



İSTANBUL AREL ÜNİVERSİTESİ
FİNAL PROJESİ



HAVAALANI OTOMASYON SİSTEMİ

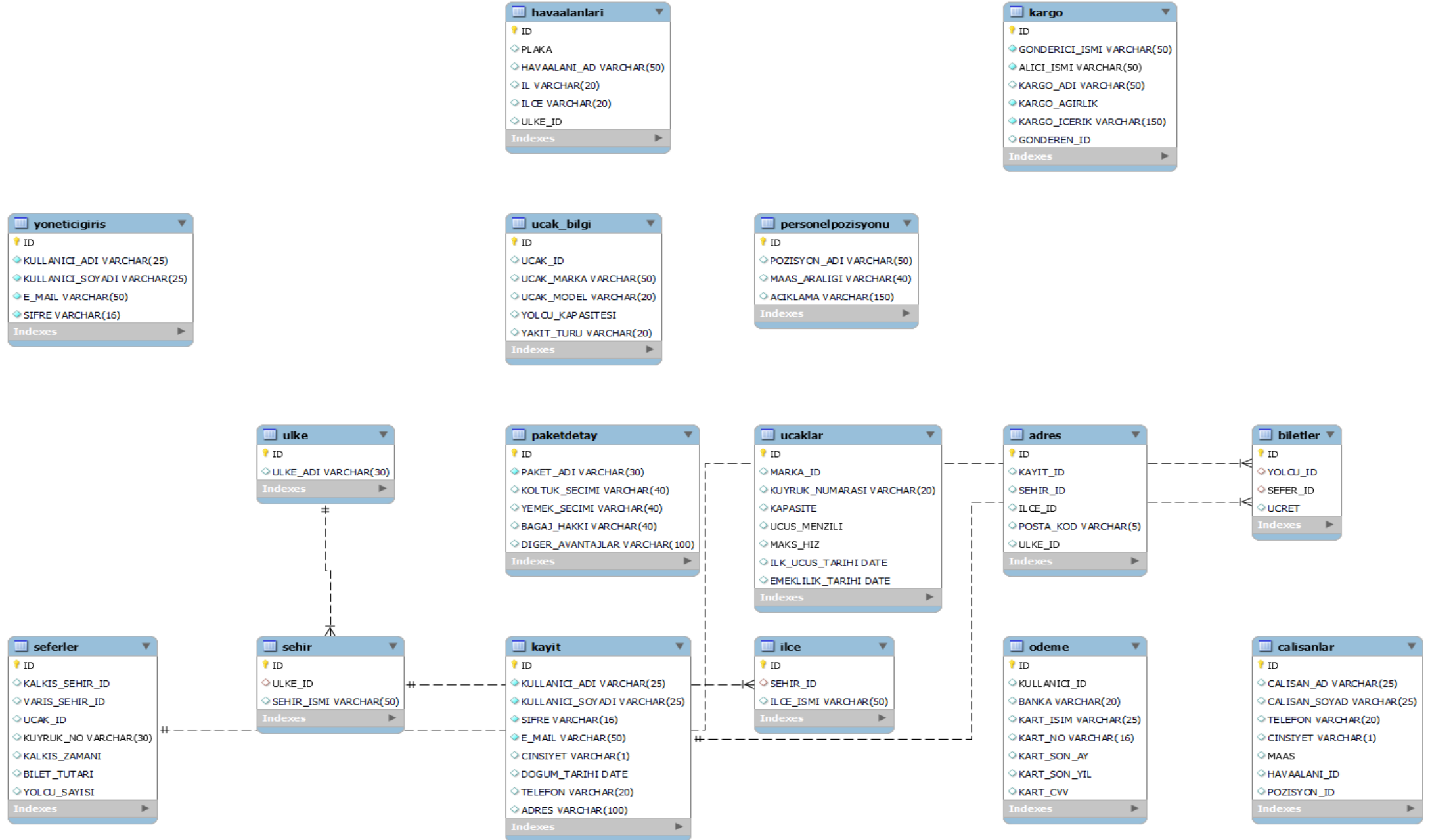


YAKUP UZUNOĞLU

210309008

BİLGİSAYAR MÜHENDİSLİĞİ(TÜRKÇE)

ER DİYAGRAMI



HAVAALANI OTOMASYON SİSTEMİ

Geliştirdiğim Havaalanı Otomasyon Sistemi, kullanıcı ve yönetici tarafında kapsamlı bir hizmet sunarak havaalanı operasyonlarını daha verimli hale getiriyor. Sistemin sunduğu önemli işlevsellikleri incelediğimizde hem kullanıcılar için hem de yöneticiler için temel gereksinimlerini karşılayacak işlemler bulunmaktadır.

Kullanıcı İşlemleri:

Seyahat Planlama ve Bilet Alma: Kullanıcılar, seyahatlerini planlamak ve bilet satın almak için kullanıcı dostu bir arayüz üzerinden hızlı ve kolay bir şekilde işlem yapabiliyor. Seferleri filtreleme, görme ve seçme özelliği sayesinde istedikleri uçuşları özelleştirebilirler. Ayrıca, bilet alım aşamasında ödeme bilgilerini sisteme kaydediliyor.

Uçakla Kargo Gönderimi: Sistem, kullanıcılara uçakla kargo gönderme imkânı da sunar. Kullanıcılardan gerekli bilgileri alıp sisteme kayıt eder.

Yönetici İşlemleri:

Sefer Yönetimi: Yöneticiler, sistemi kullanarak yeni seferler oluşturabilir ve mevcut seferleri görüntüleyebilir. Bu sayede havaalanı operasyonlarını planlamak ve düzenlememizi sağlar.

Çalışan Yönetimi: Sistem, havaalanına yeni katılan çalışanların bilgilerini eklemek ve mevcut çalışanları görüntülemek için yöneticilere özel sayfa oluşturuldu. Bu sayede personel yönetimi daha sistemli ve verimli bir şekilde gerçekleştirilir.

Uçak Yönetimi: Yeni alınan uçakların bilgilerini sisteme eklemek ve tüm uçakları görüntülemek yöneticilere kolaylık sağlar. Bu özellik, hava filosunu etkili bir şekilde yönetme imkanı sunar.

Veri Yönetimi:

Kod yazarken, alınan verileri minimum yer kaplaması için akıllı bir yaklaşım benimsedim. Örneğin, kullanıcıdan alınan il ve ilçe bilgilerini daha önce oluşturulan tablolar üzerinden sorgulayarak, bu bilgileri ID değerleri üzerinden bağladım. Böylece, aynı veriyi tekrar tekrar depolamak yerine, ilgili ID değerleri aracılığıyla bu bilgilere erişebildik. Bu veri yönetimi stratejisi, veritabanını daha etkin bir şekilde kullanmak adına tasarladım.

Sonuç:

Havaalanı Otomasyon Sistemi, modern bir havaalanı işletmesi için gerekli olan temel işlevselliği sağlayarak hem kullanıcılar hem de yöneticiler için bir deneyim sunar. Aşağıda oluşturduğum sayfalarda yaptığım işlemleri, sorguları anlattım.

DATABASE BAĞLAMA FONKSİYONU

database() fonksiyonunda mysql.connector.connect komutu ile host, user, password bilgilerini ve oluşturduğum database'in adını vererek gerekli bilgileri doldurdum ve bu parametrelerle eşleşen database bağlamış oldum.

MYSQL sorgusu çalıştırdığım her yerde database fonksiyonunu çağırdım. Global olarak mydb ve mycursor değişkenlerimi tanımlayarak başka fonksiyonlarda tekrar tekrar tanımlama ihtiyacımı ortadan kaldırdım.

```
def database():  
    global mydb  
    global mycursor  
    mydb = mysql.connector.connect(  
        host="localhost",  
        user="root",  
        password="1234",  
        database="havaalani"  
    )  
    mycursor = mydb.cursor()
```

CLEAR_WIDGETS FONKSİYONU

Tüm sayfaları tek bir py dosyasın da topladığım için (fonksiyonlar aracılığı ile) bir sayfadan başka bir sayfaya geçerken widgetleri temizliyor ve farklı sayfalardaki widgetler birbirinin üstüne binmemiş oluyor.

```
def clear_widgets(frame):  
    for widget in frame.winfo_children():  
        widget.destroy()
```

GİRİŞ YAP EKRANI

```
def login(): # Çıkış yap butonlarına basıldığında bu fonksiyona yönlendiriliyor ve uygulamanın karşılama ekranı
```

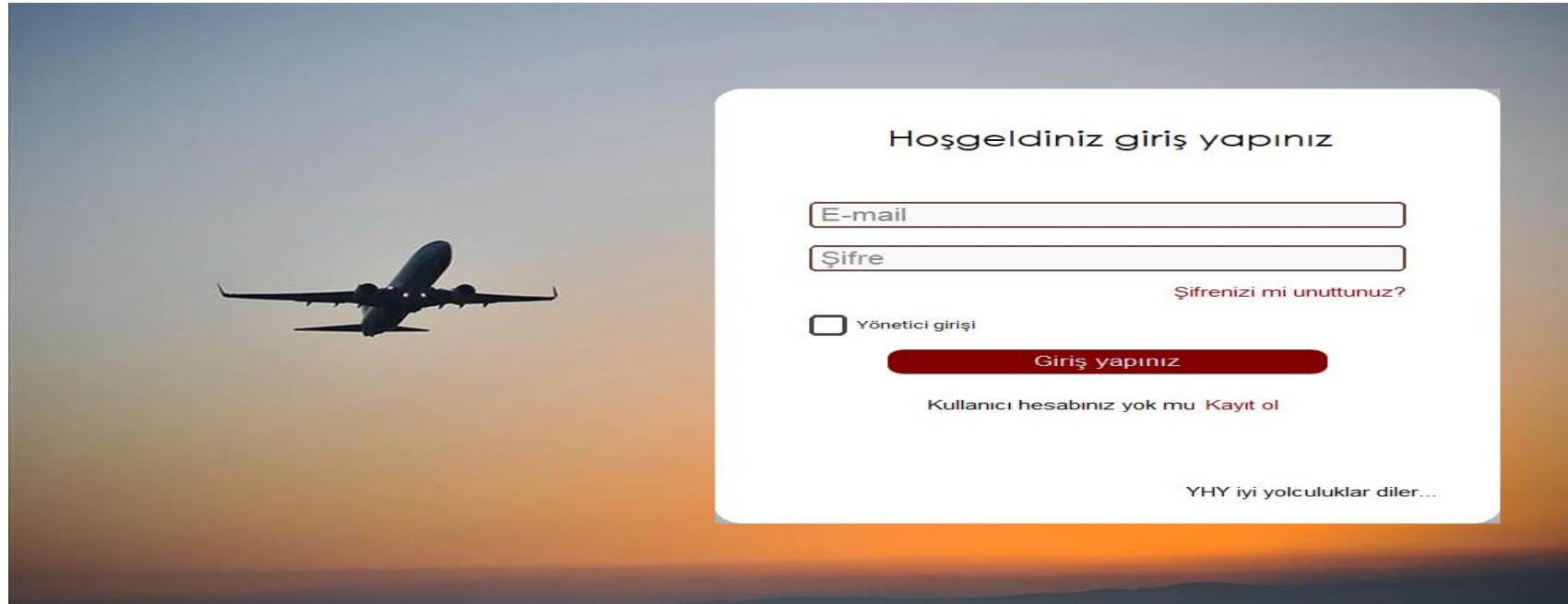
Giriş ekranımda 2 farklı kullanıcı tipi için giriş seçeneği sundum (kullanıcı ve yönetici girişi). Kullanıcıdan e-mail ve şifre aldım ve bu bilgileri sql sorgumda kullandım.

```
SELECT  
ID, KULLANICI_ADI, KULLANICI_SOYADI, E_MAIL, SIFRE  
FROM YONETICIGIRIS  
WHERE E_MAIL=%S AND SIFRE=%S", (E_MAIL, SIFRE))
```

Böyle bir kayıt varsa gerekli sayfaya yönlendirilecektir. Bu şartı sağlayan kayıt yoksa hata mesajı alacaktır.

YÖNETİCİ GİRİŞİ	
ID	INT AUTO_INCREMENT
E_MAIL	VARCHAR (50)
SIFRE	VARCHAR (16)
KULLANICI_ADI	VARCHAR (25)
KULLANICI_SOYADI	VARCHAR (25)

KAYIT	
ID (MP)	INT AUTO_INCREMENT
KULLANICI_ADI	VARCHAR (25)
KULLANICI_SOYADI	VARCHAR (25)
SIFRE	VARCHAR (16)
E_MAIL	VARCHAR (50)
CINSİYET	VARCHAR (1)
DOĞUM_TARIHI	DATE
TELEFON	VARCHAR (20)



Hoşgeldiniz giriş yapınız

E-mail

Şifre

Şifrenizi mi unuttunuz?

☐ Yönetici girişi

Giriş yapınız

Kullanıcı hesabınız yok mu [Kayıt ol](#)

YHY iyi yolculuklar diler...

SAYFALARA YÖNLENDİRME İŞLEMİ

Yönetici girişi yapmak için checkbox oluşturdum ve üzerine tıklandığı zaman check_var değerim “on” oluyor .

```
check_var = customtkinter.StringVar(value="off")
checkbox = customtkinter.CTkCheckBox(master=frame, text="Yönetici girişi",variable=check_var,
onvalue="on", offvalue="off",fg_color="#820000",hover_color="#673d35")
checkbox.place(x=60,y=260)
```

check_var değişkenini get() fonksiyonu ile değerini aldım ve yönetici değerine atıyorum ve if koşulu ile yönetici butonum aktif edilmiş mi diye kontrol ettiriyorum eğer basıldı ise sefer oluşturma fonksiyonuna(ekranına) gönderiliyor.

Eğer basılmadı ise bilet fonksiyonuna yani kullanıcı ekranına yollanıyor.

```
yonetici=check_var.get()
if yonetici=='on':
    . . . . .
    sefer_()
else:
    . . . . .
    bilet()
```

KAYIT OL EKRANI


Kullanıcının giriş yapabilmesi için ad, soyad, şifre, mail, telefon, cinsiyet, il, ilçe, posta kodu gibi parametreleri alıp kayıt tablosuna ekledim.

```
SELECT ID FROM KAYIT WHERE E_MAIL=%S
```

2.adım olarak oluşturduğum kaydın ID sini çekerek adres tablosundaki KAYIT_ID kolonuna ekledim ve geri kalan değerleri insert ederek adres tablomu doldurdum

```
insert1 = "INSERT INTO adres  
(KAYIT_ID,SEHIR_ID,ILCE_ID,POSTA_KOD,ULKE_ID) VALUES (%s, %s,%s,%s,%s)"  
degerler1 = (id,Sehir_id,Ilce_id,posta_kod,1)  
mycursor.execute(insert1, degerler1)  
mydb.commit()
```

ADRES	
ID	INT AUTO INCREMENT
KAYIT_ID (F)	INT
SEHIR_ID (F)	INT
ILCE_ID (F)	INT
POSTA_KOD	VARCHAR(5)
ULKE_ID (F)	INT

 Hoş geldiniz

Giriş yap

Hoşgeldiniz giriş yapınız

Adınız :

Soyadınız :

Şifre :

Mail :

Telefon :

Doğum tarihiniz :

Cinsiyet : K

İl :

İlçe :

Posta kodu :

Kayıt ekle


SEFER OLUŞTURMA EKRANI

Yönetici hesabı ile giriş yaptığımızda bizi karşılayan ilk ekran uçakların kalkışı için sefer oluşturma ekranıdır. Kullanıcıdan aldığım uçak kalkış ve varış şehirlerini where şartı ile şehirlerin ID değerlerini çekip seferler tabloma kayıt ekliyorum.

```
select="SELECT ID FROM sehir WHERE SEHIR_ISMI=%s"
mycursor.execute(select,(kalkis_nokta,))
Kalkis = mycursor.fetchone()
mycursor.execute(select,(varis_nokta,))
Varis=mycursor.fetchone()
```

SEFERLER	
ID (P)	INT AUTO INCREMENT
KALKIS_SEHIR_ID (F)	INT
VARIS_SEHIR_ID (F)	INT
UCAK_ID (F)	INT
KALKIS_ZAMANI	DATETIME
BILET_TUTARI	INT
KUYRUK_NO	VARCHAR (30)

Oluşturduğum kayıtları treeview widget'i ile listeletiyorum ve kayıt silme işlemi yapıyorum



yakup uzunoglu

Sefer ekle

Uçak ekle

Paket ekle

Çalışan ekle

Çıkış yap

SEFER OLUŞTUR

Kalkış noktası :

Varış noktası :

Yolcu Sayısı :

Bilet fiyatı :

Kalkış zamanı:

Kuyruk no:

Sefer ekle

Verileri Listele				Kayıt Sil			
ID	KALKIS_SEHIR_ID	VARIS_SEHIR_ID	KUYRUK_NO	KALKIS_ZAMANI	BILET_TUTARI	YOLCU_SAYISI	
2	34	16	x78	2023-12-25 12:30:	1200	250	
3	34	61	x96y	2023-12-25 14:00:	2000	250	
4	34	61	x96y	2023-12-25 23:00:	2000	250	
5	34	36	x96y	2023-12-25 23:00:	2000	250	
7	34	45	x96y	2023-12-25 23:00:	2000	250	
8	34	3	x96y	2023-12-25 23:00:	2000	250	
9	34	63	x96y	2023-12-25 23:00:	2000	250	
10	34	66	x96y	2023-12-25 23:00:	2000	250	
11	34	33	x96y	2023-12-25 23:00:	2000	250	
12	34	7	x96y	2023-12-25 23:00:	2000	250	
13	34	38	x96y	2023-12-25 23:00:	2000	250	
14	34	38	x96y	2023-12-25 23:00:	2000	250	
15	34	38	x96y	2023-12-25 23:00:	2000	250	

ÇALIŞAN EKLEME EKRANI

Çalışan ekle ekranında pozisyon değeri hariç alınan bilgiler çalışanlar tablosuna ekleniyor. Pozisyon değeri ise ;

```
poz=pozisyon.get()
select="SELECT ID FROM personelpozisyonu WHERE POZISYON_ADI=%s"
mycursor.execute(select,(poz,))
poz_id= mycursor.fetchone()
```


bu şekilde personel pozisyon tablosundan pozisyonun ID sini çekip çalışanlar tablosuna ekledim. Havaalanı seç butonuna basıldığında havaalanının ID değerini aşağıdaki kod ile çekiyorum.

```
havaalani_id = tree.item(selected_item, 'values')[0]
```

Aldığım ID değerini çalışanlar tablosuna ekliyorum.

ÇALIŞANLAR	
ID (P)	INT AUTO_INCREMENT
CALISAN_AD	VARCHAR (25)
CALISAN_SOYAD	VARCHAR (25)
TELEFON	VARCHAR (20)
CINSIYET	VARCHAR (1)
MAAS	INT
HAVAALANI_ID	INT
POZISYON_ID	INT

PERSONELPOZISYONU	
ID	INT AUTO_INCREMENT
POZISYON_ADI	VARCHAR (50)
MAAS_ARALIGI	VARCHAR (40)
ACIKLAMA	VARCHAR (150)



yakup uzunoglu

Sefer ekle

Uçak ekle

Paket ekle

Çalışan ekle

Çıkış yap

CALISAN EKLE

Çalışan ad :

Çalışan soyad :

Telefon :

Cinsiyet :

Pozisyon :

Maaş :

Çalışan ekle

Havaalanlarını Listele		Havaalanı seç	
ID	HAVAALANI AD	İL	İLÇE
1	Atatürk Havalimanı	İstanbul	Bakırköy
2	Sabiha Gökçen Havalimanı	İstanbul	Pendik
3	Esenboğa Havalimanı	Ankara	Akyurt
4	Adnan Menderes Havalimanı	İzmir	Gaziemir
5	Antalya Havalimanı	Antalya	Muratpaşa
6	Şakirpaşa Havalimanı	Adana	Seyhan
7	Yenişehir Havalimanı	Bursa	Yenişehir
8	Diyarbakır Havalimanı	Diyarbakır	Yenişehir
9	Trabzon Havalimanı	Trabzon	Akçaabat
10	Oğuzeli Havalimanı	Gaziantep	Oğuzeli
11	Merzifon Havalimanı	Amasya	Merzifon
12	Çardak Havalimanı	Denizli	Çardak
13	Milas Bodrum Havalimanı	Muğla	Milas

BİLET ALMA EKRANI

Kullanıcı girişi yaptığımız da bizi karşılayan ilk ekran bilet alma ekranı bu ekranda bilet alma, sefer seçme, fiyat hesaplama ve seferleri filtreleme işlemi yapılmaktadır.

```
for row in verilerim:  
    ID=row[0]
```

Kullanıcı giriş yaptığinde tüm bilgilerini fetcholl komutu ile verilerim değişkenine atadım ve verilerimin 0.indisinde bulunan ID değerini çektim ve bilet alındığı zaman ekleyeceğim kayıttın YOLCU_ID değerine ekliyorum.


```
selected_id = tree.item(selected_item, 'values')[0]
```

Sefer seç butonuna basıldığı zaman seferin ID değerini yukardaki kod ile çekiyorum ve ekleyeceğim kayıttın SEFER_ID değerine ekliyorum.

```
mycursor.execute("SELECT BİLET_TUTARI FROM seferler WHERE ID = %s", (selected_id,))  
tutar=mycursor.fetchone()
```

Yukardaki kod ile seçtiğim seferin ücretini where şartı ile çekiyorum ve hesaplama fonksiyonuna yolluyorum.

BİLETLER	
ID	INT AUTO INCREMENT
YOLCU_ID (F)	INT
SEFER_ID (F)	INT
ÜCRET	INT



YAKUP UZUNOGLU

Bilet al

Ödeme bilgileri

Kargo ekle

Çıkış yap

Sefer seç						
ID	KALKIS_SEHIR_ID	VARIS_SEHIR_ID	KUYRUK_NO	KALKIS_ZAMANI	BİLET_TUTARI	YOLCU_SAYISI
2	İstanbul	Bursa	x78	2023-12-25 12:30:	1200	250
3	İstanbul	Trabzon	x96y	2023-12-25 14:00:	2000	250
4	İstanbul	Trabzon	x96y	2023-12-25 23:00:	2000	250
7	İstanbul	Manisa	x96y	2023-12-25 23:00:	2000	250
8	İstanbul	Afyon	x96y	2023-12-25 23:00:	2000	250
9	İstanbul	Şanlıurfa	x96y	2023-12-25 23:00:	2000	250
10	İstanbul	Yozgat	x96y	2023-12-25 23:00:	2000	250
11	İstanbul	Manisa	x96y	2023-12-25 23:00:	2000	250

Kalkış nokta : Varış nokta :

Ek Özellikler

☐ First class (+2000)

☐ Business class (+1500)

☐ Premium economy class (+1000)

☐ Economy class (+500)

0 TL

Hesapla

Ödeme yap

```
firstclass_label=customtkinter.IntVar(value=0)
firstclass=customtkinter.CTkCheckBox(master=frame_sag_ust,text="First class (+2000)",variable=firstclass_label, onvalue=2000,
offvalue=0)
```

Yukardaki kod da seçilen paketin ekstra fiyatını alıyorum ve hesapla fonksiyonumun içinde get fonksiyonu ile çekiyorum

```
def hesapla():
    hes1=firstclass_label.get()
    hes2=businessclass_label.get()
    hes3=premium_label.get()
    hes4=economyclass_label.get()
    global hes5
    hes5=hes1+hes2+hes3+hes4+tutar[0]
    label_text = customtkinter.IntVar()
    label_text.set(hes5)
    label1=customtkinter.CTkLabel(master=frame_sag_ust,textvariable=label_text,font=('century
gothic',17),fg_color="#808080",width=300,corner_radius=10,bg_color="white")
    label1.place(x=400,y=350)
```

Yukardaki hesapla fonksiyonu ile toplam fiyatı hesaplatıp tutar değişkenine atıyorum ve biletler tablosunun UCRET değerine ekliyorum ve tutarı label widgeti ile ekrana yazdırıyorum.

```
insert2 = "INSERT INTO biletler (YOLCU_ID,SEFER_ID,UCRET) VALUES (%s, %s, %s)"
degerler2 = (ID,selected_id,hes5)
mycursor.execute(insert2, degerler2)
mydb.commit()
```

Yukardaki kod ile aldığım tüm parametreleri biletler tabloma ekliyorum

FİLTRELEME İŞLEMİ

Seferleri listelerken istenilen şehirler arasındaki seferleri listelemek için;

```
SELECT s.ID, s1.SEHİR_ISMI AS KALKIS_SEHIR, s2.SEHİR_ISMI AS VARIS_SEHIR, s.KUYRUK_NO, s.KALKIS_ZAMANI, s.BİLET_TUTARI,
s.YOLCU_SAYISI
FROM seferler AS s
JOIN sehir s1 ON s1.ID = s.KALKIS_SEHIR_ID
JOIN sehir s2 ON s2.ID = s.VARIS_SEHIR_ID
WHERE s.KALKIS_SEHIR_ID = %s AND s.VARIS_SEHIR_ID = %s;
```

Yukardaki SQL sorgusu ile kayıtları filtreleme işlemi yapıyorum

