tarih daha önce ITD\_KONTROL tablosuna eklenmemişse tabloya ekler, eklenmişse bir işlem yapmaz. Böylece, öğretmenlerin işledikleri toplam ders sayıları düzenli olarak güncellenmiş ve takip edilmiş olur. ISLENEN\_TOPLAM\_DERS tablosundaki veriler de OBS kısmında Join yapısı ile kullanılarak Devamsızlık tablosuna sütun olarak eklenir.

```

/***** Object: Trigger [dbo].[TRG_DevamsizlikGuncelle] *****/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

-- [dbo].[YOKLAMA] tablosuna INSERT işlemi yapıldığında tetiklenecek olan trigger-tetikleyici oluşturuldu
CREATE TRIGGER [dbo].[TRG_DevamsizlikGuncelle] ON [dbo].[YOKLAMA]
AFTER INSERT
AS
BEGIN
    -- Yerel değişkenler tanımlandı
    DECLARE @Ogrenci_No BIGINT, @Tarih smalldatetime, @Gun VARCHAR(10), @Saat TIME, @Ogretmen_ID BIGINT

    -- INSERT edilen satırları almak için bir cursor tanımlandı
    DECLARE SatirKontrol CURSOR FOR
    SELECT Ogrenci_No, Tarih
    FROM INSERTED

    -- Cursor açıldı ve ilk kayıt alındı
    OPEN SatirKontrol
    FETCH NEXT FROM SatirKontrol INTO @Ogrenci_No, @Tarih

    -- Cursor üzerindeki kayıtlar boyunca döngü başlatıldı
    WHILE @@FETCH_STATUS = 0
    BEGIN
        -- Tarihten gün adı ve saat bilgisi alındı
        SET @Gun = DATENAME(WEEKDAY, @Tarih)
        SET @Saat = CAST(@Tarih AS TIME)

        -- DERS_PROGRAMI tablosundan dersin öğretmen ID'si alındı
        SELECT @Ogretmen_ID = Ogretmen_ID
        FROM DERS_PROGRAMI
        WHERE Ders_Gunu = @Gun
        AND @Saat BETWEEN Ders_Baslangic_Saati AND Ders_Bitis_Saati

        -- Eğer öğretmen ID'si varsa yani ders varsa
        IF @Ogretmen_ID IS NOT NULL
        BEGIN
            -- DEVAMSIZLIK tablosunda ilgili öğretmen ve öğrenci kaydı güncellendi
            IF EXISTS (
                SELECT 1
                FROM DEVAMSIZLIK
                WHERE Ogretmen_ID = @Ogretmen_ID
                AND Ogrenci_No = @Ogrenci_No
            )
            BEGIN
                UPDATE DEVAMSIZLIK
                SET Mevcut = Mevcut + 1
                WHERE Ogretmen_ID = @Ogretmen_ID
                AND Ogrenci_No = @Ogrenci_No
            END
            ELSE
            BEGIN
                -- Eğer ilgili öğretmen ve öğrenci kaydı yoksa yeni kayıt eklendi
                INSERT INTO DEVAMSIZLIK (Ogretmen_ID, Ogrenci_No, Mevcut)
                VALUES (@Ogretmen_ID, @Ogrenci_No, 1)
            END
        END

        -- Cursor'dan bir sonraki kayıt alındı
        FETCH NEXT FROM SatirKontrol INTO @Ogrenci_No, @Tarih
    END

    -- Cursor kapatıldı
    CLOSE SatirKontrol
    DEALLOCATE SatirKontrol
END;
GO

-- [dbo].[YOKLAMA] tablosu için tetikleyici etkinleştirildi.
ALTER TABLE [dbo].[YOKLAMA] ENABLE TRIGGER [TRG_DevamsizlikGuncelle]
GO

```

```

/***** Object: Trigger [dbo].[TRG_ToplamDersGuncelle] 1 *****/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

-- [dbo].[YOKLAMA] tablosuna INSERT işlemi yapıldığında tetiklenecek olan tetikleyici oluşturuldu
CREATE TRIGGER [dbo].[TRG_ToplamDersGuncelle] ON [dbo].[YOKLAMA]
AFTER INSERT
AS
BEGIN
    -- Yerel değişkenler tanımlandı
    DECLARE @Ogrenci_No BIGINT, @Tarih smalldatetime, @Gun VARCHAR(10), @Saat TIME, @Ogretmen_ID BIGINT,
    @Ders_Baslangic_Saati TIME, @Ders_Bitis_Saati TIME, @Ders_Tarih DATE

    -- INSERT edilen satırları almak için bir cursor tanımlandı
    DECLARE SatirKontrol CURSOR FOR
    SELECT Ogrenci_No, Tarih
    FROM INSERTED

    -- Cursor açılıp ilk kayıt alındı
    OPEN SatirKontrol
    FETCH NEXT FROM SatirKontrol INTO @Ogrenci_No, @Tarih

    -- Cursor üzerindeki kayıtlar boyunca döngü başlatıldı
    WHILE @@FETCH_STATUS = 0
    BEGIN
        -- Tarihten gün adı, saat ve tarih bilgisi alındı
        SET @Gun = DATENAME(WEEKDAY, @Tarih)
        SET @Saat = CAST(@Tarih AS TIME)
        SET @Ders_Tarih = CAST(@Tarih AS DATE)

        -- DERS_PROGRAMI tablosundan dersin öğretmen ID'si, başlangıç ve bitiş saatleri alındı
        SELECT @Ogretmen_ID = Ogretmen_ID, @Ders_Baslangic_Saati = Ders_Baslangic_Saati, @Ders_Bitis_Saati = Ders_Bitis_Saati
        FROM DERS_PROGRAMI
        WHERE Ders_Gunu = @Gun
        AND @Saat BETWEEN Ders_Baslangic_Saati AND Ders_Bitis_Saati

        -- Eğer öğretmen ID'si bulunursa yani ders varsa
        IF @Ogretmen_ID IS NOT NULL
        BEGIN
            -- ITD_KONTROL tablosunda öğretmen ve tarih kombinasyonu yoksa eklenir
            IF NOT EXISTS (
                SELECT 1
                FROM ITD_KONTROL
                WHERE Ogretmen_ID = @Ogretmen_ID
                AND Tarih = @Ders_Tarih
            )
            BEGIN
                -- ISLENEN_TOPLAM_DERS tablosunda öğretmen için bir kayıt varsa guncellenir
                IF EXISTS (
                    SELECT 1
                    FROM ISLENEN_TOPLAM_DERS
                    WHERE Ogretmen_ID = @Ogretmen_ID
                )
                BEGIN
                    UPDATE ISLENEN_TOPLAM_DERS
                    SET Toplam = Toplam + 1
                    WHERE Ogretmen_ID = @Ogretmen_ID
                END
            ELSE
            BEGIN
                -- Eğer öğretmen için bir kayıt yoksa yeni kayıt eklenir
                INSERT INTO ISLENEN_TOPLAM_DERS (Ogretmen_ID, Toplam)
                VALUES (@Ogretmen_ID, 1)
            END

            -- ITD_KONTROL tablosuna yeni bir kontrol kaydı eklenir
            INSERT INTO ITD_KONTROL (Ogretmen_ID, Tarih)
            VALUES (@Ogretmen_ID, @Ders_Tarih)
        END
    END

    -- Cursor'dan bir sonraki kayıt alındı
    FETCH NEXT FROM SatirKontrol INTO @Ogrenci_No, @Tarih
END

-- Cursor kapatıldı.
CLOSE SatirKontrol
DEALLOCATE SatirKontrol
END;
GO

-- [dbo].[YOKLAMA] tablosu için tetikleyici etkinleştirildi
ALTER TABLE [dbo].[YOKLAMA] ENABLE TRIGGER [TRG_ToplamDersGuncelle]
GO

```

## •Yüz Tanıma Sistemi

```
import cv2
import dlib
import face_recognition
from datetime import datetime, timedelta
import time
import pypyodbc
```

Öncelikle program için gerekli kütüphaneler yüklenir. Sırasıyla:

- Cv2 görüntü üzerinde işlem yapmak için
- Dlib ve face\_recognition yüz tespiti için
- Datetime ve time zaman verisini almak için
- Pypyodbc sql veri tabanına bağlanmak için

```
# Veritabanı bağlantı bilgileri
connection_string = (
    'Driver={ODBC Driver 17 for SQL Server};'
    'Server=yuztanima.database.windows.net;'
    'Database=yuztanima;'
    'UID=emir;'
    'PWD=P@ssw0rd!;'
)
```

Bu aşamada zamanı gelince verilerin gönderilecek olduğu veri tabanına bağlantı için gerekli bilgiler girilir.

```
# Veritabanına bağlanma
try:
    connection = pypyodbc.connect(connection_string)
    print("Veritabanı bağlantısı başarılı!")
except pypyodbc.DatabaseError as e:
    print("Veritabanı bağlantısı başarısız:", e)
    exit(1)
```

Bu aşamada bilgileri verilmiş veri tabanına bağlantı denenir. Bağlanılamazsa hata mesajı yazdırılır.

```
# Cursor oluşturma
cursor = connection.cursor()
```

Veri tabanı ile işlemler yapmak için cursor nesnesi oluşturulur.

```
# SQL sorgusu (tabloya veri ekleme)
insert_query = """
INSERT INTO YOGLAMA (Ogrenci_No, Tarih)
VALUES (?, ?)
"""
```

Bu aşamada Yöklama adlı tabloya öğrenci no ve tarih olmak üzere iki adet veri göndereceğimiz tanımlarız.



```
def ekle(numara, son_giris_zamani, simdiki_zaman):
    # Son giriş zamanını ve şimdiki zamanı datetime nesnesine çevirme
    zaman_simdiki = datetime.strptime(simdiki_zaman, __format: "%H:%M")
    zaman_onceki = datetime.strptime(son_giris_zamani, __format: "%H:%M")

    # Son girişten geçen saniye farkının hesaplanması
    fark = zaman_simdiki - zaman_onceki
    fark_saniye = fark.total_seconds()

    if fark_saniye > (40 * 60): # 40 dakikadan fazla geçmişse
        try:
            cursor.execute(insert_query, (numara, datetime.now().strftime("%Y-%m-%d %H:%M:%S")))
            connection.commit()
            print("Veri başarıyla eklendi!")
        except pypyodbc.DatabaseError as e:
            print("Veri eklenirken hata oluştu:", e)
            print(f"{numara} kaydedildi")
            return simdiki_zaman
        else:
            return son_giris_zamani
```

Bu aşamada daha sonraları sql tablosuna veri yüklemek için kullanılacak bir fonksiyon oluşturulur.

Fonksiyon numara, son\_giris\_zamani, simdiki\_zaman olmak üzere 3 parametre alır son giriş tarihi ile anlık zamanı kıyaslar. Tabloda veri karmaşası olmaması ve sadece ders başına bir veri gönderilmesi için aradaki farkın 40 dakikadan fazla olduğu zaman veri tabanına öğrencinin numarasını ve anlık zaman bilgisini ( yıl ay gün saat : dakika : saniye formatında ) gönderir. Daha sonra bu veri işlenerek obs sistemine eklenir.

```
# Kişilerin yüz tanımları
temur = face_recognition.load_image_file("temur.jpg")
temur_enc = face_recognition.face_encodings(temur)[0]
temur_son_giris = "00:00"
```

İlgili dizinden kişilerin ön yüzünün çekildiği fotoğrafı ilgili değişkene yükler ve daha sonra bu fotoğrafı çözümleyerek yüzün ayırt edici özelliklerini çıkarır. Ardından daha sonra işlem yapılması için kişiye bir son giriş değişkeni oluşturulur.

Bu işlem her bir kişi için gerçekleştirilir.

```
# Yüz dedektörü
detector = dlib.get_frontal_face_detector()

# Kameradan görüntü alma
cap = cv2.VideoCapture(0)
```

Bir yüz dedektörü oluşturulur ve webcam açılır.

```
try:
    while True:
        time.sleep(0.3)
        ret, frame = cap.read()
        face_loc = []
        faces = detector(frame)

        for face in faces:
            x = face.left()
            y = face.top()
            w = face.right()
            h = face.bottom()
            face_loc.append((y, w, h, x))

        face_enc = face_recognition.face_encodings(frame, face_loc)

        for i, face in enumerate(face_enc):
            y, w, h, x = face_loc[i]

            temur_s = face_recognition.compare_faces(known_face_encodings: [temur_enc], face)
            kurt_s = face_recognition.compare_faces(known_face_encodings: [kurt_enc], face)
            baran_s = face_recognition.compare_faces(known_face_encodings: [baran_enc], face)
            emir_s = face_recognition.compare_faces(known_face_encodings: [emir_enc], face)
            ishak_s = face_recognition.compare_faces(known_face_encodings: [ishak_enc], face)
            yusuf_s = face_recognition.compare_faces(known_face_encodings: [yusuf_enc], face)
```

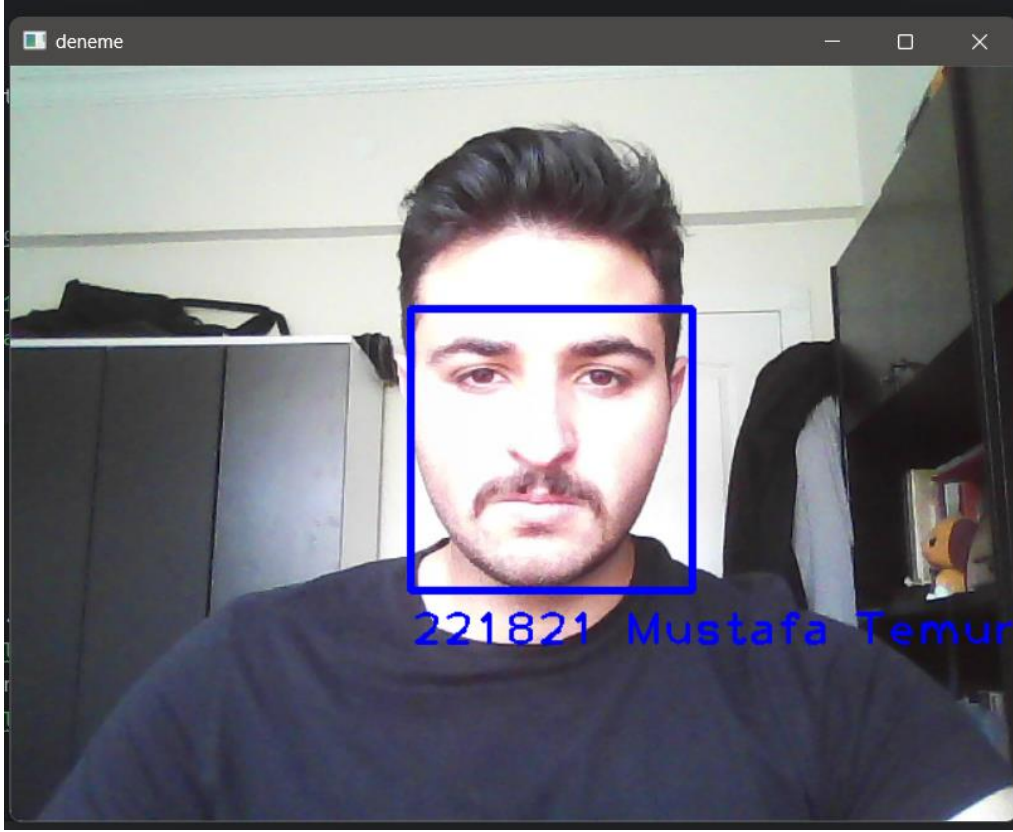
```

current_time = datetime.now().strftime('%H:%M')
if temur_s[0]:
    cv2.rectangle(frame, (x, y), (w, h), (255, 0, 0), 3)
    cv2.putText(frame, text: "221821 Mustafa Temur TUARAN", org: (x, h + 35), cv2.FONT_HERSHEY_PLAIN, fontScale: 2, color: (255, 0, 0), thickness: 2)
    temur_son_giris = ekle( numara: "221821", temur_son_giris, current_time)
elif kurt_s[0]:
    cv2.rectangle(frame, (x, y), (w, h), (255, 0, 0), 3)
    cv2.putText(frame, text: "221837 Necmettin Sönmez", org: (x, h + 35), cv2.FONT_HERSHEY_PLAIN, fontScale: 2, color: (255, 0, 0), thickness: 2)
    kurt_son_giris = ekle( numara: "221837", kurt_son_giris, current_time)
elif baran_s[0]:
    cv2.rectangle(frame, (x, y), (w, h), (255, 0, 0), 3)
    cv2.putText(frame, text: "2218080072 Baran ASAR", org: (x, h + 35), cv2.FONT_HERSHEY_PLAIN, fontScale: 2, color: (255, 0, 0), thickness: 2)
    baran_son_giris = ekle( numara: "221872", baran_son_giris, current_time)
elif emir_s[0]:
    cv2.rectangle(frame, (x, y), (w, h), (255, 0, 0), 3)
    cv2.putText(frame, text: "221849 Emir ÇÖŞKÜN", org: (x, h + 35), cv2.FONT_HERSHEY_PLAIN, fontScale: 2, color: (255, 0, 0), thickness: 2)
    emir_son_giris = ekle( numara: "221849", emir_son_giris, current_time)
elif ishak_s[0]:
    cv2.rectangle(frame, (x, y), (w, h), (255, 0, 0), 3)
    cv2.putText(frame, text: "231833 Ishak YORGACI", org: (x, h + 35), cv2.FONT_HERSHEY_PLAIN, fontScale: 2, color: (255, 0, 0), thickness: 2)
    ishak_son_giris = ekle( numara: "231833", ishak_son_giris, current_time)
elif yusuf_s[0]:
    cv2.rectangle(frame, (x, y), (w, h), (255, 0, 0), 3)
    cv2.putText(frame, text: "221856 Yusuf DUMAN", org: (x, h + 35), cv2.FONT_HERSHEY_PLAIN, fontScale: 2, color: (255, 0, 0), thickness: 2)
    yusuf_son_giris = ekle( numara: "221856", yusuf_son_giris, current_time)
else:
    cv2.rectangle(frame, (x, y), (w, h), (255, 0, 0), 3)
    cv2.putText(frame, text: "Yabancı", org: (x, h + 35), cv2.FONT_HERSHEY_PLAIN, fontScale: 2, color: (255, 0, 0), thickness: 2)

```

Bu kod parçası program çalıştığı sürece sürekli çalışacak sonsuz bir döngüdür

- İlk olarak her döngü arasında 0,3 saniye bekler.
- Ardından webcam görüntüsünün her bir karesini ayırarak içinde yüz olup olmadığını kontrol eder.
- Ardından her bulunan yüzün fotoğraf karesi içerisindeki koordinatlarını belirleyip değişken içine kaydeder.
- Ardından tespit edilen yüzler daha önce tanıtılmış yüz verileri ile karşılaştırılır.
- Tespit edilen yüzün daha önce tanıtılan yüzlerle eşleşmesi halinde webcam'den alınan görüntüdeki yüzü mavi bir çerçeve içine alır ve altına ilgili öğrenciye ait numara ile ismi yazdırır.



Daha sonra öğrenci numarası, son giriş tarihi ve anlık zamanı ekle fonksiyonuna gönderir.

Fonksiyon daha öncede bahsedildiği gibi aradaki farkı ölçer ve 40 dk dan fazla zaman geçmesi halinde veriyi veri tabanına işler ve anlık zaman verisini geri gönderir ve böylece son giriş tarihi güncellenmiş olur

40 dakikanın altında bir süre geçmiş ise yeni bir veriyi veri tabanına işlemez ve eski son giriş zamanını geri gönderir. Bu sayede son giriş zamanı değişmemiş olur

Tespit edilen yüzün daha önce tanıtılan yüzlerle eşleşmesi halinde webcamden alınan görüntüdeki yüzü mavi bir çerçeve içine alır ve altına ilgili yabancı yazılır ve veri, veri tabanına işlenmez

```
cv2.imshow( winname: "deneme", frame)
if cv2.waitKey(10) & 0xFF == ord("q"):
    break
```

Klavyeden “q” tuşuna basılması halinde döngü kırılır.

```
finally:  
    # Kaynakları serbest bırak  
    cap.release()  
    cv2.destroyAllWindows()  
    connection.close()
```

Döngünün kırılmasından sonra:

- Webcam kapatılır
- Görüntünün gösterildiği pencere kapatılır
- Sql bağlantısı kapatılır
- Ve program sonlanır

## •OBS Sistemi: Login Ekranı

```
private void girisYapButonu_Click(object sender, EventArgs e)
{
    string connectionString = "Data Source=yuztanima.database.windows.net;Initial Catalog=yuztanima;User ID=Yusuf;Password=P@ssw0rd!";
    string yetki = comboBox1.SelectedItem?.ToString();
    string id = idTextBox.Text.Trim();
    string sifre = sifreTextBox.Text.Trim();

    if (string.IsNullOrEmpty(yetki))
    {
        MessageBox.Show("Lütfen bir yetki seçin.");
        return;
    }
    if (string.IsNullOrEmpty(id) || string.IsNullOrEmpty(sifre))
    {
        MessageBox.Show("Lütfen kullanıcı adı ve şifreyi girin.");
        return;
    }
}
```

Giriş ekranında verilerin girilip girilmediğinin kontrolü yapılır.

```
switch (yetki)
{
    case "Yönetici":
        sql = "SELECT * FROM ADMIN WHERE Admin_ID = @id AND Admin_Sifre = @sifre";
        prm1 = new SqlParameter("id", idTextBox.Text.Trim());
        prm2 = new SqlParameter("sifre", sifreTextBox.Text.Trim());
        break;

    case "Öğretmen":
        sql = "SELECT * FROM OGRETMEN WHERE Ogretmen_ID = @id AND Ogretmen_Sifre = @sifre";
        prm1 = new SqlParameter("@id", idTextBox.Text.Trim());
        prm2 = new SqlParameter("@sifre", sifreTextBox.Text.Trim());
        break;

    case "Öğrenci":
        sql = "SELECT * FROM OGRENCI WHERE Ogrenci_No = @id AND Ogrenci_Sifre = @sifre";
        prm1 = new SqlParameter("id", idTextBox.Text.Trim());
        prm2 = new SqlParameter("sifre", sifreTextBox.Text.Trim());
        break;
}
```

Seçilen yetkiye göre SQL'den veri çekilir.

```

using (SqlCommand cmd = new SqlCommand(sql, conn))
{
    cmd.Parameters.Add(prm1);
    cmd.Parameters.Add(prm2);

    DataTable dt = new DataTable();
    SqlDataAdapter da = new SqlDataAdapter(cmd);
    da.Fill(dt);

    if (dt.Rows.Count > 0)
    {
        switch (yetki)
        {
            case "Yönetici":
                Form2 form2 = new Form2();
                form2.Show();
                break;

            case "Öğretmen":
                ogretmenForm ogretmenForm = new ogretmenForm(Convert.ToInt32(id), "Data Source=yuztanima.database.windows.net;Initial Catalog=yuztanima;User ID=Yusuf;Password=P@ssw@rd!");
                ogretmenForm.Show();
                break;

            case "Öğrenci":
                ogrenciForm ogrenciForm = new ogrenciForm(Convert.ToInt32(id), "Data Source=yuztanima.database.windows.net;Initial Catalog=yuztanima;User ID=Yusuf;Password=P@ssw@rd!");
                ogrenciForm.Show();
                break;
        }
    }

    this.Hide();
}

```

Ve Seçilen veriler kullanıcın girdiği verilerle karşılaştırılır. Doğruysa gerekli sayfa açılır.

## OBS Sistemi: Yönetici Ekranı

```

public class sqlvariable
{
    public static SqlConnection baglanti = new SqlConnection("Data Source=yuztanima.database.windows.net;Initial Catalog=yuztanima;User ID=Yusuf;Password=P@ssw@rd!");
    public static void BaglantiKontrol(SqlConnection tempConnection)
    {
        if (tempConnection.State == ConnectionState.Closed)
        {
            tempConnection.Open();
        }
        else
        {
            // ... 
        }
    }
}

```

Sql ile bağlantı kurulmak istenildiğinde her seferinde open, close işlemi yapılmaması için oluşturulmuş bir sınıftır. BaglantiKontrol metodu her çağırıldığında eğer bağlantı durumu kapalı ise açar, açıksa herhangi bir işlem gerçekleştirmez.

```

3 başvuru
private void pencereFormu(Form kucukForm, object btnSender)
{
    if (guncelForm != null)
        guncelForm.Close();
    anasayfaGaziResim.Visible = false;
    guncelForm = kucukForm;
    kucukForm.TopLevel = false;
    kucukForm.FormBorderStyle = FormBorderStyle.None;
    kucukForm.Dock = DockStyle.Fill;
    this.panelDesktopPane.Controls.Add(kucukForm);
    this.panelDesktopPane.Tag = kucukForm;
    kucukForm.BringToFront();
    kucukForm.Show();
    lblTitle.Text = "Yönetim";
}

```

Yönetici ekranına giriş yapıldıktan sonra işlevli bir arayüz için yazılmış bir kısımdır. Ders yönetim, öğrenci yönetim ve öğretmen yönetim ekranlarının her birini ana form üzerinde belirli bir bölgede açar. Her açılan formu güncel form olarak tutup yeni form açıldığında kapatır.

```

private void ogrenciKaydetBut_Click(object sender, EventArgs e)
{
    try
    {
        SqlCommand idKontrolKomutu = new SqlCommand("SELECT COUNT(*) FROM OGRENCI WHERE Ogresnci_No = @pid", sqlvariable.baglanti);
        idKontrolKomutu.Parameters.AddWithValue("@pid", Convert.ToInt64(ogrenciIDTbox.Text));
        sqlvariable.BaglantiKontrol(sqlvariable.baglanti);
        int kayitSayisi = (int)idKontrolKomutu.ExecuteScalar(); //sorgu sonucu değer alınır
        if (kayitSayisi > 0)
        {
            throw new Exception("Öğrenci No bir başka öğrenci ile aynı!");
        }
        //eşleşen satırların sayısı döndürülür eğer 0 dan büyük ise eşleşen id ler vardır.
        SqlCommand ekleKomutu = new SqlCommand("Insert into OGRENCI(Ogresnci_No,Ogresnci_Adi,Ogresnci_Soyadi,Ogresnci_Sifre) values (@pid,@pad,@padi,@psoyad,@ppassword)", sqlvariable.baglanti);
        ekleKomutu.Parameters.AddWithValue("@pad", tboxName.Text);
        ekleKomutu.Parameters.AddWithValue("@psoyad", tboxSurname.Text);
        ekleKomutu.Parameters.AddWithValue("@ppassword", passwordTbox.Text);
        ekleKomutu.Parameters.AddWithValue("@pid", Convert.ToInt64(ogrenciIDTbox.Text));
        ekleKomutu.ExecuteNonQuery();
        tabloYenileme();
        tboxName.Text = "";
        tboxSurname.Text = "";
        passwordTbox.Text = "";
        ogrenciIDTbox.Text = "";
    }
    catch (Exception ex)
    {
        MessageBox.Show("Öğrenci kayıt sırasında bir hata oluştu: " + ex.Message);
    }
}

```

Yeni bir öğrenci ekleneceği zaman eğer yönetici var olan bir id girerse, girdiği id tablodaki eşleşen id sayısını tutar. Sorgu sonucu eğer 0'dan büyükse bu id ile bir Öğrenci olduğu anlamına gelir. Aynı kontrol Öğretmen için de geçerlidir.



## •OBS Sistemi: Öğrenci ve Öğretmen Ekranları

```
private void ShowAttendanceInfo()
{
    // öğrencilerin bilgilerinin gösterilmesi
    string query = @"
SELECT o.Ders_Adi AS [Ders Adı], o.Ogretmen_Adi AS [Öğretmen Adı], o.Ogretmen_Soyadi AS [Öğr
FROM DEVAMSIZLIK d
JOIN OGRETMEN o ON d.Ogretmen_ID = o.Ogretmen_ID
JOIN ISLENEN_TOPLAM_DERS itd ON d.Ogretmen_ID = itd.Ogretmen_ID
WHERE d.Ogrenci_No = @studentId";

    SqlCommand command = new SqlCommand(query, connection);
    command.Parameters.AddWithValue("@studentId", Ogrenci_No);

    try
    {
        // !!bağlantı açıldı!!
        connection.Open();

        DataTable dataTable = new DataTable();
        dataTable.Load(command.ExecuteReader());
        dataGridViewyoklama.DataSource = dataTable;
    }
    catch (Exception ex)
    {
        MessageBox.Show("Veritabanı hatası: " + ex.Message);
    }
    finally
    {
        // !!bağlantı kapandı!!
        if (connection.State == ConnectionState.Open)
        {
            connection.Close();
        }
    }
}
```

Öğrenci formu içinde SQL'deki Databas'e bağlanarak öğrencinin idsi ile aynı olan yoklama tablosundaki tüm verilerin çekilmesi. Bununla beraber öğretmenin id'si kullanılarak öğretmenin bilgilerinin SQL Database'deki öğretmenlerin bulunduğu tablodan alınması ve aynı işlemin öğrencinin id'si kullanılarak da yapılması. Ayrıca bunların formsta oluşturulan datagridviewe tablo olarak aktarılması. Try catch bloğu ile veritabanında verilerin alınması ile ilgili bir hata olup olmadığının alınması.

```

try
{
    // !!bağlantı açıldı!!
    connection.Open();

    SqlDataReader reader = command.ExecuteReader();
    if (reader.Read())
    {
        ogrenci_isim.Text = reader["Ogrenci_Adi"].ToString();
        ogrenci_soyad.Text = reader["Ogrenci_Soyadi"].ToString();
        ogrenci_id.Text = reader["Ogrenci_No"].ToString();
    }
}
catch (Exception ex)
{
    MessageBox.Show("Veritabanı hatası: " + ex.Message);
}
finally
{
    // !!bağlantı kapandı!!
    if (connection.State == ConnectionState.Open)
    {
        connection.Close();
    }
}

```

Öğrenci formunda öğrencinin kendi bilgilerinin ekranda görülmesi için oluşturulan try catch bloğu. Veritabanında bir hata olup olmadığı kontrol edilir.

```

1 başvuru
private void FillLecturerInfo()
{
    string query = "SELECT Ogretmen_Adi, Ogretmen_Soyadi, Ogretmen_ID FROM OGRETMEN WHERE Ogretmen_ID = @Ogretmen_ID";
    SqlCommand command = new SqlCommand(query, connection);
    command.Parameters.AddWithValue("@Ogretmen_ID", Ogretmen_ID);

    try
    {
        // bağlantı açıldı
        connection.Open();

        SqlDataReader reader = command.ExecuteReader();
        if (reader.Read())
        {
            ogretmen_isim.Text = reader["Ogretmen_Adi"].ToString();
            ogretmen_soyad.Text = reader["Ogretmen_Soyadi"].ToString();
            ogretmen_id.Text = reader["Ogretmen_ID"].ToString();
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show("Veritabanı hatası: " + ex.Message);
    }
    finally
    {
        // bağlantı kapandı
        if (connection.State == ConnectionState.Open)
        {
            connection.Close();
        }
    }
}

```

Öğretmenin bilgilerinin ekranda gösterilmesi ve bağlantının kontrolü.

```

private void ShowAttendanceInfo()
{
    // Öğretmenin verdiği dersin tüm yoklama bilgileri gösterildi
    string query = @"
        SELECT d.Ogrenci_No, o.Ogrenci_Adi, o.Ogrenci_Soyadi, d.Mevcut, itd.Toplam
        FROM DEVAMSIZLIK d
        INNER JOIN OGRENCI o ON d.Ogrenci_No = o.Ogrenci_No
        INNER JOIN ISLENEN_TOPLAM_DERS itd ON d.Ogretmen_ID = itd.Ogretmen_ID
        WHERE d.Ogretmen_ID = @Ogretmen_ID";

    SqlCommand command = new SqlCommand(query, connection);
    command.Parameters.AddWithValue("@Ogretmen_ID", Ogretmen_ID);

    try
    {
        // Bağlantıyı aç
        connection.Open();

        SqlDataReader reader = command.ExecuteReader();
        while (reader.Read())
        {
            string studentId = reader["Ogrenci_No"].ToString();
            string studentName = reader["Ogrenci_Adi"].ToString();
            string studentSurname = reader["Ogrenci_Soyadi"].ToString();
            int exist = Convert.ToInt32(reader["Mevcut"]);
            int total = Convert.ToInt32(reader["Toplam"]);

            // Öğrenci bilgilerini DataGridView'e ekle
            dataGridViewOgretmen.Rows.Add(studentId, studentName, studentSurname, exist, total);
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show("Veritabanı hatası: " + ex.Message);
        MessageBox.Show("Hata detayları: " + ex.ToString());
    }
    finally
    {
        // Bağlantıyı kapat
        if (connection.State == ConnectionState.Open)
        {
            connection.Close();
        }
    }
}

```

Öğretmenin id'si ile eşleşen tüm öğrencilerin yoklamalarının tablo şeklinde gösterilmesi. Öğretmen ve öğrenci id'leri kullanılarak tablolarda veri akışının sağlanması ve öğretmen ve öğrenci bilgilerinin ekranda gösterilmesi.

## 5)Sonuç ve Değerlendirme

Proje kapsamı dahilinde olan öğrencilerin yoklamalarının yüzlerinin önceden tanıtılarak hazırlanan yoklama sistemine kaydedilmesi hedefine ulaşılmıştır. Proje hedeflenene gereksinimler yönünden tamamen gerçekleştirilmiştir.

Projenin geliştirilmesi için öneriler için şunlar söylenebilir:

### 1) Güvenlik duvarları

Öğrencilerin parolalarının çeşitli şifreleme yöntemleriyle (SHA256, MD5, BLAKE2b vb.) saklanarak veriye direkt ulaşımındaki güvenlik zafiyetinin önüne geçilmesi.

Öğrencilerin kimlik doğrulamalarının Gazi Üniversitesindeki cep telefon numaralarıyla veya önceden belirlenmiş olan telefon numaralarıyla iki aşamalı doğrulama sistemi koyulması.

### 2) Öğretmenlerin ders programlarının geliştirilmesi

Oluşturulan prototipte öğretmenlerin her gün bir dersi olacağı ve her öğretmenin bir ders verdiği varsayılarak veri tabanı oluşturulmuştur. Gerçek hayatta ders programları ve verilen dersler daha kompleks olduğundan bu kapsam genişletilebilir.

### 3) Veri tabanı sunucusu

Veri tabanın bağlı olduğu Azure Sql Sunucularının daha kapsamlı ve üniversiteye bağlı olan bir sunucuyla kullanılması; girişlerdeki aksamalarının, yoğun öğrenci girişinde oluşacak sorunların önüne geçilmesi için kullanılabilir.

### 4) Gazi Üniversitesi obs sistemi ile entegre

Öğrenciler için oluşturulan bu yoklama sistemini diğer tüm işlemleri yapabileceği Gazi Üniversitesi obs sistemi ile entegre etmek sistem verimliliği açısından daha yararlıdır.

## **5) İstisnai durumların çözümü**

Raporlu öğrenci, izinli öğrenci gibi durumların farklı bir şekilde işlenmesinin sisteme entegresi sistemi daha kullanılabilir hale getirebilir.

## **6) Arayüz ve Geliştirmeler**

Arayüzün daha estetik ve kullanışlı hale getirilmesi uygulamanın kullanıcı dostu hale gelmesi yönünden önemlidir bunun yanında arayüze SSS, üniversitenin sosyal medya hesapları eklenebilir.

Uygulamanın mobil versiyonu yapıp öğrencilerin daha kolay ve efektif bir şekilde yoklamalarının takiplerini yapması sağlanabilir.

## **7) Yüz tanıma sisteminin geliştirmeleri**

Yüz tanıma sistemine derinlik algısı, daha ölçekli renk ayrımı, daha hassas tanıma ve anlamlandırma gücü sağlanılarak olası hatalar en aza indirilebilir. Bu sayede öğrenciler geldikleri veya gelmedikleri herhangi bir derste mağdur olmayacaktır.

## **Kazanımlar:**

Proje genel anlamda bir programlama dilinin efektif kullanımı, ekip ile proje geliştirme, ortak sorun çözümü, iletişim hakkında tüm ekip üyelerine kazanımlar katmıştır. Bunun yanında özellikle şu konularda kazanımlar sağlamıştır:

### **1) Windows Forms**

Proje, uygulamanın arayüz kısmının temeli olan Windows formsun efektif kullanımı, temel bilgiler, tasarım konularında ekip üyelerine kazanımlar sağlamış ve temel düzeyde onları etkili derecede geliştirmiştir.

### **2) Sql**

Verilerin depolandığı sql veri tabanı yönetim sisteminin kullanımı verilerin işlenmesi, depolanması, ilişkisel veriler ve tablolar hakkında ekip üyelerine deneyim ve kazanımlar katmıştır. Bir veri tabanına bağlanıp verilerin işlenmesi hakkında bilgi kazanan ekip üyeleri bunu yaparken ortaya çıkan sorunların çözümünü de çeşitli kaynaklara başvurarak çözüp veri tabanında çıkan sorunların çözümünde de deneyim sahibi olmuştur.

### **3) Sql Server**

Proje içinde kullanılan Azure sql server kullanımı kullanıcıların uzaktaki bir veri tabanına bağlanıp onu nasıl kontrol edeceği, oluşan bağlantı sorunlar, sunucuların güvenlik duvarlarının kontrolü, veri tabanın yedeklenmesi konularında tecrübe kazanmıştır.

### **4)Visual Studio**

Ekip üyeleri projede genellikle kullanılan Visual Studio bu IDE'nin kullanımında deneyim kazanmışlar. Üyeler VS'yi daha efektif kullanmayı VS'de dosya paylaşımı, dosya eklenmesi, kod düzenlemesi gibi benzeri birçok kullanım alanında kendilerini geliştirmişlerdir.

## **5)C#**

Arayüz tasarımıında kullanılan C# dilinde ekip üyeleri döngüler oluşturma, if else yapıları, try catch bloklar, nesne oluşturma ve kullanma gibi konularda deneyim kazanmışlardır. Oluşan sorunların beraber veya tek başına çözülmesi de genel olarak programlama ve sorun çözümü konusunda üyelere tecrübe kazandırmıştır.

## **6)C# ve Sql bağlantısı**

C# ile Sqlde bulunan bir veri tabanın manipüle edilmesi ve bunun için özelleşmiş olan, sql komutlarını barındıran kütüphanelerin kullanımı kullanıcılara bu konuda deneyim sağlamıştır.

## **7)Python ve Pycharm kullanımı**

Yüz tanıma bölümünde kullanılan Python programlama dili ve pycharm IDE'si de C# kullanımında olduğu gibi bu alanda çalışan ekip üyelerine genel programla hakkında katkı sağlamıştır bunun yanında burada oluşan sorunların çözümü de bu dile ve IDE'ye olan hakimiyeti artırmıştır.

## **8) OpenCV kütüphanesi**

Yüz tanıma bölümünde kullanılan OpenCV kütüphanesi ile bu alanda olan hakimiyet artırılmıştır. Bunun yanında temel olarak yapay zekanın çalışma prensipleri de öğrenilmiştir.

## **9) Genel Kazanımlar**

En başta da belirtildiği gibi en önemli konulardan olan bu projenin genel olarak ekip üyelerini bir proje nasıl geliştirilir, proje üyeleri ile iletişim nasıl olur, sorunlar nasıl çözülür, planlamalar neye göre yapılır, proje eksikleri fark edilir gibi konular hakkında deneyim kazanmışlardır.

## Kaynakça

### Programlama Dili ve IDE'ler

1. C# dili eğitimi, Youtube,  
<https://www.youtube.com/watch?v=VytV9w2dAFs&list=PLKnjBHu2xXNPkeQtMOJczzEO6LK5OV35K>
2. Visual Studio, Youtube,  
<https://www.youtube.com/watch?v=RJ-4hIXK-Ms&list=PLURN6mxdcwL960S-bRuf1F6K09yzNjgcn>  
[https://www.youtube.com/watch?v=liKyYxO29GE&list=PL4Igle6SmfRUwgPXbp\\_Plyv33ocA2MHwb](https://www.youtube.com/watch?v=liKyYxO29GE&list=PL4Igle6SmfRUwgPXbp_Plyv33ocA2MHwb)

### Veritabanı yönetimi ve işlemleri

1. C# SQL dersleri, Youtube,  
<https://www.youtube.com/watch?v=3tnKh3AOaCI&list=PLD54hVH5EaMbGsUpCm96ZRAbNpREfrvpR>
2. SQL Server Eğitimi, Udemy,  
[https://www.udemy.com/share/104rZA3@arMv3iPsZOg9\\_JsKm2zv91XJqZx65lat26HNH1FnSmRapZJ3WqbeOJEWU0CuW6yZ/](https://www.udemy.com/share/104rZA3@arMv3iPsZOg9_JsKm2zv91XJqZx65lat26HNH1FnSmRapZJ3WqbeOJEWU0CuW6yZ/)
3. NuGet Paketleri  
<https://www.nuget.org/packages/Microsoft.Data.SqlClient>  
<https://www.nuget.org/packages/System.Data.SqlClient>
4. Kaynak  
<https://www.sqlservertutorial.net/>
5. Microsoft Azure SQL, Youtube,  
<https://www.youtube.com/watch?v=6joGkZMVX4o&t=379s>

### Yüz Tanıma ve Python

1. Python dersleri, Udemy,  
[https://www.udemy.com/share/104DN63@IPXuRM2T\\_2PQd5yHGWjs\\_i1KvFBVGuU0cgTp-GlbOz4q95okiKenqagmzEal72RX/](https://www.udemy.com/share/104DN63@IPXuRM2T_2PQd5yHGWjs_i1KvFBVGuU0cgTp-GlbOz4q95okiKenqagmzEal72RX/)
2. OpenCV eğitimi, Youtube,  
[https://youtube.com/playlist?list=PLkJUWWxr1XL7ZKq5oL2DtpljgQO4mLVA1&si=IX\\_FWJkehzJxIEVx](https://youtube.com/playlist?list=PLkJUWWxr1XL7ZKq5oL2DtpljgQO4mLVA1&si=IX_FWJkehzJxIEVx)
3. Kütüphane, Github,  
[https://github.com/CodeEZ-Dev/face-recognition\\_dlib\\_library](https://github.com/CodeEZ-Dev/face-recognition_dlib_library)



## **Kullanıcı Arayüzü ve Tasarım**

1. <https://youtu.be/d13riSy08ZA?si=hkrdq5vYbLQtopCo>
2. <https://www.youtube.com/watch?v=BtOEztT1Qzk&t=6s>
3. <https://www.youtube.com/watch?v=mCntyIFh04s&t=179s&pp=ygUadmlzdWFsIHN0dWRpbyB0YWJsbyBkZXNpZ24%3D>
4. <https://icons8.com/app/windows>
5. [https://youtube.com/playlist?list=PLD54hVH5EaMb9eLt3oZMuP0EMJaHalwd&si=QFTiKsdYTPUKZR\\_o](https://youtube.com/playlist?list=PLD54hVH5EaMb9eLt3oZMuP0EMJaHalwd&si=QFTiKsdYTPUKZR_o)

## **Genel Kaynaklar**

1. <https://learn.microsoft.com/tr-tr/>
2. <https://stackoverflow.com/>
3. <https://www.w3schools.com/>
4. <https://www.sqlsvertutorial.net/>
5. <https://azure.microsoft.com/tr-tr>

## **Hataların Belirlenmesi ve Yorumların Düzenlenmesi**

1. <https://chatgpt.com/>
2. <https://gemini.google.com/app?hl=tr>

## **Proje Dosyaları ve Sunum Videosu**

[https://drive.google.com/drive/folders/1S3p-sZZCN\\_TFZ22ToBRaTTXN3bbdmqEv?usp=sharing](https://drive.google.com/drive/folders/1S3p-sZZCN_TFZ22ToBRaTTXN3bbdmqEv?usp=sharing)