

ERZİNCAN BİNALİ YILDIRIM ÜNİVERSİTESİ
TERCAN MESLEK YÜKSEKOKULU



Her gün için en az bir sayfa kullanılacak (fazlada kullanılabilir.) varsa sayfa ekleri arka kısma ilave edilecek.

Yapılan Çalışma ile ilgili çizim ve açıklamalar

İlk günüm tanışma ile başladı. Bana nasıl öğrenebileceğim nasıl çalışmam gerektiği hakkında bilgiler verdi. İlk olarak not almak için keep kullanmaya başladım. Hangi ideyi kullanacağımızı gösterdi. İDE : geliştirme ortamı.

İde üreticisi olan JetBrains'in idelerini kullanmak için kaydolduk. Python öğrenmeye başlayacağım için PyCharm'ı kullanacağım. Bana PyCharm'ı kurmamı ve git kullanımını araştırmamı söyledi. C++ da öğrendiklerimi denemek için ek olarak CLion kurulumunu da yapmamı istedi ve ilk günümüz bitti.

Stajımda yaptığım her şeyi commitledim, projelendirdim, tamamını aşağıdaki adreste bulabilirsiniz.

Git hub adresim : <https://github.com/eemreturk0?tab=repositories>

Tarih	İşin Adı Ne Amaçla ve Nereye yapıldığı	İş Yeri Yetkilisi İmza	Sayfa No

ERZİNCAN BİNALİ YILDIRIM ÜNİVERSİTESİ
TERCAN MESLEK YÜKSEKOKULU



Her gün için en az bir sayfa kullanılacak (fazlada kullanılabilir.) varsa sayfa ekleri arka kısma ilave edilecek.

EK-VI

CLion'u açtık ve eskiden yapmış olduğum bir projeyi ide de açmayı öğrendim.

Proje açmak : File -> Open -> proje yolu -> Trusct project -> New Window yolu ile proje eklemesi yapılır.

PyCharm'a geçtik ve ilk olarak atamaları ve yazdırma kodlarını gösterdi.

Random fonksiyonunun kullanımını ve if elif ve else yapılarını gördüm. Bunlara condition yapıları durum yapıları denir. Bölme ve modlama operatörlerini ve kullanımını gördüm. For ve While döngüsünü basit olarak gösterdi.

Kodların kolayca bulunabilmesi için küçültme ve büyültme kısayolları öğrenildi.

Bilgisayarımıza git indirdik.Git huba kaydolup projemiz için git hubda bölüm oluşturduk. Pycharm da git'i bağladık ve artık projemizi kolayca git huba aktarıp bilgisayarlar arasında kolayca aktarım sağlanabiliyor ve projelerde git'in kullanımının önemini öğrendim. Pycharm'da .ignore dosyası oluşturduk. git ignore dosyası oluşturmayı öğrendim. Git ignore dosyası git huba aktardığımız projemizde nelerin aktarılmamasını yazdığımız dosyadır.

Atamalar ve yazdırma :

```
1 x=5
2 y="m"
3 z="cihad"
4 n="ekrem"+z
5 print(n)
```

Bu yazdırma sonucunda ekranımıza ekremcihad çıktısını alırız.

If Else yapıları:

```
# x = 1 | # None
# if x:
#     a = 5
# elif x:
#     a = 7
# else:
#     a = 6
# print(a)
```

Burada x'imiz 1 yani True olduğu için ilk if'e girer ve diğerlerine girmeden 5'i yazdırır. Eğer x=0 olsaydı 7 yi yazdırırdı.

For döngüsü:

```
# a="ekrem"  
# for x in a:  
#     print(x)
```

-> bir a dizisi olur ve x'i döngüde döndürerek ekremin alt alta harflerin çıktısını verir.

While döngüsü:

```
# a = 2
# while a < 5:
#     print(a)
#     a += 1
```

-> Döngüde a=2 den başlatarak her seferinde bir atılcak şekilde 5 e kadar gider çıktı olarak : 2,3,4 verir.

Tarih	İşin Adı Ne Amaçla ve Nereye yapıldığı	İş Yeri Yetkilisi İmza	Sayfa No

ERZİNCAN BİNALİ YILDIRIM ÜNİVERSİTESİ
TERCAN MESLEK YÜKSEKOKULU



Her gün için en az bir sayfa kullanılacak (fazlada kullanılabilir.) varsa sayfa ekleri arka kısma ilave edilecek.

EK-VI

Yapılan Çalışma ile ilgili çözüm ve açıklamalar

Bugün pythonda asal sayı bulma kodunu yazmaya başladık. isPrime adında fonksiyon oluşturup number atayarak

asal sayı kodunu yazdık. Kodun başlayacağı yeri belirterek while döngüsünde dışardan sayı alıp fonksiyon çağırarak

sayı asal mı değil mi bulan kodu tamamladık. Ardından git hub'a kodumuzu aktardık.

Sayı tahmin kodunu yazdık ve githuba aktardık. sayı tahmin kodunda while döngüsünde alınan sayıya şart koşmayı

yani örnek olarak 4 rakamdan oluşma koşulunu sağlamayı öğrendim. İstenilen int değeri girilmez ise except koşulu

ile ekrana print ederek burada hata var mesajı yazdırdık.

aynı gün içinde yapmam için yıldızlardan şekil yapma ödevleri verdi bana ve dersimiz sona erdi.

Asal sayı :

```
turke
def isPrime(number):
    myDividers=[]
    prime=True
    for i in range(2, number):
        if (number % i) == 0:
            myDividers.append(i)
            prime=False
    return prime, myDividers

if __name__ == "__main__":
    while True:
        sayi = int(input("Sayıyı girin :"))
        result, notPrime = isPrime(sayi)
        if result:
            print(sayi, "Asal sayıdır.\n\n")
        else:
            print(sayi, "Asal sayı değildir. ", notPrime)
```

Sayı tahmin:

```

def checkResult(tahmin, myNumber):
    if tahmin == myNumber:
        return True, ""
    else:
        if tahmin > myNumber:
            return False, "Tutulan sayı daha küçük"
        else:
            return False, "Tutulan sayı daha büyük"

turke

def getNumberFromUser():
    guessLocal = None
    while not guessLocal:
        try:
            temp = input("Tahmininizi giriniz :")
            if len(temp) == 4: # length uzunluk demek len karakter sayısını verir
                guessLocal = int(temp)
            else:
                print("Tahmin 4 haneli olmalı")
        except Exception as ex:
            print("Burada Hata Var Tahmin Int Olmalı")
    return guessLocal

if __name__ == "__main__":
    result = False
    msg = ""
    guess = 0
    sayi1 = 5234

```

Tarih	İşin Adı Ne Amaçla ve Nereye yapıldığı	İş Yeri Yetkilisi İmza	Sayfa No

ERZİNCAN BİNALİ YILDIRIM ÜNİVERSİTESİ
TERCAN MESLEK YÜKSEKOKULU



EK-VI

İlk olarak yıldızlardan şekil yapmayı öğrenerek başladım. Benim araştırıp yaptığım kodu inceledik. mergedTriangleStar , rectangelStar, r everseTriangleStar, triangleStar olarak 4 tane yıldız şeklini döngüler ile yapmayı öğrendim.

Nesne tabanlı programlamaya geçiş yaptık. Nesne tabanlı programlamanın mantığını öğrendim. Nesne tabanlı programlama : gerçek hayatı programlamaya uyarlamak olarak düşünülebilir. Örnek olarak bir öğrenci otomasyon sistemi yazmak istiyoruz bunun için öğretmenleri öğrencileri ve kursları bir nesne olarak oluşturmamız gerekiyor böyle bir sistemi oluşturmamız içinde her bir nesnenin yapısını tanımlayıp daha sonra bu yapılardan bu nesnelerini oluşturmamız gerekiyor.

Obje Nesne nedir ?

Etrafımıza baktığımızda aslına her bir eşyanın bir obje olduğunu görüyoruz. Örneğin bir televizyon kumandası.

Kendi içinde değişik özellikleri ve fonksiyonları bulunuyor. Örneğin kumandanın markası, tuşları aslında bu kumandanın özellikleridir. Kumandanın kırmızı tuşuna bastığımızda televizyonun kapanmasında methodlarındandır.

triangleStar :

```

while True:
    height = int(input("yükseklik giriniz :"))
    if height < 4:
        print("4 ve 20 arasında olmak zorunda")
        #print("4 ve 20 arasında olmak zorunda")
        break
    else:
        for i in range(height):
            #for j in range(height-i-1):
            print(" "*(height-i-1),end="")
            print("*"*(2*i+1), end="")
            #for j in range(2*i+1):
            #    print("*",end="")

        print()

```

```

yükseklik giriniz :4
    *
   ***
  *****
 *****

```

çıktı ise bu şekilde alınır.

Nesne oluşturmak ve özellik atamak :


```
class Araba():  
    model = "kanqoo"  
    renk = "gümüſ"  
    beygir_gücü = 130  
    silindir = 8  
  
araba = Araba() # --> araba objesi oluſturuldu.  
  
print(araba.model)  
print(araba.renk)  
print(araba.beygir_gücü)  
print(araba.silindir)
```

Tarih	İſin Adı Ne Amaçla ve Nereye yapıldığı	İſ Yeri Yetkilisi İmza	Sayfa No

ERZİNCAN BİNALİ YILDIRIM ÜNİVERSİTESİ
TERCAN MESLEK YÜKSEKOKULU



Her gün için en az bir sayfa kullanılacak (fazlada kullanılabilir.) varsa sayfa ekleri arka kısma ilave edilecek.

Nesne tabanlı programlamada objeye init fonksiyonu oluşturarak özellikleri parametre vererek objemize özelliklerini vermeye başladık. init fonksiyonu içinde özellik verme "self.model=model" örnek olarak init fonksiyonumuzdan kullanarak model özelliğini ekledik. Cars adında obje oluşturarak objemize gereken özellikleri atamasını yaptık. artık yazdırma işleminde objenin özelliklerini yazdırıyor ekrana. Sonradan objede değişiklik yaparken cars = Cars(silindir=4) olarak yazdığımızda objenin silindiri 4 silindir olarak değiştirilir ve düzenlenmiş olur.

Nesne tabanlı programlamada Methodlar

Bir sınıf içerisinde method tanımlama. Yazılımcı adında yeni bir class oluşturduk ve içine yazılımcı özellikleri isim, soyisim, numara, maaş, bildiği diller olarak tanımladık. Method olarak yeni bir fonksiyon olarak bilgileri göster adında bir fonksiyon yazdık. yazılımcı objemizi oluşturup içine özelliklerini girdikten sonra artık bilgileri göster fonksiyonumuzu kullanarak istediğimiz yazılımcının bilgilerini gösteren methodu yazmış olduk. Dil ekle adında yeni bir fonksiyon oluşturarak olan bir listeye append yapmayı öğrendim. artık dil ekle fonksiyonunu çağırarak yazılımcıya bir yazılım dili ekleyen methodu yaptık. Yeni bir method olarak yazılımcının maaşına zam yap methodu kodunu yazdık. Aynı şekilde fonksiyon çağırarak bir yetkili yazılımcının maaşına zam yapabilir gibi düşünülebilir. Oluşturduğumuz fonksiyonlar method olduğu için objede kullanıldığından self referansını göndermemiz gerekiyor. Tekrar ederek ve kodları incileyerek günü bitirdim.

Classımızın oluşturulması :

```
class Cars():  
    def __init__(self,model,renk,silindir):  
        print("init fonk. çağırıldı.")  
        self.model=model  
        self.renk=renk  
        self.silindir=silindir  
  
car = Cars("Kangoo", "Gümüş", 4)  
car1 = Cars("T0GG", "Beyaz", 8)  
print(car.model)  
print(car1.model)  
print(car.silindir)  
print(car1.silindir)
```

Methodların oluşturulması :

```
def bilgileriGöster(self):
    print("""
        Yazılımcı özellikleri

        isim : {}
        soyisim : {}
        numara : {}
        Diller : {}

        """).format(self.isim, self.soyisim, self.numara, self.diller))

turke
def dil_ekle(self, yeni_dil):
    print("Dil ekleniyor...")
    self.diller.append(yeni_dil)
```

Yazılımcı objemizin oluşturulması ve dil ekleme methodunun kullanımı :

```
yazılımcı = Yazılımcı("Emre", "TÜRK", 1453, ["Python", "C", "C++"])
yazılımcı.dil_ekle("php")
```

Tarih	İşin Adı Ne Amaçla ve Nereye yapıldığı	İş Yeri Yetkilisi İmza	Sayfa No

ERZİNCAN BİNALİ YILDIRIM ÜNİVERSİTESİ
TERCAN MESLEK YÜKSEKOKULU



EK-VI



Her gün için en az bir sayfa kullanılacak (fazlada kullanılabilir.) varsa sayfa ekleri arka kasma ilave edilecek.

Nesne tabanlı programlamada Kalıtım

Kalıtım bir sınıfın başka bir sınıftan özelliklerini ve methodlarını miras almasına denir.

Kalıtım nerede işimize yarar ? Örneğin, bir şirketin çalışanlarını tasarlamak için sınıflar oluşturuyoruz. Bunun için Yönetici, Proje direktörü, İşçi gibi sınıflar oluşturmamız gerekiyor. Aslında baktığımız zaman bu sınıfların hepsinin belirli ortak methodları ve özellikleri bulunuyor. O zaman bu ortak özellikleri ve metodları tekrar tekrar bu sınıfların içinde tanımlamak yerine bir tane ana class tanımlayıp bu classın özelliklerini ve metodlarını almalarını sağlayabiliriz. Kalıtımın temel mantığı budur.

```
class Calisan():
    # turke
    def __init__(self, isim, maas, departman):
        print("çalışan sınıfının init fonksiyonu.")

        self.isim=isim
        self.maas=maas
        self.departman=departman

    # turke
    def bilgileriGöster(self):
        print("Çalışan sınıfın bilgileri : ")
        print(" isim : {}\n maas : {}\n departman : {}\n".format(self.isim, self.maas, self.departman))

    # turke
    def departman_degis(self, yeni_departman):
        self.departman=yeni_departman

# turke
class Yönetici(Calisan):
    pass

yonetici= Yönetici("Emre TÜRK", 3000, "PC")
yonetici.departman_degis("PC HİZMETLERİ")

print(dir(yonetici))

print(yonetici.isim, yonetici.maas, yonetici.departman)
```

Burada çalışan sınıfı oluşturduk ve iki metod tanımladık. Aşağıda ise yönetici sınıfı tanımladık ve bir

yönetici adında obje oluşturup özelliklerini girdik. yazdığımızda yöneticinin özelliklerini çalışan sınıfından miras alması yapıldı. yönetici objesini kullanarak örnek bir departman değiştirme metodu kullanıldı.

Nesne Tabanlı Programlama - Özel Metodlar

Özel metodlar bizim özel olarak çağırmadığımız ancak her classa ait metodlardır. Bunların çoğu biz tanımlamasak bile python tarafından varsayılan olarak tanımlanır. Ancak bu metodların bazılarını da özel olarak bizim tanımlamamız gerekmektedir. Init metodu bu metodlara bir örnektir.

Örnek kod :

```
class Kitap():
    def __init__(self, isim, yazar, sayisi, tür):
        print("init fonksiyonu")
        self.isim = isim
        self.yazar = yazar
        self.sayisi = sayisi
        self.tür = tür

    def __str__(self):
        return "isim: {}\nYazar: {}\nSayfa Sayısı: {}\nTürü: {}".format(self.isim, self.yazar, self.sayisi, self.tür)

kitap = Kitap("İstanbul Hatırası", "Ahmet Ümit", 561, "Polisiye")

print(kitap)
```

```
init fonksiyonu
isim: İstanbul Hatırası
Yazar: Ahmet Ümit
Sayfa Sayısı: 561
Türü: Polisiye
```

Tarih	İşin Adı Ne Amaçla ve Nereye yapıldığı	İş Yeri Yetkilisi İmza	Sayfa No

ERZİNCAN BİNALİ YILDIRIM ÜNİVERSİTESİ
TERCAN MESLEK YÜKSEKOKULU



MESLEK YÜKSEKOKULU

Her gün için en az bir sayfa kullanılacak (fazlada kullanılabilir.) varsa sayfa ekleri arka kısma ilave edilecek.

EK-VI

Dışardan alınacak şekilde 3 not 1 proje notu alıp not ortalamasını bulan kodu yazdık. Bunu nesne tabanlı programlamaya uyarlamaya başladık. Student adında yeni class yarattık. init fonksiyonu oluşturarak özellik olarak not1, not2, not3 ve proje notu olarak özellikleri belirttik. ortalama için self.avarege kullanmayı öğrendim. yaptığımız student liste student i eklemeyi öğrendim append yapmayı. Objeyi direk yazdırmada çıkan sorunda __str__ ve __repr__ ekleyerek classımıza sorunu gideririz. Notların ve proje notunun getirisi aynı olmayacağı için onlara dersin getirisini belriten eklemeleri yaptık. Add student adında fonksiyon haline getirerek student objesi oluşturan fonksiyonu tamamladık. Whille döngüsünde student ekleme ve yazdırma işlemlerini hallettik. ve tekrar ederek kod inceleyerek gün bitti.

```

GRADEONE_RATE = 0.15
GRADETWO_RATE = 0.2
GRADETREE_RATE = 0.25
PROJECTGRADE_RATE = 0.4

turke

class Student():
    turke
    def __init__(self,name:str,grade1:int,grade2:int,grade3,project_grade):
        self.name=name
        self.grade1=grade1
        self.grade2=grade2
        self.grade3=grade3
        self.project_grade=project_grade
        self.average = (grade1*GRADEONE_RATE + grade2*GRADETWO_RATE + grade3*GRADETREE_RATE + project_grade*PROJECTGRADE_RATE)

    turke
    def __repr__(self):
        return f"student:grade1:{self.grade1} grade2:{self.grade2} grade3:{self.grade3} project_grade:{self.project_grade} average:{self.average}"

    turke
    def __str__(self):
        return self.__repr__()

print("*****")
print()
print("\t Not Ortalaması Bulma Python \t")
print()
print("*****")
studentlist1 = []

```

```

def add_student():
    not1 = int(input("1.Notunuzu girin :"))
    not2 = int(input("2.Noutunuzu girin :"))
    not3 = int(input("3.Notunuzu girin :"))
    proje = int(input("Proje Notunuzu girin :"))
    student1 = Student("Emre", not1, not2, not3, proje)
    studentlist1.append(student1)

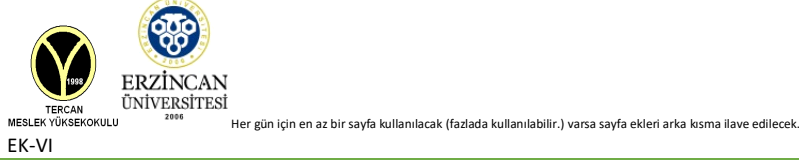
while True:
    #dictinoary
    add_student()

    print("Öğrenciler : ",studentlist1)

```

Tarih	İşin Adı Ne Amaçla ve Nereye yapıldığı	İş Yeri Yetkilisi İmza	Sayfa No

ERZİNCAN BİNALİ YILDIRIM ÜNİVERSİTESİ
TERCAN MESLEK YÜKSEKOKULU



Bugün beraber ders gördüğüm arkadaşımızın projesine bakarak başladık projede yapılan kodları inceledik.

Web siteleri için QR kod oluşturucu bir program yaptık. Projemiz için xml diyagram oluşturmamı araştırmamı

istendi ve araştırmaları yapıp ufak bir school için diyagram yaptım. Kalıtım konusunu tekrar gözden geçirdik

ve başlayacağımız school system için inceledik.

Super fonksiyonu öğrenildi. Modül mantığı öğrenildi. Kendi kütüphanemizi oluşturmamıza yarar. bütün projelerde import edilerek kullanılır.

import etme -> import modül_adi olarak yapılır.

Math Modülü

dir(math) modüldeki tüm fonksiyonları getirmeyi sağlar.

help(math) -> tanımını içeriğini gösterir.

Okul projesine başlandı. SchoolSystem dosyasında classlar ve fonksiyonlar yazıldı. Model.py dosyası oluşturularak peopletype, Lessontype, Lesson, Room, Teacher, Student ve schhol classlarımız oluşturuldu.

Bu classlarımızı schoolSystem.py dosyamızda çağırarak kullanıyoruz.


```

class PeopleType(enum.Enum):
    STUDENT = 1
    TEACHER = 2
    ADMIN = 3

turke *

class LessonType(enum.Enum):
    MATH = 1
    FIZIK = 2
    KIMYA = 3
    ING = 4
    TURKCE = 5
    TARİH = 6

turke *

class Lesson():
    turke *
    def __init__(self, lesson_type: LessonType, code: str):
        self.lesson_type = lesson_type
        self.code = code
        self.lessonRoom = None

```

```

turke *

class Room():
    turke
    def __init__(self, room_name):
        self.room_name = room_name

```

```

class People:
    turke
    def __init__(self, name, type: PeopleType):
        print("People sınıfının init fonksiyonu")

        self.name = name
        self.type = type

```

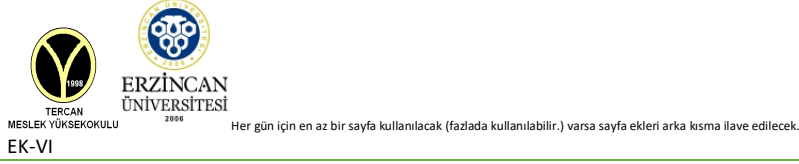
```
class Teacher(People):
    turke *
    def __init__(self, name, lesson_type: LessonType):
        super().__init__(name, PeopleType.TEACHER)
        print("Teacher sınıfının init fonksiyonu")
        self.lesson_type = lesson_type
        self.lesson_list = []
```

```
class Student(People):
    turke
    def __init__(self, name, number: int):
        super().__init__(name, PeopleType.STUDENT)
        self.lesson_list = []
        self.number = number
```

```
class School:
    turke
    def __init__(self, name: str):
        self.name = name
        self.admin = None
        self.student_list = []
        self.teacher_list = []
        self.room_list = []
        self.lesson_list = []
        self.admin_list = []
```

Tarih	İşin Adı Ne Amaçla ve Nereye yapıldığı	İş Yeri Yetkilisi İmza	Sayfa No

ERZİNCAN BİNALİ YILDIRIM ÜNİVERSİTESİ
TERCAN MESLEK YÜKSEKOKULU



9.Gün bugün okul projemize sınıflar oluşturmayı öğrendik Type ile lessonlar belirlendi

lessonlara field eklendi. lessona teacher ekleme kodu yazdım bunu yaparken lessonlarad teacher atamayı öğrendim.derslere öğrenciyi atamayı yaptık.yaptıklarımı git hub' a pushladım. ör:school_list[0] yaparak schoolun lessonlarından 0. olanı çekmeyi gördüm.

pythonda reference,memory location araştırıldı.

school sistemimizde artık her lessonun bir roomu ve cod'u var, teacher ların roomları ve lessonları atandı.room liste ise her rooma bir ders ve sınıfların öğretmenleri var artık.

Lesson type :

```
class LesseonType(enum.Enum):  
    MATH = 1  
    FIZIK = 2  
    KIMYA = 3
```

yeni ders ekleme fonksiyonu ve classmethodların oluşturulması :

```

def add_lesson(self, lesson_type: LesseonType, code: str):
    """
    :param lesson_type:
    :param code:
    :return:
    """
    print("\nadding lesson..")
    lessonObj = Lesson(lesson_type, code)
    self.lesson_list.append(lessonObj)
    print("successful..")

    @classmethod
    def add_lesson_to_room(cls, lessonObj: Lesson, roomObj: Room):
        """
        Bu kod Lessona Room ekler aynı zamanda Roomun lessonlistesine lesson ekler.
        :param lessonObj: eklenecek ders
        :param roomObj: eklenecek sınıf
        :return:
        """
        roomObj.lesson_list.append(lessonObj)
        lessonObj.lessonRoom = roomObj

    @classmethod
    def add_lesson_to_teacher(cls, lessonObj: Lesson, teacherObj: Teacher):
        lessonObj.teacher_list.append(teacherObj)
        teacherObj.teacherLesson = lessonObj

```

methodların çağırılması :

```

School.add_lesson_to_room(schoolObj.lesson_list[0], schoolObj.room_list[0])#classmethod çağırısı yaptık
School.add_lesson_to_room(schoolObj.lesson_list[1], schoolObj.room_list[1])
School.add_lesson_to_room(schoolObj.lesson_list[2], schoolObj.room_list[2])
School.add_lesson_to_teacher(schoolObj.lesson_list[2], schoolObj.teacher_list[0])
School.add_lesson_to_teacher(schoolObj.lesson_list[0], schoolObj.teacher_list[1])

```

ders ve sınıfların eklenmesi :

```

schoolObj.add_admin("EMRE")
schoolObj.add_room("PC1")
schoolObj.add_room("PC2")
schoolObj.add_room("PC3")
schoolObj.add_lesson(LesseonType.FIZIK,"FZ101")
schoolObj.add_lesson(LesseonType.KIMYA,"KM102")
schoolObj.add_lesson(LesseonType.MATH,"MT103")

```

Tarih	İşin Adı Ne Amaçla ve Nereye yapıldığı	İş Yeri Yetkilisi İmza	Sayfa No

ERZİNCAN BİNALİ YILDIRIM ÜNİVERSİTESİ
TERCAN MESLEK YÜKSEKOKULU



Her gün için en az bir sayfa kullanılacak (fazlada kullanılabilir.) varsa sayfa ekleri arka kısma ilave edilecek.

10.Gün bugün yaptığımız okul projesini menüler olarak tasarlamaya ayrı ayrı yeniden yazmaya başladık. menu içinde öğrenciler sınıflara yerleştirildi 3 öğrenci olarak eklendi.

ana menü tasarlandı ana menüde bütün bilgileri gösteren yeni fonksiyonlar yazıldı bütün classlar için student menüsünde dışarıdan student ekleme ,derse öğrenci ekleme,öğretmen menüsünde ise öğretmen ekleme kodları yazıldı.

Type olarak menüye ekleme fonksiyonu :

```
def student_type(text: str) -> Student:
    comingText = input(text).lower()
    if comingText == "0":
        return schoolObj.student_list[0]
    if comingText == "1":
        return schoolObj.student_list[1]
    if comingText == "2":
        return schoolObj.student_list[2]
    if comingText == "3":
        return schoolObj.student_list[3]
```

menü kodunda kullanımı :

```
if cevap == 4:
    print(schoolObj.list_writer("student"))
    schoolObj.add_student_to_lesson(schoolObj.student_list[int(input())],schoolObj.lesson_list[int(input())])
    print(schoolObj.student_list)
```

Menüde sadece istenilen bilgiyi yazdıran fonksiyonun oluşturumu :

```

def list_writer(self,value:str):
    if(value == "student"):
        for s in self.student_list:
            s.writer()

    elif(value == "teacher"):
        for t in self.teacher_list:
            t.writer()
    elif(value == "room"):
        for r in self.room_list:
            r.writer()
    elif(value == "lesson"):
        for l in self.lesson_list:
            l.writer()

```

menüde çağırılması ve kullanımı:

```

while True:
    cevap = Menu_main()

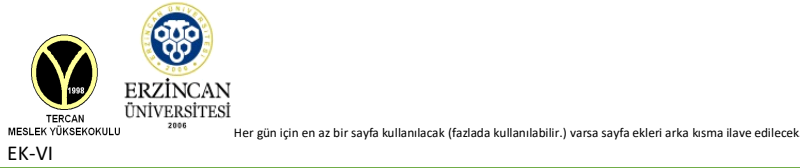
    if cevap == 1:
        schoolObj.list_writer("student")
    elif cevap == 2:
        schoolObj.list_writer("teacher")
    elif cevap == 3:
        schoolObj.list_writer("room")
    elif cevap == 4:
        schoolObj.list_writer("lesson")
    elif cevap == 0:
        break

```

schoolObj.list_writer("") -> diyerek schoolObj objesinden list_writer fonksiyonu çağırılıyor

Tarih	İşin Adı Ne Amaçla ve Nereye yapıldığı	İş Yeri Yetkilisi İmza	Sayfa No

ERZİNCAN BİNALİ YILDIRIM ÜNİVERSİTESİ
TERCAN MESLEK YÜKSEKOKULU



11.Gün başlangıcında yeni arkadaşla tanıştık ona giti ve pythonun başlangıcını anlattık yardımcı oldum.
method almayı öğrendim.

method alma : CTRL + ALT + M

teacher type ve add_student_to_lesson a try execpt ekleyerek döngüye aldım artık yanlış girilen değer var ise exept sayesinde ekrana hata printi yazdırıyoruz. try ile doğru seçeneklerin alımını yapıyoruz. okul sisteminde öğrenci silme fonksiyonu yazıldı. okul sisteminde öğrenci arama sistemi yazıldı . Öğrenci aramada öğrencinin adından harfleri yazarak filtreleme sistemi yapıldı.öğrenciden ders çıkarma fonksiyonu yazıldı. öğretmen sisteminde öğretmen silme yapıldı.teacher menüye öğretmen ekle yapıldı.

add student to lesson :

```
def add_student_to_lesson_fromterminal():
    while True:
        try:
            printList("student",_schoolObj.student_list)
            studentNumber = int(input("student number yazınız :"))
            printList("lesson",_schoolObj.lesson_list)
            lessonNumber = int(input("lesson number yazınız :"))
            schoolObj.add_student_to_lesson(schoolObj.student_list[studentNumber], schoolObj.lesson_list[lessonNumber])
            printPurple("Başarıyla eklendi.")

            break
        except:
            printError("listedeki numara girilmeli")
```

teacher type :

```
def teacher_type(text: str) -> LessonType:
    while True:
        try:
            comingText = input(text).lower()
            if comingText == "f" or comingText == "FİZİK":
                return LessonType.FİZİK
            elif comingText == "k" or comingText == "KİMYA":
                return LessonType.KİMYA
            elif comingText == "m" or comingText == "MATH":
                return LessonType.MATH
            elif comingText == "i" or comingText == "İNG":
                return LessonType.İNG
            elif comingText == "t" or comingText == "TÜRKÇE":
                return LessonType.TÜRKÇE
            elif comingText == "ta" or comingText == "TARİH":
                return LessonType.TARİH
            else:
                printError("FİZİK , KİMYA, MATH, İNG, TÜRKÇE, TARİH OLMAK ZORUNDA")
        except:
            printError("Girdiğiniz değer f,k,m,i,t,ta FİZİK,KİMYA,MAT,İNG,TÜRKÇE,TARİH olmak zorunda.")
```

student filtresi :

```
def writer(self):
    txt = "ogrenci: {ogrenci}"
    print(txt.format(ogrenci=self.name))
```

Tarih	İşin Adı Ne Amaçla ve Nereye yapıldığı	İş Yeri Yetkilisi İmza	Sayfa No

ERZİNCAN BİNALİ YILDIRIM ÜNİVERSİTESİ
TERCAN MESLEK YÜKSEKOKULU



MESLEK YÜKSEKOKULU
EK-VI

Her gün için en az bir sayfa kullanılacak (fazlada kullanılabilir.) varsa sayfa ekleri arka kısma ilave edilecek.

12.Gün öğretmenden ders çıkarma fonksiyonu yazıldı sisteme.yeni dersler eklendi. aynı dersi farklı kodlarla belirtip filtreleme yapcaz. Lesson daki writer fonksiyonu düzenlendi artık dersleri yazdırırken kodunu da yazdırıyor.ders tipine göre ekrana dersler geliyor numara yazarak dersleri filtreleme fonksiyonu yazıldı.

teacherdan lesson çıkarma :

```
def dell_teacher_to_lesson_fromterminal():  
    while True:  
        try:  
            printList("teacher",schoolObj.teacher_list)  
            teacherNumber = int(input("teacher number yazınız :"))  
            printList("lesson",schoolObj.lesson_list)  
            lessonNumber = int(input("lesson number giriniz :"))  
            schoolObj.del_lesson_to_teacher(schoolObj.teacher_list[teacherNumber], schoolObj.lesson_list[lessonNumber])  
            print(schoolObj.teacher_list)  
            printPurple("Başarıyla silindi.")  
            break  
        except:  
            printError("listeden numara girilmeli.")
```

lesson writer :

```
def writer(self):  
    txt = "ders: {ders} code: {code}"  
    print(txt.format(ders=self.lesson_type.name, code=self.code))
```

Tarih	İşin Adı Ne Amaçla ve Nereye yapıldığı	İş Yeri Yetkilisi İmza	Sayfa No

ERZİNCAN BİNALİ YILDIRIM ÜNİVERSİTESİ
TERCAN MESLEK YÜKSEKOKULU



MESLEK YÜKSEKOKULU
EK-VI

Her gün için en az bir sayfa kullanılacak (fazlada kullanılabilir.) varsa sayfa ekleri arka kısma ilave edilecek.

13.Gün başlangıcı bugün sabahtan okul sistemimizde çalışmaya başladık. teacher type fonksiyonumuza sonradan eklenen dersleri koda ekledim. Artık öğretmen eklerken diğer dersleride rahatça gösteren printerımız var. printList adında yeni fonksiyon yazdık artık listeleri filtreleme yaparken sadece istenilen bilgiyi veriyor. debug kullanarak kodumuzdaki hataları tespit etmeyi ne eksik var bulmayı öğrendim. Debug -> F8 tuşu ile ilerleme yaptırarak kodun ne ne yaptığını öğrenebiliyoruz. öğrenci ve öğretmen listesinin filtrelerini yeni yaptığımız fonksiyona göre ayarladık. projemizin son halini githuba attık. git hub da projenizde yapılan değişiklikler nelerdir görebiliyoruz.

```
def teacher_type(text: str) -> LessonType:
    while True:
        try:
            comingText = input(text).lower()
            if comingText == "f" or comingText == "FİZİK":
                return LessonType.FİZİK
            elif comingText == "k" or comingText == "KİMYA":
                return LessonType.KİMYA
            elif comingText == "m" or comingText == "MATH":
                return LessonType.MATH
            elif comingText == "i" or comingText == "İNG":
                return LessonType.İNG
            elif comingText == "t" or comingText == "TÜRKÇE":
                return LessonType.TÜRKÇE
            elif comingText == "ta" or comingText == "TARİH":
                return LessonType.TARİH
            else:
                printError("FİZİK , KİMYA, MATH, İNG, TÜRKÇE, TARİH OLMAK ZORUNDA")
        except:
            printError("Girdiğiniz değer f,k,m,i,t,ta FİZİK,KİMYA,MAT,İNG,TÜRKÇE,TARİH olmak zorunda.")
```

Tarih	İşin Adı Ne Amaçla ve Nereye yapıldığı	İş Yeri Yetkilisi İmza	Sayfa No

ERZİNCAN BİNALİ YILDIRIM ÜNİVERSİTESİ
TERCAN MESLEK YÜKSEKOKULU



Her gün için en az bir sayfa kullanılacak (fazlada kullanılabilir.) varsa sayfa ekleri arka kısma ilave edilecek.

EK-VI

14.Gün Reader Writer tasarlamaya geçiyoruz. Json modülü kullanımını öğrenmeye başladık.

Json modülü nedir ?

json Modülü

JSON farklı diller arasında iletişimi sağlamak için geliştirilmiş basit bir veri formatıdır. Bu veri formatı Python'daki sözlük ve listelere çok benzer. İki tür kullanımı vardır. Birincisi anahtar-değer mantıyla çalışır.

```
{  
  "Ad": "Fırat",  
  "Soyad": "Özgül"  
}
```

JSON ifadelerinde her öğe arasında virgül olmalıdır. Anahtar ve değer ikilisi arasında ise iki nokta kullanılır.

İkinci kullanımı ise liste tipine çok benzer. Bu kullanımda anahtar-değer değil, sadece değer verilir.

```
{  
  "Fırat",  
  "Özgül"  
}
```

Json modülü öğrenci sistemi için yapıldı. Artık classlardan tüm listeye ulaşmak yerine diğerlerinden çağırma fonksiyonu yazıldı. Arayüz tasarlama, programlama için araştırmalara başlandı.

Teacher, lesson, room 'a json modülü uyarlandı. Write To File fonksiyonu yazıldı. Sıkıntısı şekilde artık veriler json dosyamıza kayıt olmaktadır.

Tarih	İşin Adı Ne Amaçla ve Nereye yapıldığı	İş Yeri Yetkilisi İmza	Sayfa No

ERZİNCAN BİNALİ YILDIRIM ÜNİVERSİTESİ
TERCAN MESLEK YÜKSEKOKULU



Her gün için en az bir sayfa kullanılacak (fazlada kullanılabilir.) varsa sayfa ekleri arka kısma ilave edilecek.

EK-VI

15.Gün Arayüz programlamaya geçildi. Label oluşturma öğrenildi. Button kullanımı ve fonksiyon atama öğrenildi. Canvas oluşturma öğrenildi. Canvas bizim ekranımız neyin nereye konulacağı yer. pack ile programın çalışmasını sağlayan kodu öğrendim. Arka plan renk oluşturma ve düzenleme öğrenildi.

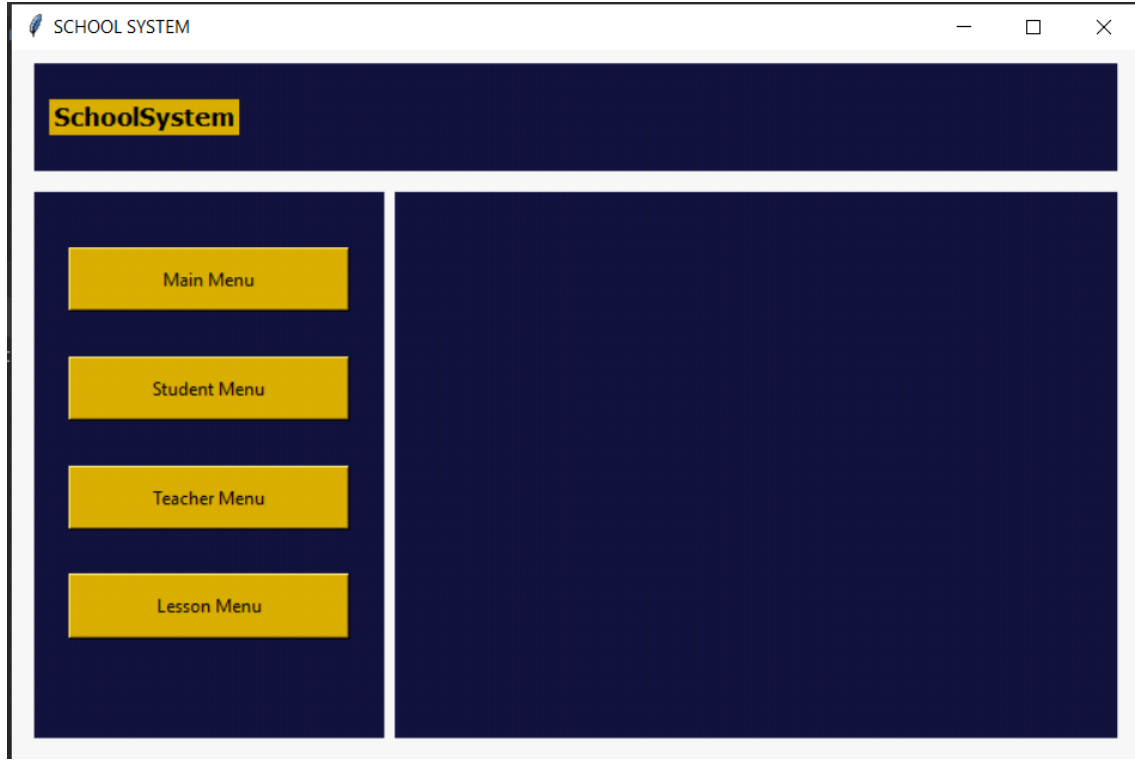
Arayüz oluşturuldu. Örnek bir canvas tasarlandı program dışındı örnek için. projelere başlamadan herşeyi planlamak ve düzenlemek gerekir yoksa projelerde yazılımcılar yanlışlıklar yapabiliyor ve sonradan sorunlar çıkabiliyor. Buttona basınca konsola yazdıran kod yazıldı. Buttona basınca uyarı ekranı veren kod yazıldı. button ile text değiştirme öğrenildi. buttona basınca arka plan rengi değiştiren kod yazıldı.

buttonun üzerine gelince rengi değiştirme yapıldı. Proje ekranı büyültme küçültme engellemek için resizable öğrenildi.

örnek label :

```
label1 = tk.Label(master, text='Öğrenci Giriniz : ')\nlabel1.config(font=('helvetica', 10))\ncanvas1.create_window(100, 75, window=label1)
```

ilk örnek arayüzümüz :



Tarih	İşin Adı Ne Amaçla ve Nereye yapıldığı	İş Yeri Yetkilisi İmza	Sayfa No

ERZİNCAN BİNALİ YILDIRIM ÜNİVERSİTESİ
TERCAN MESLEK YÜKSEKOKULU



Her gün için en az bir sayfa kullanılacak (fazlada kullanılabılır.) varsa sayfa ekleri arka kısma ilave edilecek.

EK-VI

16.Gün Arayüz Programlama

School systemimizin arayüz buttonlarını kendi kendine döngü ile oluşturan fonksiyonu yazdık. Sub menu olan orta kısma scroll bar yapma fonksiyonu yazdık. Main, student, teacher, lesson, menülerinin sub menüleri okutulup oluşturuldu. sub menülere nasıl fonksiyon çağrılır öğrenildi. Sub menülere printer sistemi yapıldı. Arayüzü büyültme küçültme her bilgisayarda aynı olması için engelledik. Arayüzü oluşturan frameleri fonksiyon haline getirdik. Creat menu olarak ana menüleri oluşturan fonksiyonu yazdık. Change menu adında menüler arası gezmeye yarayan fonksiyonu yazdık. Fonksiyonları gerekli yerlerde çağırıp çalışmasını sağladık. Teacher ve lessona button fonksiyonları eklendi. Menülerin ve sub menülerin ismi düzenlendi.

for döngüsünde buttonları otomatik oluşturan kod :

```
for i, text in enumerate(subMenuMain):
    Button1 = Button(frame_buttons, text=text, width=20, height=2, bg='#B5C6B8', activebackground="#eee8aa")
    Button1.grid(row=i, column=0, padx=5, pady=5, sticky='nwes')
frame_buttons.update_idletasks()
frame_canvas.config(width=200 + vsb.winfo_width(), height=SCREEN_HEIGHT)
canvas.config(scrollregion=canvas.bbox("all"))
```

sub menu oluřturucu :

```
def submenuCreator(menu_name, subMenuItems=[]):
    frame_canvas = Frame(frame_alt_sag)
    submenuFrames[menu_name] = frame_canvas
    frame_canvas.grid(row=0, column=0, pady=(5, 0), sticky='nw')
    frame_canvas.grid_rowconfigure(0, weight=1)
    frame_canvas.grid_columnconfigure(0, weight=1)
    # Set grid_propagate to False to allow 5-by-5 buttons resizing later
    frame_canvas.grid_propagate(False)

    canvas = Canvas(frame_canvas, bg="#8FA2CA")
    canvas.grid(row=0, column=0, sticky="news")
    vsb = Scrollbar(frame_alt_sag, orient="vertical", command=canvas.yview)
    vsb.grid(row=0, column=1, sticky='ns')
    canvas.configure(yscrollcommand=vsb.set)
    frame_buttons = Frame(canvas, bg="blue")
    canvas.create_window((0, 0), window=frame_buttons, anchor='nw')
    for i, obj in enumerate(subMenuItems):
        Button1 = Button(frame_buttons, text=obj["text"], command=obj["func"], width=20, height=2, bg='#B5C6B8', activebackground="#eee8aa")
        Button1.grid(row=i, column=0, padx=5, pady=5, sticky='news')
    frame_buttons.update_idletasks()
    frame_canvas.config(width=200 + vsb.winfo_width(), height=SCREEN_HEIGHT)
    canvas.config(scrollregion=canvas.bbox("@all"))
```

Tarih	İřin Adı Ne Amaçla ve Nereye yapıldığı	İř Yeri Yetkilisi İmza	Sayfa No

ERZİNCAN BİNALİ YILDIRIM ÜNİVERSİTESİ
TERCAN MESLEK YÜKSEKOKULU



Her gün için en az bir sayfa kullanılacak (fazlada kullanılabilir.) varsa sayfa ekleri arka kısma ilave edilecek.

17.Gün Menülerin düzenlenmesi

Artık menüler yeni pencerede açılacak şekilde yeniden tasarlandı. yeniden proje planlaması yapıldı.

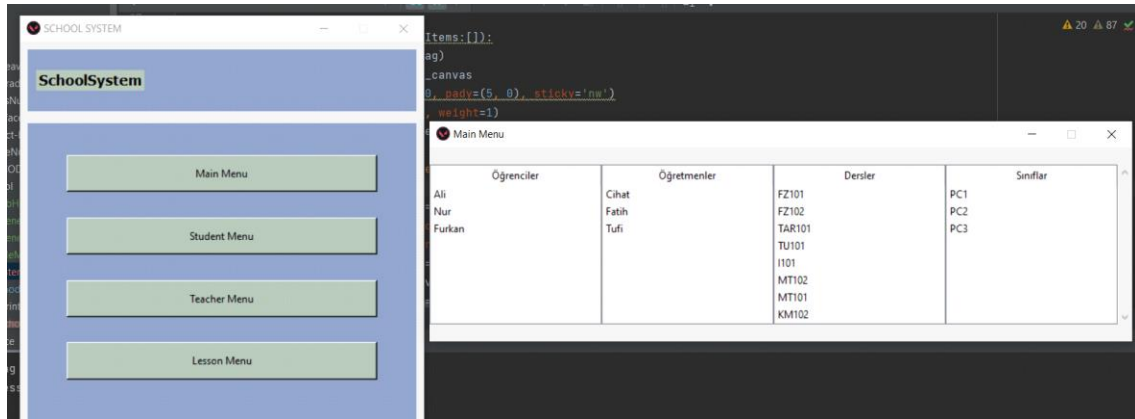
Yeni pencereyi liste şeklinde yapma ve butonlara fonksiyon kullanılması öğrenildi.

Her menü için yeni pencere oluşturma yapıldı.

Pencerelere butonlar koyulup fonksiyonlar yazıldı.

Student windowunun butonlarına öğrenci ekleme ve silme kodları yazıldı. gün tekrar edilerek bitirildi

Arayüz :



Student window :



Main Menu



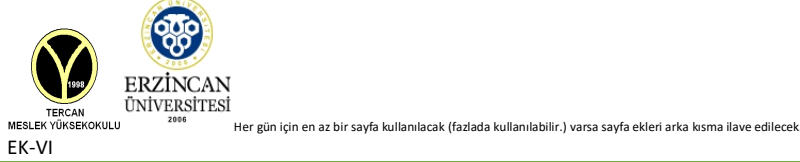
Adı	Numarası	Ders Sayısı
Ali	870	1
Nur	23	2
Furkan	691	1

Öğrenci ekle

Öğrenciyi sil

Tarih	İşin Adı Ne Amaçla ve Nereye yapıldığı	İş Yeri Yetkilisi İmza	Sayfa No

ERZİNCAN BİNALİ YILDIRIM ÜNİVERSİTESİ
TERCAN MESLEK YÜKSEKOKULU



18.Gün Projenin düzenlenmesi

Projeye tkintera .ico uzantılı resim koyma öğrenildi ve projenin logosu değiştirildi.

hata olan import edilmemiş fonksiyonu import etme kısayolu ALT + ENTER

School systemimizde yazdırmalar düzenlenirken hatalar çıktı ve kod hata ayıklama modunda hataları aradık.

Hataların çözülmesi 2 saat sürdü ve arayüz düzenlemesine devam edildi ve düzenlendi. Arayüz daha kullanılır hale getirildi. Araştırarak listede ekli olan birinin mouse ile seçerek silme kodu yapıldı.

School system.py dosyamızdan öğrenciler ve öğretmenleri başladığımız arayüz programına aktarmasını yaptık.

Arayüz olarak lessonada sadece ders sınıf yazdırılıyor. student button fonksiyonu tekrar düzenlendi.

Teachera silme buttonu eklendi ve fonksiyonu yazıldı. Günü tekrar ederek bitirdim.

School system in bağlanması :

```

def init():
    schoolObj = School("Ebyu")
    adminObj = Admin("Emre")
    schoolObj.admin = adminObj
    schoolObj.add_teacher("Cihat", LessonType.MATH)
    schoolObj.add_teacher("Tufi", LessonType.FIZIK)
    schoolObj.add_teacher("Fatih", LessonType.KIMYA)
    schoolObj.add_student("Furkan")
    schoolObj.add_student("Nur")
    schoolObj.add_student("Ali")
    schoolObj.add_admin("EMRE")
    schoolObj.add_room("PC1")
    schoolObj.add_room("PC2")
    schoolObj.add_room("PC3")
    schoolObj.add_lesson(LessonType.FIZIK, "FZ101")
    schoolObj.add_lesson(LessonType.FIZIK, "FZ102")
    schoolObj.add_lesson(LessonType.KIMYA, "KM101")
    schoolObj.add_lesson(LessonType.KIMYA, "KM102")
    schoolObj.add_lesson(LessonType.MATH, "MT101")
    schoolObj.add_lesson(LessonType.MATH, "MT102")
    schoolObj.add_lesson(LessonType.ING, "I101")
    schoolObj.add_lesson(LessonType.TURKCE, "TU101")
    schoolObj.add_lesson(LessonType.TARIH, "TAR101")
    School.add_lesson_to_room(schoolObj.lesson_list[0], schoolObj.room_list[0]) # cla
    School.add_lesson_to_room(schoolObj.lesson_list[1], schoolObj.room_list[1])
    School.add_lesson_to_room(schoolObj.lesson_list[2], schoolObj.room_list[2])
    School.add_lesson_to_teacher(schoolObj.lesson_list[0], schoolObj.teacher_list[1])
    School.add_lesson_to_teacher(schoolObj.lesson_list[1], schoolObj.teacher_list[2])
    School.add_lesson_to_teacher(schoolObj.lesson_list[2], schoolObj.teacher_list[0])

```

```

if __name__ == "__main__":
    global schoolObj
    schoolObj = init()
    createPanels()
    createMenu()
    master.mainloop()

```

Teacher arayüzü :

✓ Main Menu

Adı	Dersi	Ders Sayısı
Fatih	LessonType.KIMYA	1
Tufi	LessonType.FIZIK	1
Cihat	LessonType.MATH	1

Öğretmen ekle

Derse öğretmen ekle

Dersten öğretmen çıkar

Öğretmeni sil

Tarih	İşin Adı Ne Amaçla ve Nereye yapıldığı	İş Yeri Yetkilisi İmza	Sayfa No

ERZİNCAN BİNALİ YILDIRIM ÜNİVERSİTESİ
TERCAN MESLEK YÜKSEKOKULU



Her gün için en az bir sayfa kullanılacak (fazla kullanılabılır.) varsa sayfa ekleri arka kısma ilave edilecek.

EK-VI

19.Gün Arayüz Programlama Düzenlenmeler

Lessonun yazdırmaları düzenlendi. Teacher sekmesinin içinde artık ismi dersi ve ders sayısı yazıcak. Lesson menüye silme özelliği eklendi. MySQL e giriş yaptık. SQLAlchemy ile tablo oluşturma öğrenildi. Yeni bir SQL deneme dosyası oluşturuldu. Projeye sql import etme yapıldı.

import sqlalchemy -> import edilme şekli. pip install sqlalchemy pythona sqlalchemy i kurmak. Ders öğretmen tabloları oluşturuldu. tablolarda SQL ile değişiklik yapılması öğrenildi. Konular ile alakalı videolar izlendi. Diagram oluşturma öğrenildi. DrawSQL sitesi örnek diagram oluşturmak için projelerde kullanılmak için öğrenildi. örnek diagram oluşturuldu ve araştırıldı.

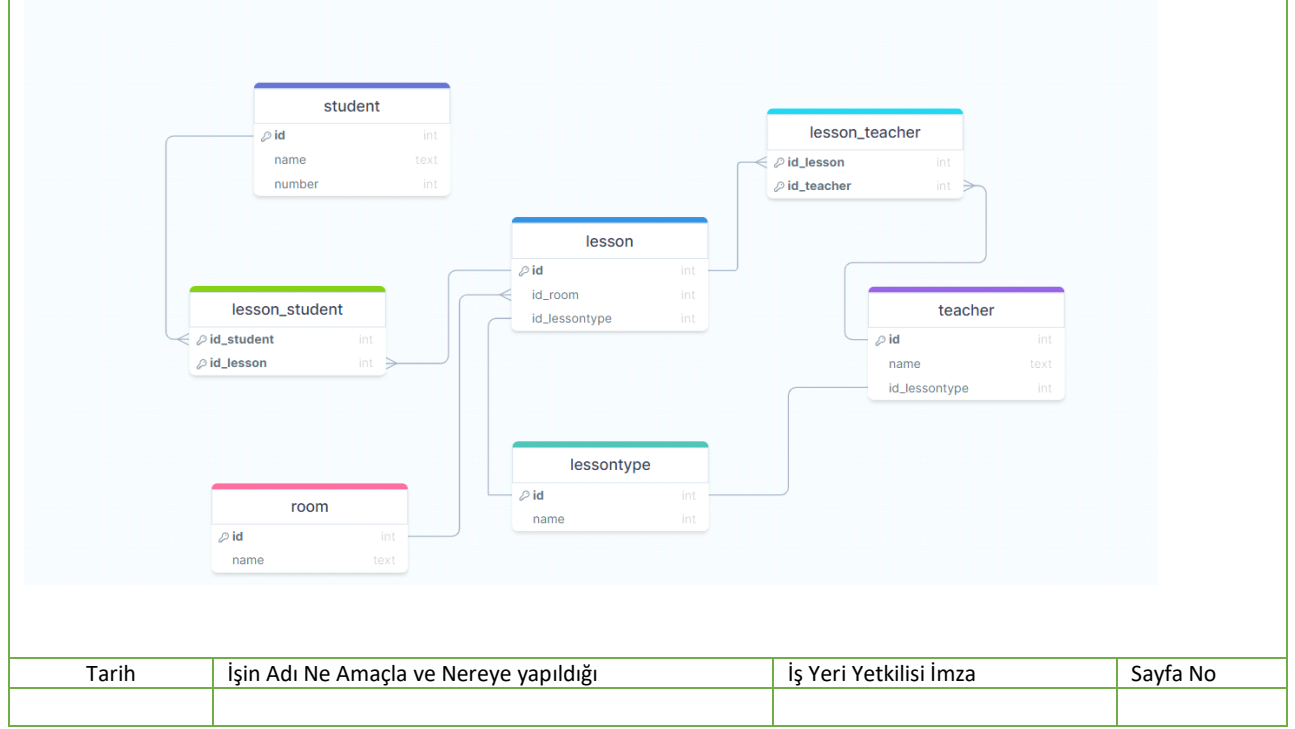
SQL column oluşturulumu :

```
class Lesson(Base):
    __tablename__ = "lesson"
    id = Column(Integer, primary_key=True)
    teachers = relationship("teacher", secondary=association_table, back_populates="lessons")
    children = relationship("room")
    id_room = Column(Integer, ForeignKey("room.id"))

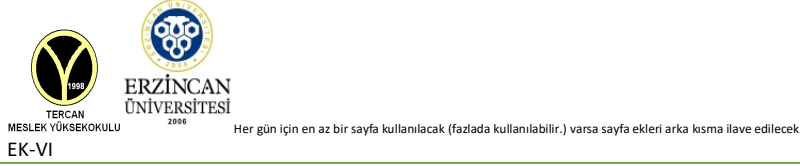
class Room(Base):
    __tablename__ = "room"
    id = Column(Integer, primary_key=True)
    parent_id = Column(Integer, ForeignKey("lesson.id"))

class Teacher(Base):
    __tablename__ = "teacher"
    id = Column(Integer, primary_key=True)
    lessons = relationship("lesson", secondary=association_table, back_populates="teachers")
```

Diagram :



ERZİNCAN BİNALİ YILDIRIM ÜNİVERSİTESİ
TERCAN MESLEK YÜKSEKOKULU



20. Gün Son günümde yeni basit projeler yapmaya başladık. Ufak bir sesli asistan yaparak projemizi başlattığımızda bize geri cevap veren gerekli google araması yaptırılan ve bilgisayarda istediğin programı konuşarak ismini örnek olarak blender ı başlat diyerek bilgisyarda kendi kendine uygulamayı açan sesli asistan yapıldı.

Pycharm da youtube videoları indirmek için kodlar öğrenildi ve youtube downloader yapıldı. Sonradan geliştirerek youtube da video listesi indirmeyi sağlayan bir program yaptık.

Pycharm da imagerleri pdf'e çeviren kodu yazdık. Python'da e-mail gönderme işlemini yapan kodu yazdık. Son günümüzü farklı projeler deneyip öğrenerek bitirdik.

İmage to pdf :

```

from fpdf import FPDF
from fpdf import FPDF

class PDF(FPDF):...

# Instantiation of inherited class

pdf = PDF()

list_of_images = ["1.jpg", "2.jpg"]

for i in list_of_images:
    pdf.add_page()
    pdf.image(i,x=50,y=100,w=100,h=100)
    for i in range(1, 2):
        pdf.cell(0, 10, 'Printing line number ' + str(i), 0, 1)
    pdf.output("yourfile.pdf", "F")

```

yt downloader :

```

import pytube

link = input('Enter Youtube Video Url')

if "list" in link:
    p = pytube.Playlist(link)
    for video in p.videos:
        video.streams.get_highest_resolution().download()
else:
    yt = pytube.YouTube(link)
    yt.streams.get_highest_resolution().download()

print('downloaded',link)

```


Send e-mail :

```
import smtplib

my_email = "turkemre1966@gmail.com"
apppassword = "oxxu1aghiemnosgo"

connection = smtplib.SMTP("smtp.gmail.com", 587)
connection.starttls()
connection.login(user=my_email, password=apppassword)

connection.sendmail(from_addr=my_email,
                    to_addrs="turkemre5234@gmail.com", msg="Merhaba Emre")

connection.close()
```

Bu staj bana yazılım dili öğrenirken nasıl çalışmam gerektiğini öğretti. Gelecekte yeni diller öğrenirken Öğrendiğim şekilde proje planlaması diagram oluşturulması gibi önemli konuları unutmayacağım.

Sorumlu

Mühendisten güzel geri bildirimler aldım ve elinin klavyeye değmeden sürekli deneyip kod okuyup yazmadan

Öğrenilemeyeceğini öğrendim. Python alanında kendimi geliştirmeye devam edicem.

Tarih	İşin Adı Ne Amaçla ve Nereye yapıldığı	İş Yeri Yetkilisi İmza	Sayfa No