Bil 372: Veritabanı Sistemleri

PROJE RAPORU: GRUP 7

Arda Ege İsker, Baran Özgenç, Elçin Duman, Elif Gül

1. **Açıklama**

Projede CafféCo şirketi için kullanıcı ara yüzü ve veritabanı tasarlandı. Özel bir şirket olan CafféCo kahve üretimi, dağıtımı ve teminatından sorumludur.

Projeyi geliştirirken veritabanı yönetim sistemi olarak PostgreSQL kullanıldı. Yazılım geliştirme dili olarak Python tercih edildi ve Web uygulama ihtiyaçlarını karşılamak için Flask kütüphanesinden yararlanıldı.

Kullanıcı ara yüzünde şirketin alım satım gibi ticari işleri hakkında ve kahve üreticisi üzerine bilgiler görüntülenebilir. Şirketin şahsi araçları ve araçların oluşturduğu filo listelenebilir. Ayrıca koordinatörün sorumlu olduğu teknik bilgilerin yer aldığı kısım vardır. Bu bölümde kahve öğütme, kahve kavurma, biten işlemler, kahve çekirdeği ve kahve paketlenmesi hakkında detaylı bilgiler bulunmaktadır. Ara yüzün her sekmesinde o sekmeye ait bilgiler tablo şeklinde listelenmektedir ve tabloyu filtreleme özelliği mevcuttur. Burada veritabanında arama gerçekleştirilebilir ve çıkan sonuçlar listelenebilir. Ara yüzde ayrıca iletişim kısmı da yer almaktadır. İletişim sayfası kullanıcıdan feedback almak için tasarlanmıştır.

Ara yüzde üç adet kullanıcı türü vardır. Bunlar Admin, Koordinatör ve Nakliyeci rolleridir. Admin arayüze personel eklenmesinden, kahve üretim süreci için yeni işlemler belirlenmesinden ve şirket araçlarının veritabanına kaydından sorumludur. Koordinatör ve Nakliyeciler personeldir. Koordinatör kahve hakkında teknik bilgilerin sisteme girilmesinden ve alım/satım işlemlerinden sorumludur. Nakliyeci ise üretici ile firma arasında bağlantı görevini üstlenmektedir. Üreticiden kahve çekirdekleri teminatı ve sisteme üretici kaydıyla ilgilenmektedir.

Web sitesine girişte ilk olarak login ve register seçeneği çıkmaktadır. Eğer bir personelin veritabanında kaydı admin tarafından yapılmışsa o kullanıcı register sekmesinden kendine bir parola belirleyip kaydolabilir. Daha sonra Login ekranından bilgilerini kullanarak giriş yapabilir. Login aşamasında girilen TCKN veritabanında aranır ve o kullanıcı hangi role sahipse ona göre erişebileceği operasyonlara yönelik girişler gerçekleştirilebilir.

Appendix kısmında HIPO Diagram, ER, Relational Mapping ve Data Flow mevcuttur.

* 1. **Tablolar ve Web Sayfaları**

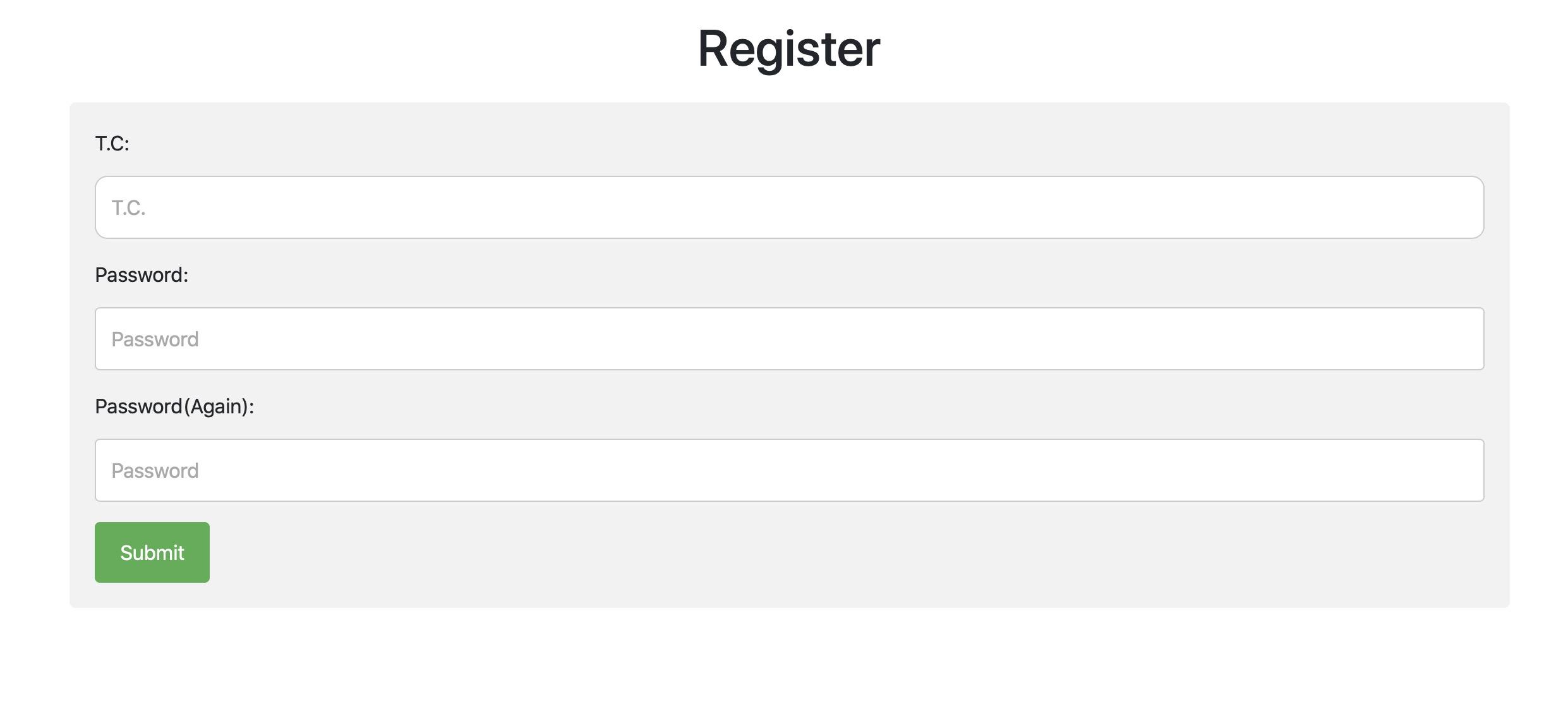
1. **Register**

Bu sayfada kayıt olmak isteyen personel TCKN’sini girer ve kendine bir şifre belirleyerek sisteme kaydolur. Eğer daha önceden personel tablosunda kaydı yoksa hata verilir. Sisteme kayıt olabilmek için personel olma şartı vardır. Buradaki ilk SQL cümlesi Register olmaya çalışan kullanıcının veritabanında personel kaydı var mı onu bulmak için çalıştırılır.

roleQuery = "SELECT \* FROM public.\"Personel\" WHERE tckn = '" + tckn + "' "

İkinci SQL cümlesi ise Register olmaya çalışan kullanıcının şifre ve TCKN bilgilerini veritabanına kaydeder.

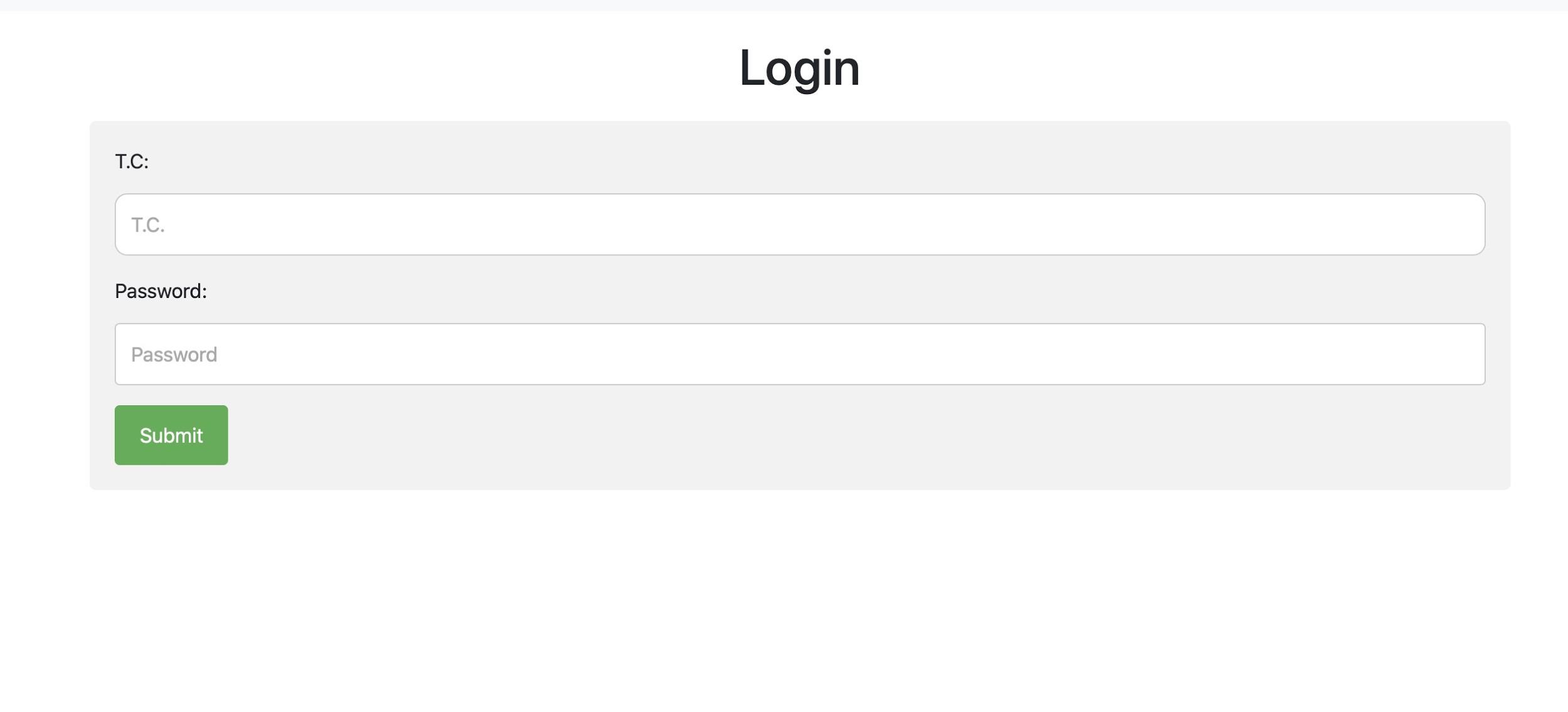
query = "INSERT INTO public.\"Login\"(tckn, passcode, personel\_tipi) VALUES ('" + tckn + "' , '" + password + "', '" + userType + "')"

.

1. **Login**

Login ekranında ise daha önceden sisteme kaydolan personel belirlediği şifre ve TCKN ile giriş yapar. Eğer şifre hatalıysa sistem hata mesajı verir ve kullanıcıdan yeniden girmesi istenir. Giriş işlemi başarılıysa kullanıcı tipine göre admin, nakliye veya koordinatör ana sayfası açılır. SQL cümlesi ile login yapmaya çalışan kullanıcının kaydı veritabanında var mı ona bakılır. Eğer var ise daha sonra parolası da doğru girilmişse giriş işlemi gerçekleştirilir.

**query = "SELECT \* FROM public.\"Login\" WHERE tckn = '" + tckn + "' "**



1. **ADMİN GİRİŞİ**
2. **Personel Sayfası**

Bu sayfa admin için ana sayfadır. Bu sayfada admin personel ekleyebilir, ekli olan personelin bilgilerini düzeltebilir veya personeli silebilir. Silmesi ve düzenlemesi durumunda cascade yapılır, böylece foreign key violation hatası ile karşılaşılmaz.

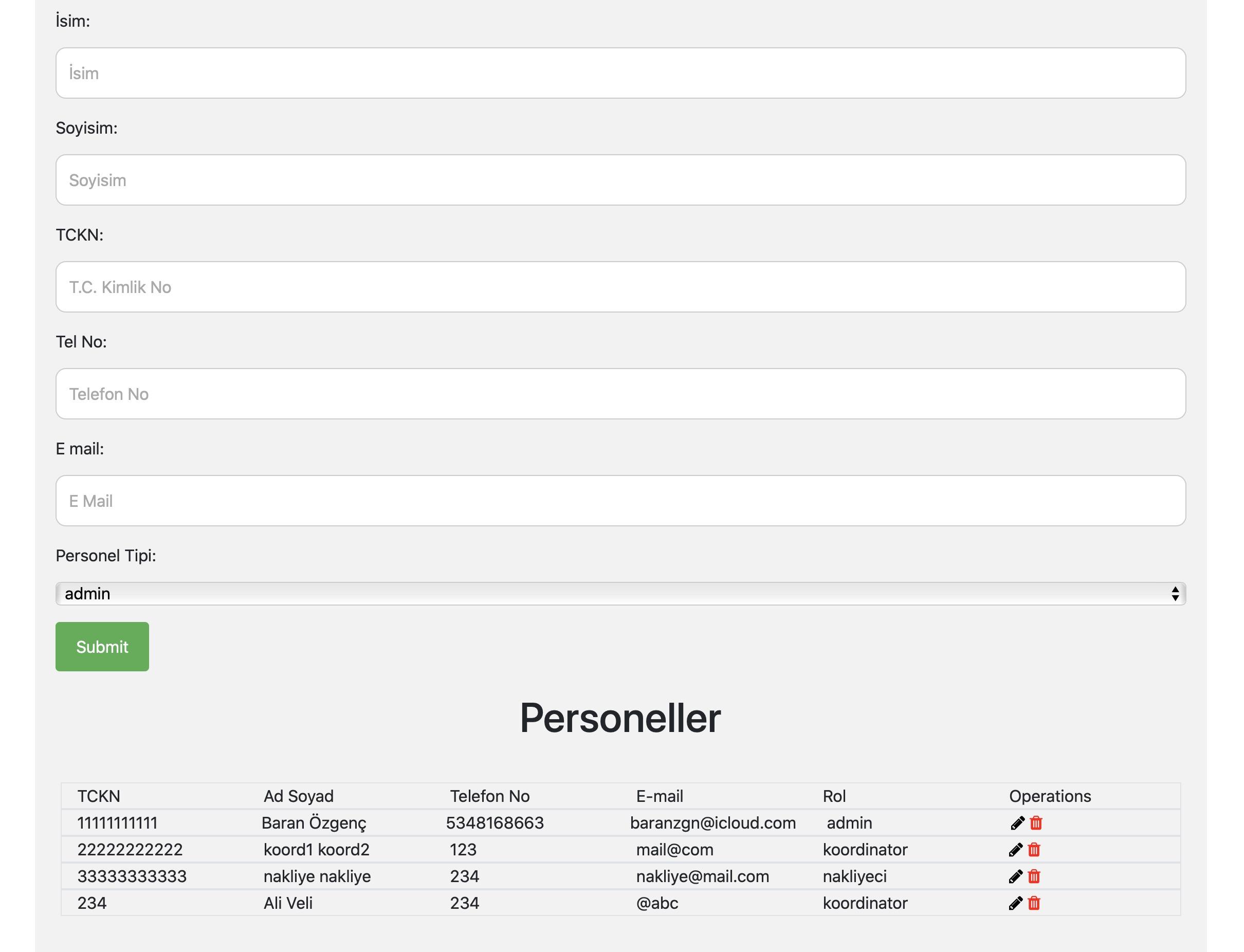
Daha güvenli olması açısından örneğin primary key olan TCKN’nin güncellenmesine izin verilmez ve TCKN input alanı read only yapılır. Ayrıca personeller tablosunu görüntülerken filtreleme yapılma özelliği de bulunmaktadır.

Insert: insertQuery = "INSERT INTO public.\"Personel\"(tckn, ad, soyad, tel\_no, email, personel\_tipi) VALUES ('" + tckn + "','" + ad + "','" + soyad + "', '" + tel\_no + "' , '" + email + "', '" + personel\_tipi + "')"

Update: update\_query = "UPDATE public.\"Personel\" SET tckn = '" + tckn + "', ad = '" + ad + "', soyad = '" + soyad + "', tel\_no = '" + tel\_no + "', email = '" + email + "', personel\_tipi = '" + personel\_tipi + "' WHERE tckn = '" + *str*(id) + "'"

Select: queryPersonel = "SELECT \* FROM public.\"Personel\""

Delete: query2 = "DELETE FROM public.\"Personel\" WHERE tckn = '" + *str*(id) + "'"



1. **İşlem Türü**

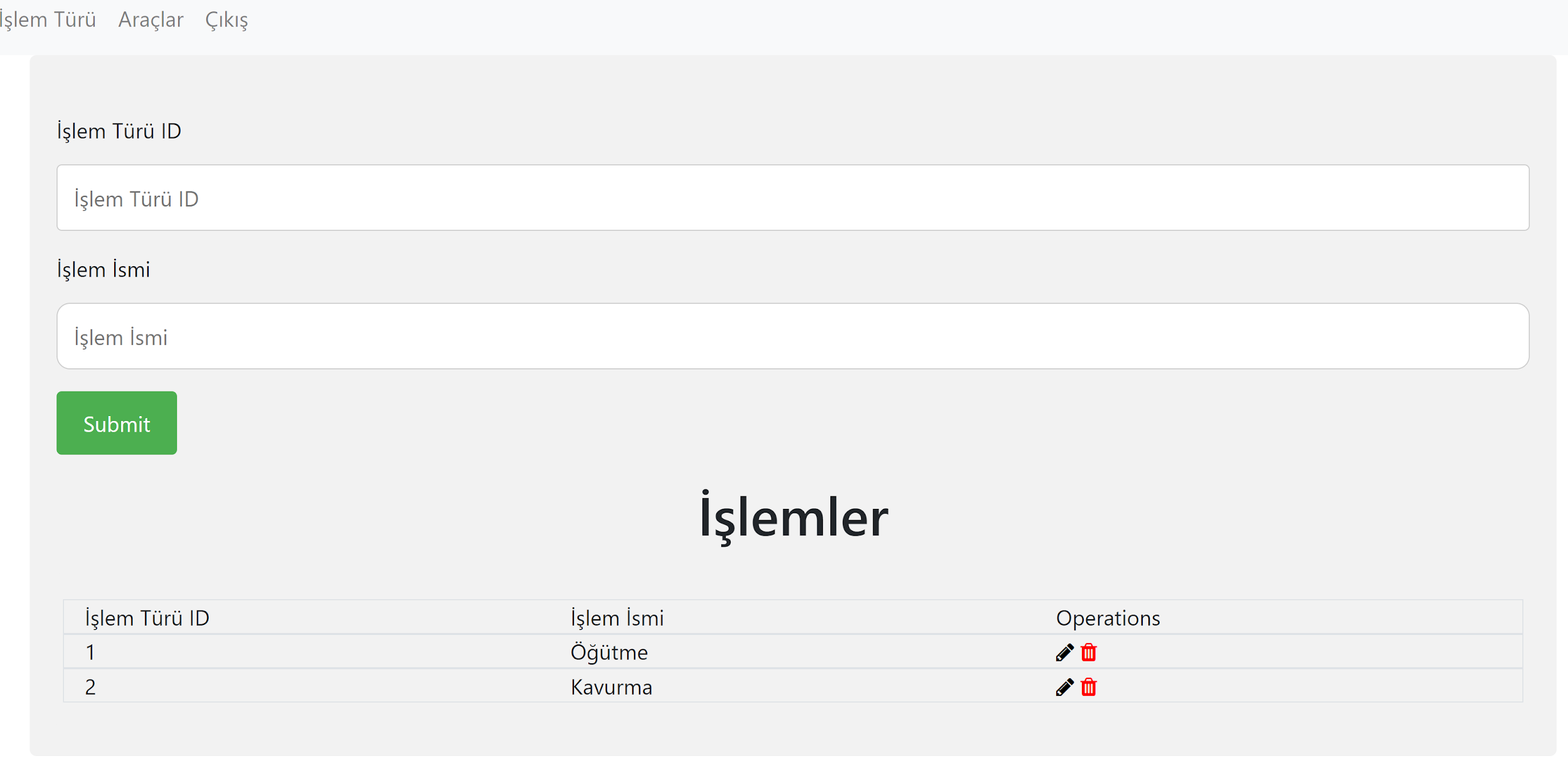
Bu sayfada admin kahve çekirdeğinin geçeceği işlemlere ekleme yapabilir silebilir veya görüntüleyebilir. Tasarımda mevcut olarak öğütme ve kavurma işlemleri vardır ancak yeni bir işlem eklenmesi senaryosunda esneklik sağlayabilmek adına böyle bir sayfa oluşturulmuştur. Burada yeni bir işlem türü oluşturulduktan sonra yapılması gereken tek şey veritabanı üzerinde yeni işlem için bir tablo oluşturmaktır.

Insert: query = "INSERT INTO public.\"Islem\_Turu\"(tur\_id, islem\_ismi) VALUES ('" + tur\_id + "','" + islem\_ismi + "')"

Update: query2 = "UPDATE public.\"Islem\_Turu\" SET islem\_ismi = '" + islem\_ismi + "'" + " WHERE tur\_id = '" + *str*(tur\_id) + "'"

Delete:query = "DELETE from public.\"Islem\_Turu\" where tur\_id = '" + *str*(tur\_id) + "'"

Select: query = "SELECT \* FROM public.\"Islem\_Turu\""



1. **Araçlar**

Bu sayfada admin araç ekleyebilir, mevcut araçları düzenleyebilir veya silebilir. Burada oluşturulan araçlar veritabanında Araç tablosuna kaydolur ve nakliyeci rolündeki personeller bu araçları kullanarak üreticiden satın alım işlemlerini gerçekleştirir. Ayrıyeten, filo görüntüleme tablosunda araç plakasına göre filtreleme yapma özelliği mevcuttur.

Insert: query = "INSERT INTO public.\"Arac\"(plaka, kapasite) VALUES ('" + plaka + "' , '" + kapasite + "')"

Update:queryUpdate = "UPDATE public.\"Arac\" SET kapasite = '" + kapasite + "'" + " WHERE id = '" + *str*(id) + "'"

Select:queryAraclar = "SELECT \* FROM public.\"Arac\""

Delete:queryDel = "DELETE FROM public.\"Arac\" WHERE id = '" + *str*(id)



1. **Koordinatör Girişi**
2. **Öğütme**

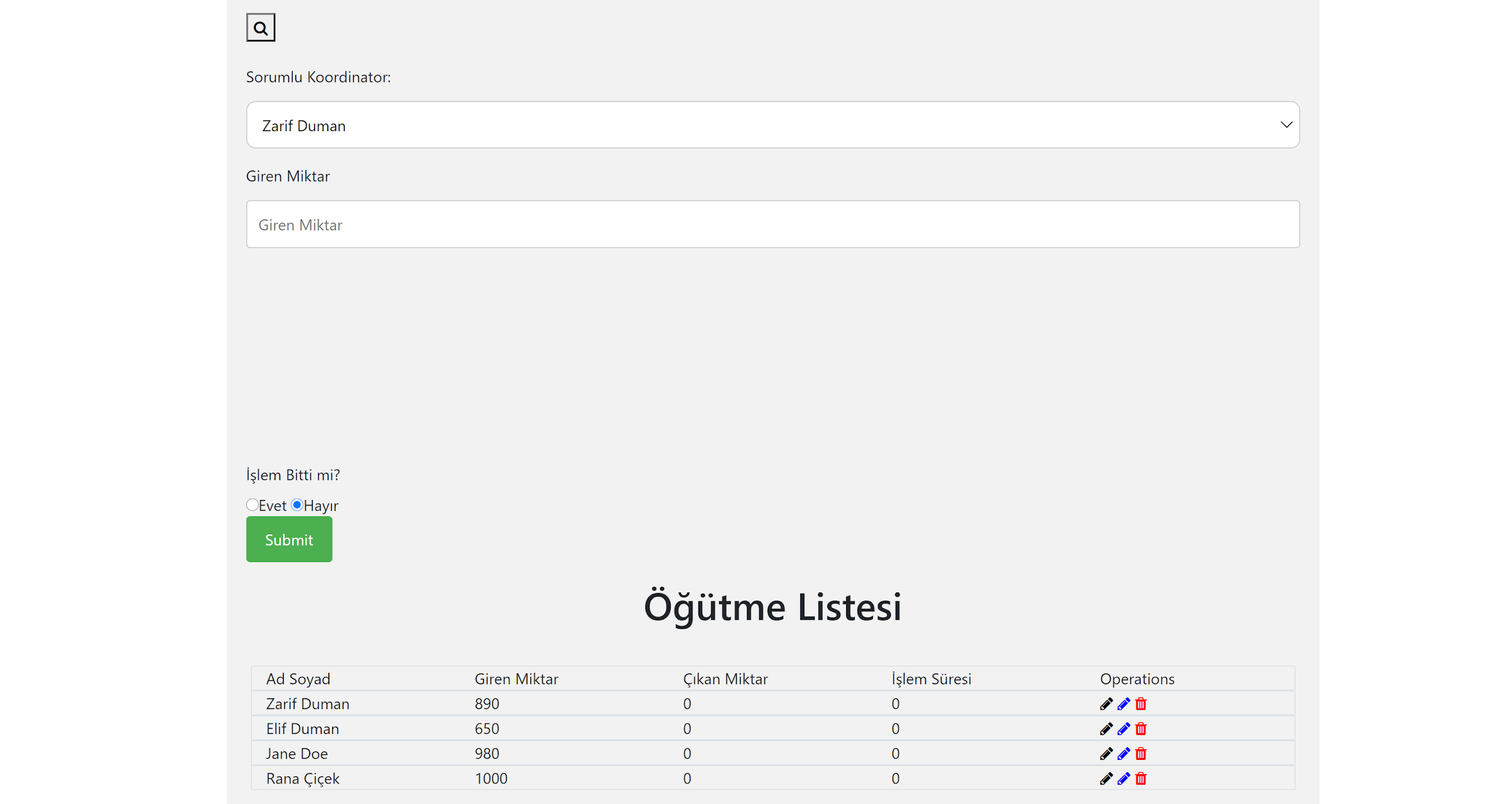
Bu sayfada önce öğütme işleminden sorumlu koordinatör seçilir. Daha sonra işleme giren ham kahve miktarı girilir. İşlem başlatıldığı için radio button default olarak Hayır seçilir.

Operations kısmındaki ilk kalem işareti ile kullanıcı giren miktarı düzenleyebilir. İkinci (mavi) buton ise işlemi bitirmek için vardır. Mavi düzenleme butonu seçilince form alttaki hale döner ve oluşan miktar ve işlem süresi girilerek işlem sonlandırılır. Eğer işlem bitmiş ise Biten işlem İşlem Sonu tablosunda listelenir. Bitmemiş seçilirse aynı sayfadaki tabloda görüntülenir.

İnsert SQL: query = "INSERT INTO public.\"Ogutme\" (sorumlu\_koordinator\_tckn, tur\_id, giren\_miktar, cikan\_miktar, islem\_suresi, bitti\_mi) VALUES ('" + sorumlu\_koordinator\_tckn + "','" + str(1) + "','" + str(giren\_miktar) + "','" + str(0) + "','" + str(0) + "','" + str(bitti\_mi) + "')"

Update SQL: query = "UPDATE public.\"Ogutme\" SET sorumlu\_koordinator\_tckn = '" + sorumlu\_koordinator\_tckn + "', tur\_id = '" + tur\_id + "', giren\_miktar = '" + giren\_miktar + "', cikan\_miktar = '" + cikan\_miktar + "', islem\_suresi = '" + islem\_suresi + "', bitti\_mi = '" + bitti\_mi + "' WHERE id = '" + str( id) + "'"

Delete iSQL:query = "DELETE from public.\"Ogutme\" where id = '" + str(id) + "'"





****

1. **Kavurma sayfası:**

Bu sayfada önce kavurma işleminden sorumlu koordinatör seçilir. Daha sonra işleme giren kahve miktarı girilir. İşlem başlatılırken altta görülen radio button default olarak Hayır seçilidir. Eğer kullanıcı giriş miktarını düzenlemek isterse Operations bölümünden ilk kalem işaretine tıklar ve giren miktarı değiştirir.

İşlemi bitirmek için kullanıcı Operations bölümünden ikinci (mavi) kalem işaretini seçer. Bu defa form aşağıda görüldüğü gibi genişler ve kullanıcıdan oluşan kahve miktarını ve işlem süresi bilgileri girmeleri istenir. Bu noktada eğer işlem bitmişse kullanıcı Evet’i seçince mevcut işlem alttaki tablodan kaybolarak işlem sonu tablosuna aktarılır. Eğer hayır seçilirse işlem tabloda görüntülenmeye devam eder.

Kavrulma sayfası için SQL cümleleri:

Insert SQL:

insertQuery = "INSERT INTO public.\"Kavurma\" (sorumlu\_koordinator\_tckn, tur\_id, giren\_miktar, cikan\_miktar ,islem\_suresi, bitti\_mi) VALUES ('" + tckn + "' , '" + tur\_id + "' ,'" + giren\_miktar + "' , '" + + *str*(id) + “’”

Update SQL: query\_update = "UPDATE public.\"Kavurma\" SET sorumlu\_koordinator\_tckn = '" + tckn + "', tur\_id = '" + tur\_id + "', giren\_miktar = '" + giren\_miktar + "', cikan\_miktar = '" + cikan\_miktar + "', islem\_suresi = '" + islem\_suresi + "', bitti\_mi = '" + bitti\_mi + "' WHERE id = '" + *str*(id) + “’”

İşlem Sonu Insert SQL:queryIslemSonu = "INSERT INTO public.\"Islem\_Sonu\"(sorumlu\_koordinator\_tckn, tur\_id, id3) VALUES ('" + tckn + "' , '" + tur\_id + "' ,'" + *str*(id)

Select SQL:

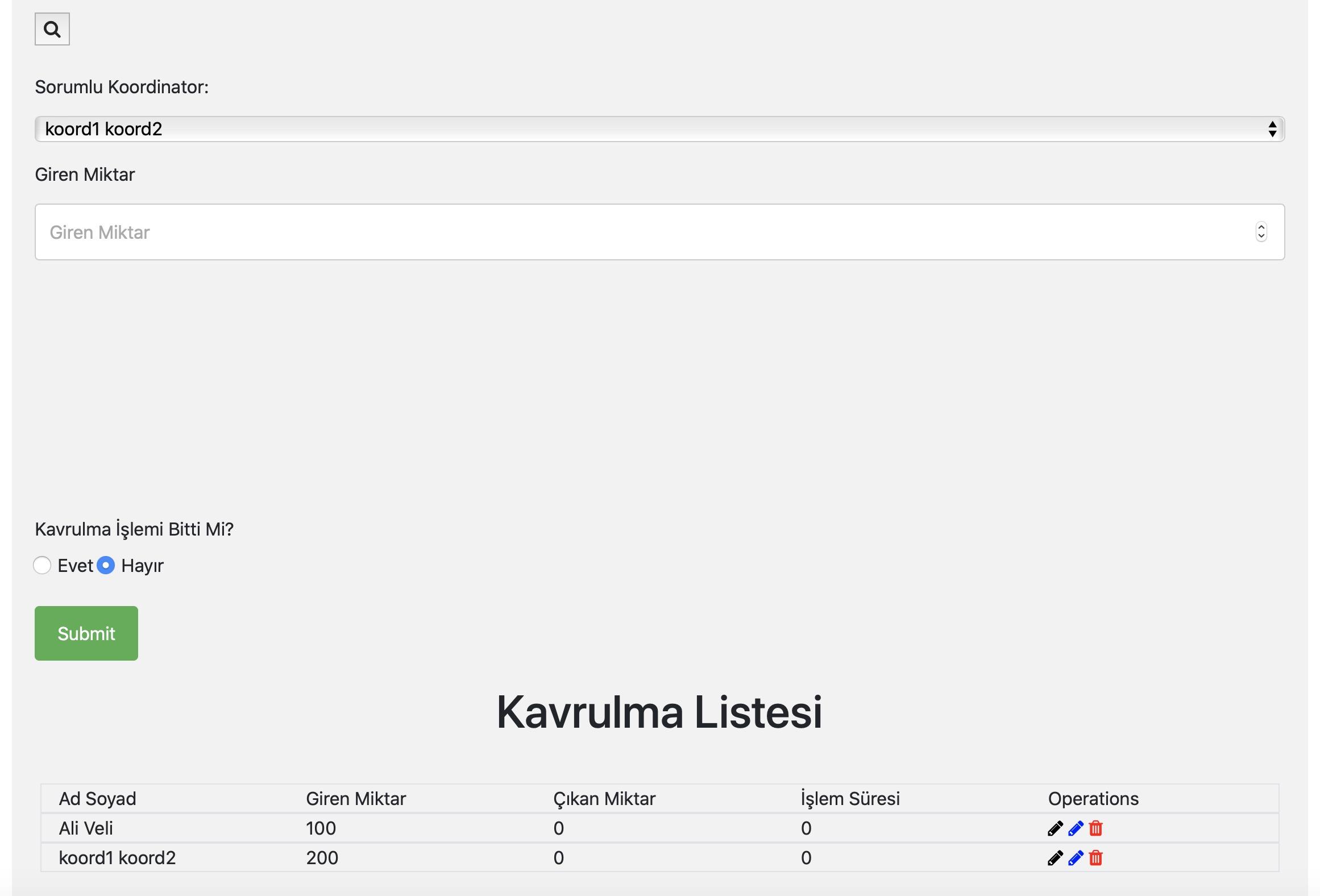
query = "SELECT \* FROM public.\"Personel\" WHERE personel\_tipi = 'koordinator'"

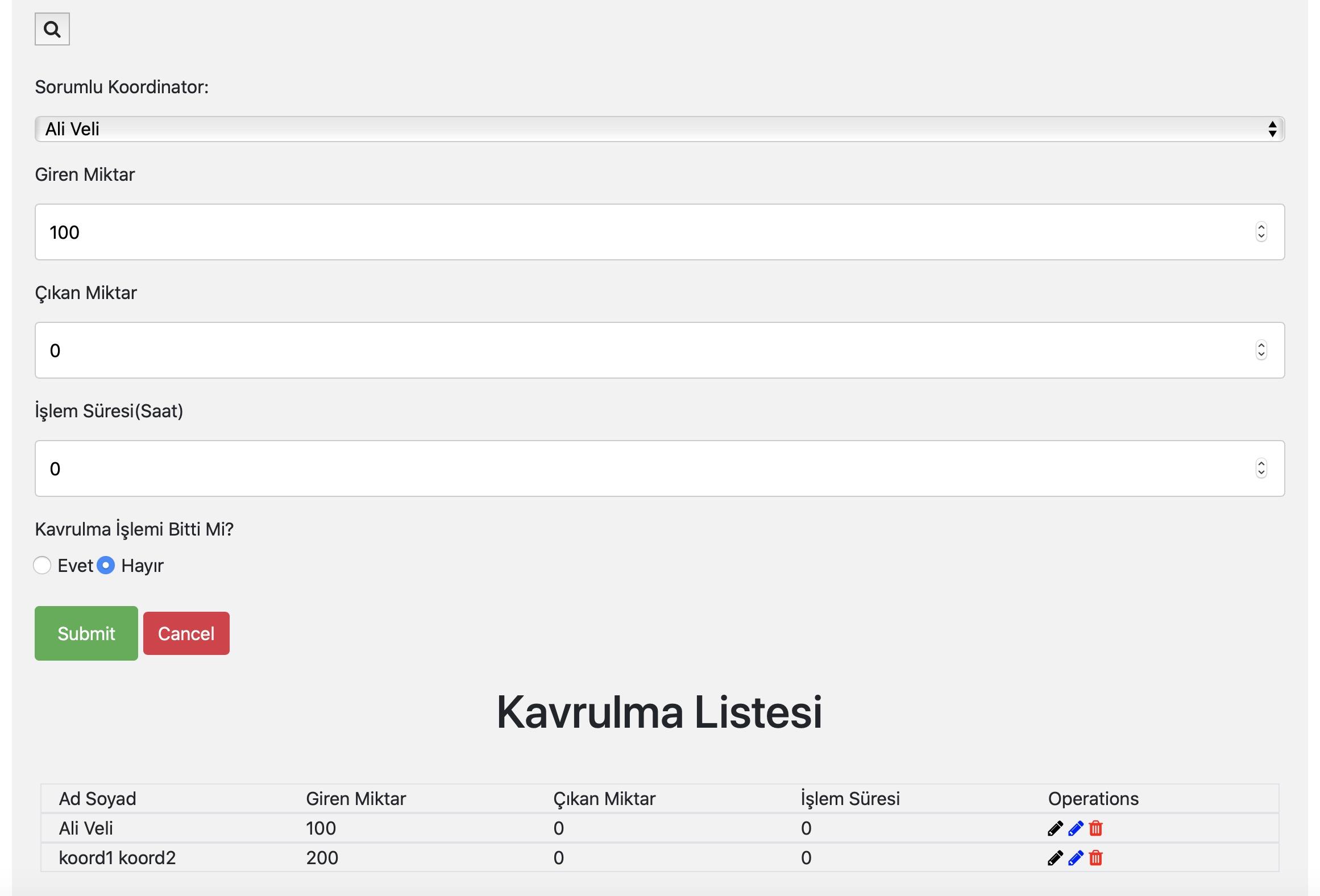
Kavurma sayfasındaki işlemleri görüntülemek için SQL:

queryJoin = "SELECT public.\"Kavurma\".giren\_miktar, public.\"Kavurma\".bitti\_mi ,public.\"Kavurma\".id, public.\"Kavurma\".sorumlu\_koordinator\_tckn, public.\"Kavurma\".cikan\_miktar, public.\"Kavurma\".islem\_suresi, public.\"Personel\".ad, public.\"Personel\".soyad FROM (public.\"Kavurma\" INNER JOIN public.\"Personel\" ON public.\"Kavurma\".sorumlu\_koordinator\_tckn = public.\"Personel\".tckn) WHERE public.\"Kavurma\".bitti\_mi = False "

Kavurma işlemi Silmek için SQL:

queryDel = "DELETE FROM public.\"Kavurma\" WHERE id = '" + *str*(id) + "'"



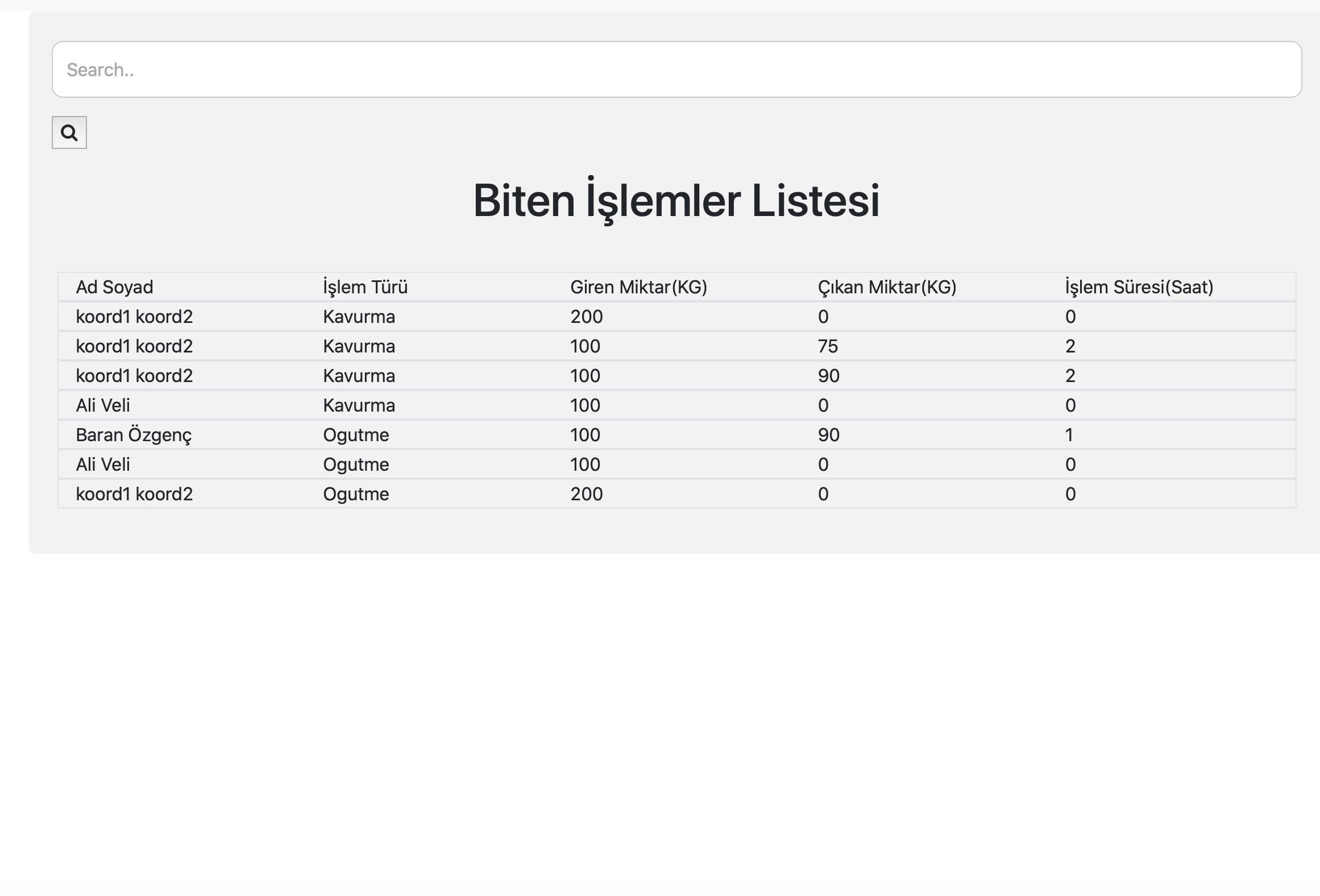


1. **İşlem Sonu sayfası:**

Bu sayfada biten işlemler listesi görüntülenir. Sırasıyla sorumlu koordinatör, işlem türü, işleme giren miktar, işlemden çıkan miktar ve işlem süresi listelenir. Burada iki farklı tablodan veri aldığımız için görüntülemek için iki ayrı SQL cümlesi kullanıldı ve ikisinin sonucu birleştirildi.

q2 = "SELECT public.\"Ogutme\".giren\_miktar, public.\"Ogutme\".cikan\_miktar, public.\"Ogutme\".islem\_suresi, public.\"Personel\".ad, public.\"Personel\".soyad, public.\"Islem\_Turu\".islem\_ismi FROM(( public.\"Ogutme\" INNER JOIN public.\"Personel\" ON public.\"Ogutme\".sorumlu\_koordinator\_tckn = public.\"Personel\".tckn) INNER JOIN public.\"Islem\_Turu\" ON public.\"Ogutme\".tur\_id = public.\"Islem\_Turu\".tur\_id)"

q3 = "SELECT public.\"Kavurma\".giren\_miktar, public.\"Kavurma\".cikan\_miktar, public.\"Kavurma\".islem\_suresi, public.\"Personel\".ad, public.\"Personel\".soyad, public.\"Islem\_Turu\".islem\_ismi FROM(( public.\"Kavurma\" INNER JOIN public.\"Personel\" ON public.\"Kavurma\".sorumlu\_koordinator\_tckn = public.\"Personel\".tckn) INNER JOIN public.\"Islem\_Turu\" ON public.\"Kavurma\".tur\_id = public.\"Islem\_Turu\".tur\_id)"



1. **Çekirdek sayfası:**

Bu sayfada Kahve çekirdekleri işlenir. Mevcut çekirdekler tablosunda, şirket veritabanında var olan çekirdekler listelenir.

Çekirdek eklemek için Çekirdek miktarı (gram cinsinden), çekirdektürü ve çekirdeğin kökeni girilir. Girilen çekirdekler mevcut çekirdekler tablosunda listelenir.Mevcut çekirdekler için ayrıca silme ve düzenleme işlemi de yapılabilir.Çekirdek sekmesinde, kullanıcı çekirdeğin miktarını, türünü ve kökenini girer.

Insert SQL:

"INSERT INTO \"Cekirdek\"(koken, miktar, tur) VALUES ('" + koken + "' , '" + miktar + "', '" + tur + "')"

Update SQL:

"UPDATE public.\"Cekirdek\" SET koken = '" + koken + "', miktar = '" + miktar + "', tur = '" + tur + "' WHERE id = '" + str(id) + "'"

Delete SQL:

"DELETE FROM public.\"Cekirdek\" WHERE id = '" + str(id) + "'"

Listelemek için kullanılan SQL:

SELECT \* FROM public.\"Cekirdek\" "



1. **Paket sayfası:**

Bu sayfada Kahve paketleri işlenir. Mevcut paketler tablosunda, şirket veritabanında var olan paketler listelenir. Paketleri eklemek için Paket gramajı, paket türü ve paketin son kullanma tarihi girilir. Girilen paketler mevcut paketler tablosunda listelenir. Mevcut paketler için ayrıca silme ve düzenleme işlemi de yapılabilir.

Update SQL’i: "UPDATE public.\"Paket\_Kahve\" SET tur = '" + tur + "', gramaj = '" + gramaj + "', skt = '" + skt + "' WHERE id = '" + str(id) + "'"

Insert SQL:

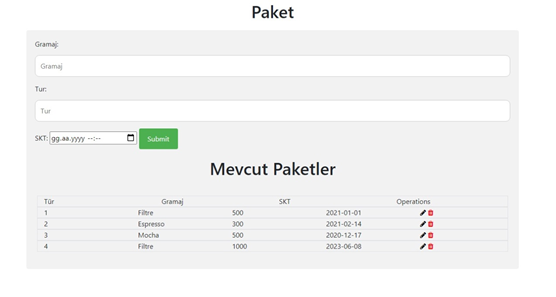
"INSERT INTO \"Paket\_Kahve\"(gramaj, skt, tur) VALUES ('" + gramaj + "','" + skt + "', '" + tur + "')"

Delete SQL’i:

query = "DELETE FROM public.\"Paket\_Kahve\" WHERE id = '" + str(id) + "'"

Listelemek için kullanılan SQL:

"SELECT \* FROM public.\"Paket\_Kahve\""



1. **Satış Sayfası:**

Bu sayfada satışla ilgili bilgiler tabloda gösterilir. Her satışın ücreti, tarihi, miktarı olur ve satış alıcı şirket ID değeri vardır. Ayrıca satış işleminin ID değişkeninde de serial değer tutulur. Satış eklemek için alıcı şirket ID değeri, satışın ücreti, satış miktarı ve satış tarihi seçilir. Ekleme, silme, arama ve düzenleme işlemleri de bu sayfadan yapılır.

Ayrıca arama butonundan şirket ID değerine göre, şirketin satışları listelenebilir.

INSERT SQL:

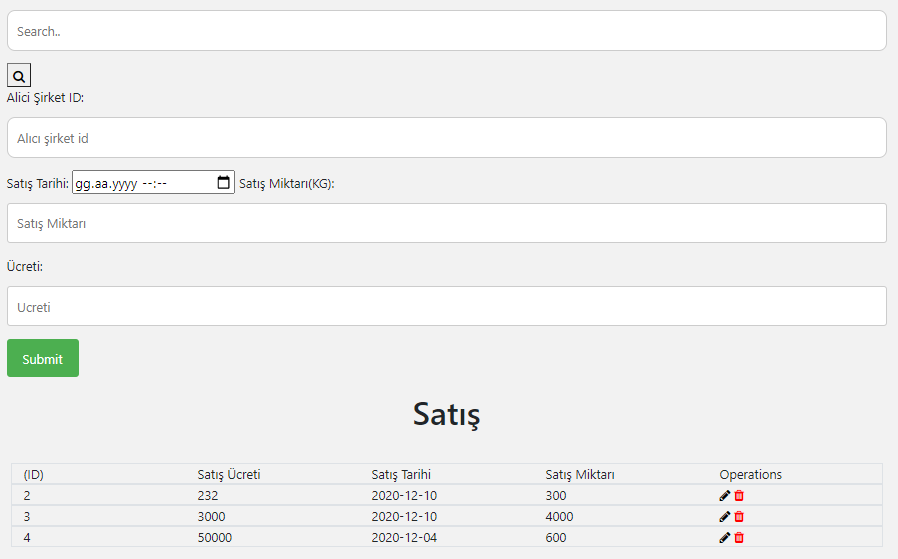
"INSERT INTO public.\"Satis\"(alici\_sirket\_id, ucret, tarih,miktar) VALUES ('" + alici\_sirket\_id + "', '" + ucret + "','" + tarih + "','" + miktar + "')"

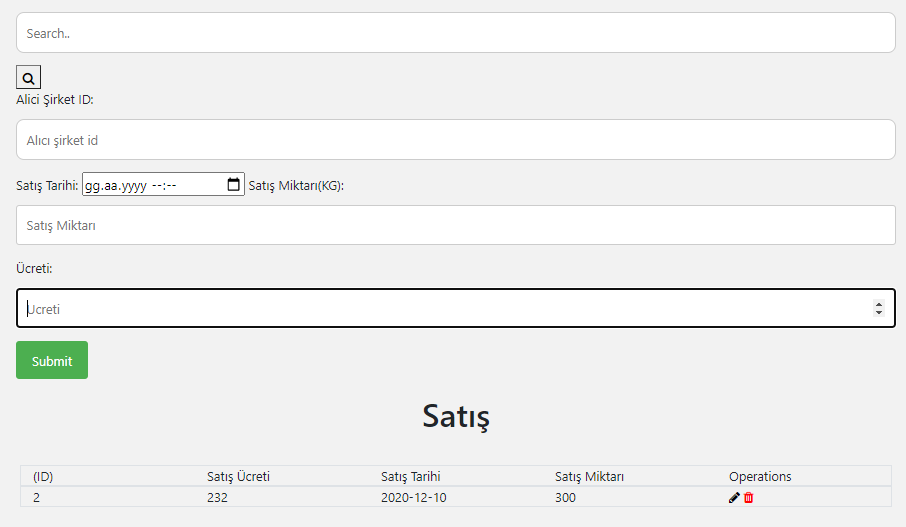
Update SQL:

"UPDATE public.\"Satis\" SET alici\_sirket\_id= '" + alici\_sirket\_id + "', tarih = '" + str(tarih) + "', ucret = '" + str(ucret) + "', miktar = '" + str(miktar) + "' where id = '" + str(id)+ "'"

Delete SQL:

"DELETE FROM public.\"Satis\" WHERE alici\_sirket\_id = '" + str(alici\_sirket\_id) + "' "





1. **Alıcı Sayfası:**

Alıcı sayfasında alım yapan şirketin adı ve ID değeri yer alır. Alıcı şirket arama, ekleme, silme ve düzenleme yapılır. Alıcı şirketler listelenmiş halde ad ve ID değerleriyle sayfanın altında bulunur.

Arama kısmına şirket ismi yazılınca listede varsa altta listelenir.

Insert SQL:

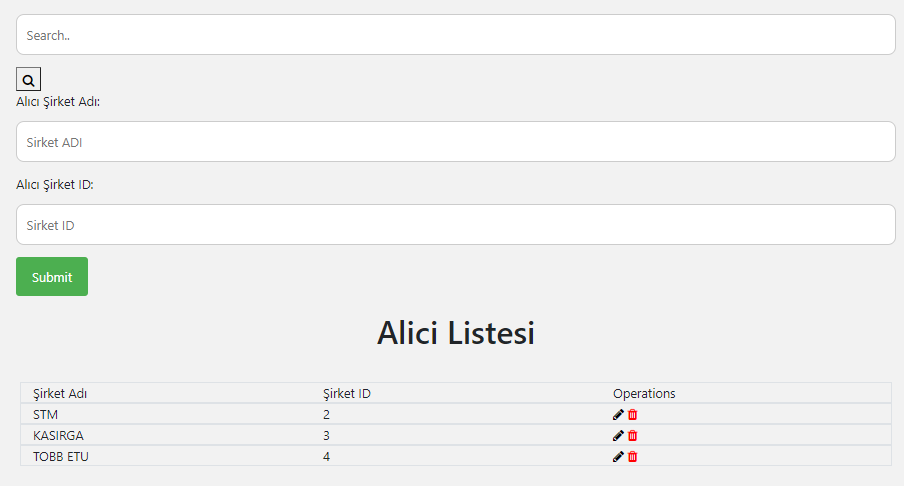
"INSERT INTO public.\"Satis\"(alici\_sirket\_id,ucret,tarih,miktar) VALUES ('" + alici\_sirket\_id + "', '" + ucret + "','" + tarih + "','" + miktar + "')"

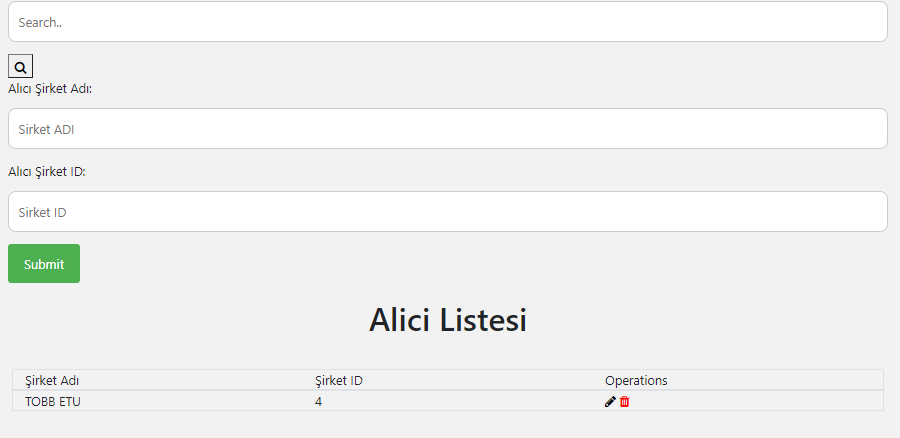
Update SQL:

"UPDATE public.\"Satis\" SET alici\_sirket\_id= '" + alici\_sirket\_id + "', tarih = '" + str(tarih) + "', ucret = '" + str(ucret) + "', miktar = '" + str(miktar) + "' where id = '" + str(id)+ "'"

Delete SQL Sorgusu:

"DELETE FROM public.\"Satis\" WHERE alici\_sirket\_id = '" + str(alici\_sirket\_id) + "' "





1. **İletişim Sayfası:**

Bu sayfa, kullanıcıdan geri bildirim almak için tasarlandı. Kullanıcı e-mailini ve mesajını girer.

Insert SQL ‘i:

query = "INSERT INTO \"Iletisim\"(email, mesaj) VALUES ('" + email + "' , '" + mesaj + "')"



1. **Nakliyeci**
2. **Satın Alım:**

Bu sayfa nakliyeci üreticiden ham kahve çekirdekleri satın alırken kullanılır.Formdaki üretici listesi alanı veritabanından veri çeker ve bu sayede üretici seçilir. Daha sonra ödeme miktarı, satın alınan ham çekirdek miktarı, üretici köyü gibi alanlar doldurulur ve kaydedilir.

Alttaki tablodan kullanıcı bir kaydı düzenleyebilir veya silebilir. Aynı zamanda tabloyu üretici adına, köyüne, satın almada kullanılan araç plakasına göre filtreleyebilir.

Insert SQL:

insertQuery = "INSERT INTO public.\"Satin\_Alir\" (uretici\_tckn, odeme\_tarihi,aciklama, odeme\_miktari ,urun\_miktari, plaka) VALUES ('" + uretici\_tckn + "' , '" + *str*(odeme\_tarihi) + "' , '" + aciklama +"' , '"+ str(odeme\_miktari) +"' , '"+ str(urun\_miktari)+ "' , '"+plaka +’)

Update:

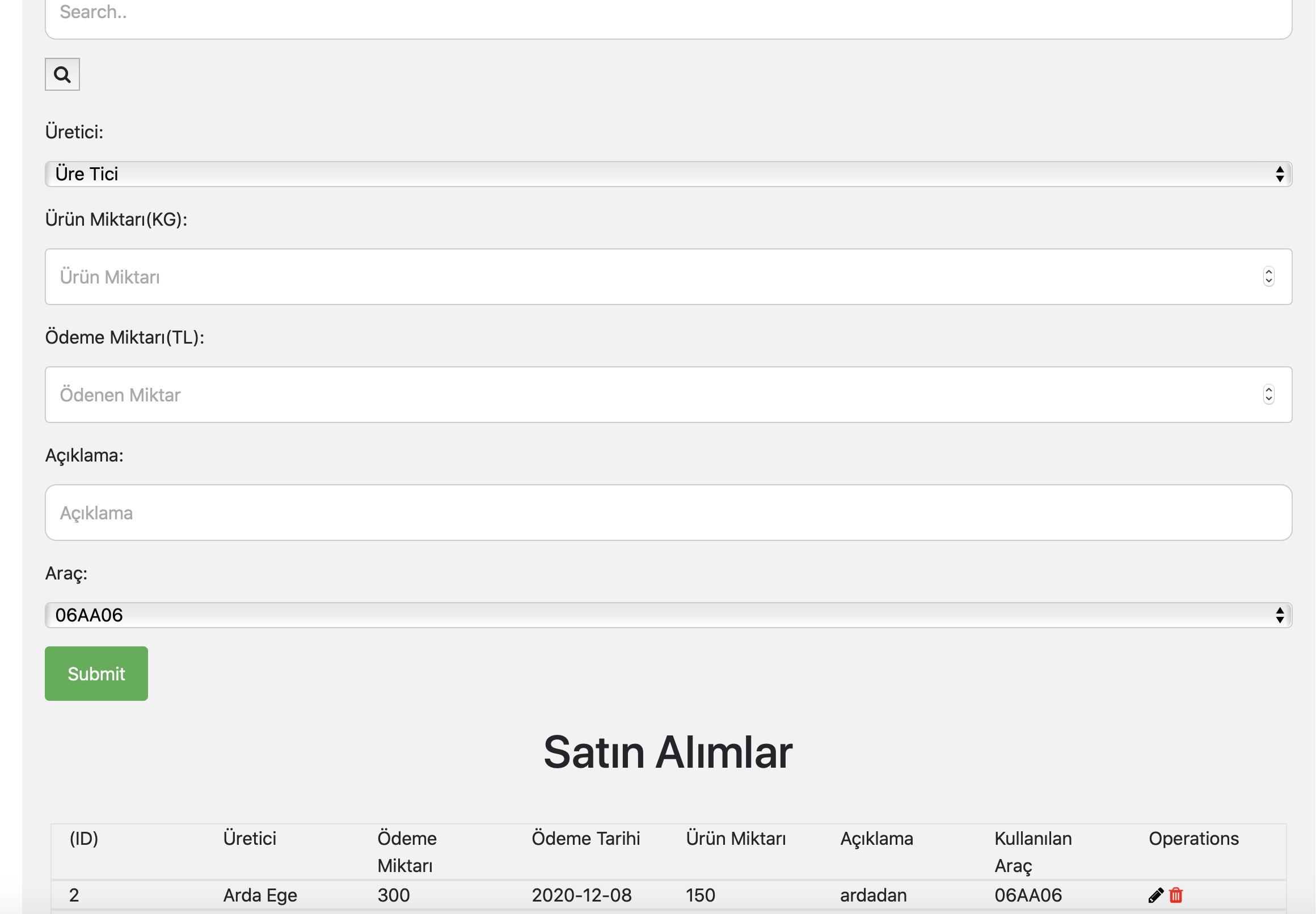
query\_update = "UPDATE public.\"Satin\_Alir\" SET uretici\_tckn = '" + uretici\_tckn + "', aciklama = '" + aciklama + "', odeme\_miktari = '" + odeme + "', odeme\_tarihi = '"

Listeleme(Select):

queryAll = "SELECT public.\"Satin\_Alir\".id, public.\"Satin\_Alir\".odeme\_miktari, public.\"Satin\_Alir\".odeme\_tarihi, public.\"Satin\_Alir\".urun\_miktari, public.\"Satin\_Alir\".aciklama, public.\"Satin\_Alir\".plaka, public.\"Uretici\".ad\_soyad FROM( public.\"Satin\_Alir\" INNER JOIN public.\"Uretici\" ON public.\"Satin\_Alir\".uretici\_tckn = public.\"Uretici\".tckn)"

Delete:

query = "DELETE FROM public.\"Satin\_Alir\" WHERE id = '" + *str*(id) + "'"



1. **Üretici**

Bu sayfada nakliyeci sisteme yeni üretici kayıtlarını gerçekleştirir, ya da mevcut kayıtları düzenleyebilir, silebilir veya görüntüleyebilir.

Insert SQL:

query = "INSERT INTO public.\"Uretici\"(ad\_soyad, tckn, koy, tel\_no) VALUES ('" + ad\_soyad + "', '" + tckn + "','" + koy + "','" + tel\_no + "')"

Update SQL:

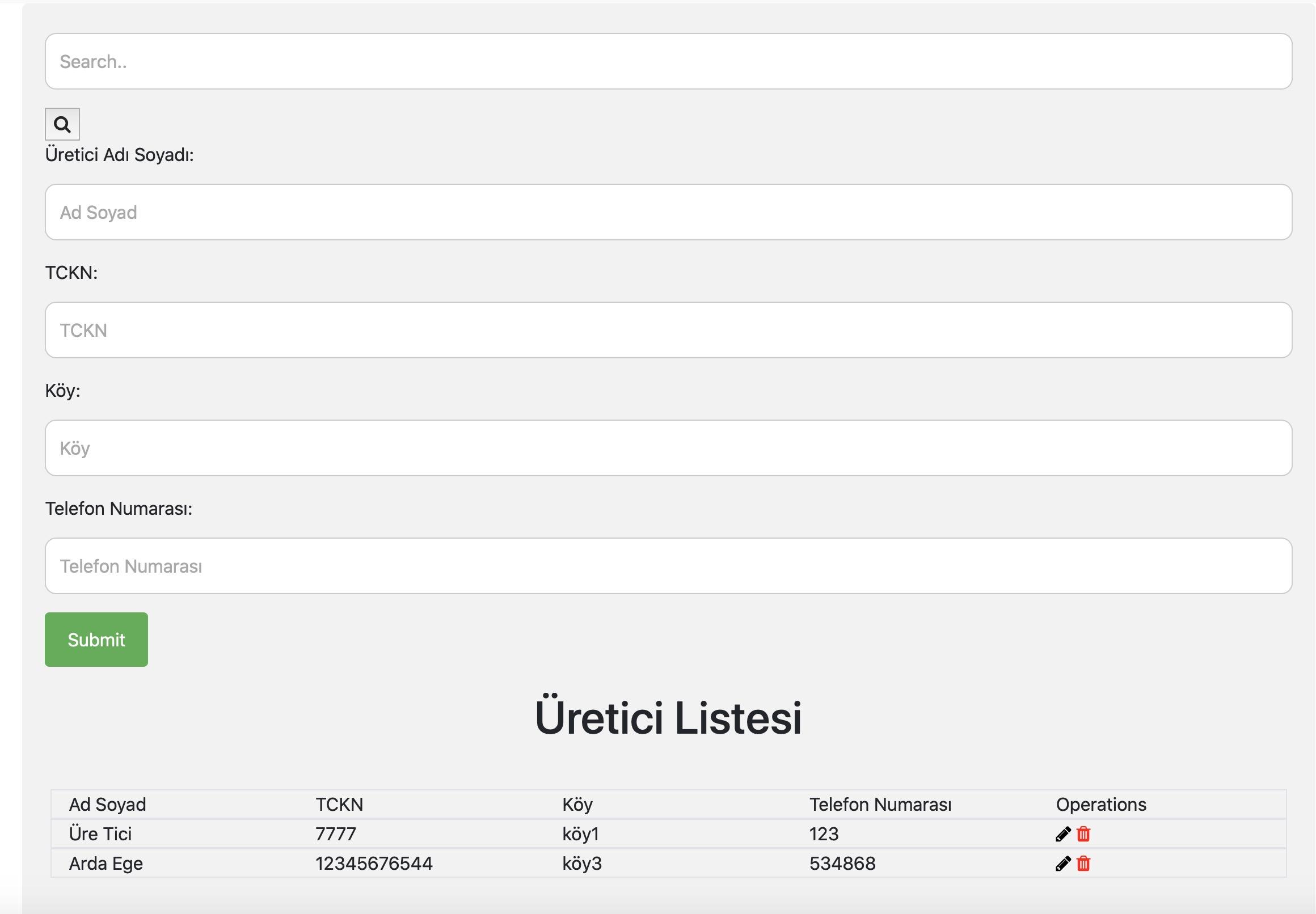
query = "UPDATE public.\"Uretici\" SET ad\_soyad ='" + ad\_soyad + "', tckn = '" + tckn + "', koy = '" + koy + "', tel\_no = '" + tel\_no + "' where tckn = '" + tckn + "'"

Delete SQL:

query = "DELETE FROM public.\"Uretici\" where tckn = '" + tckn + "'”

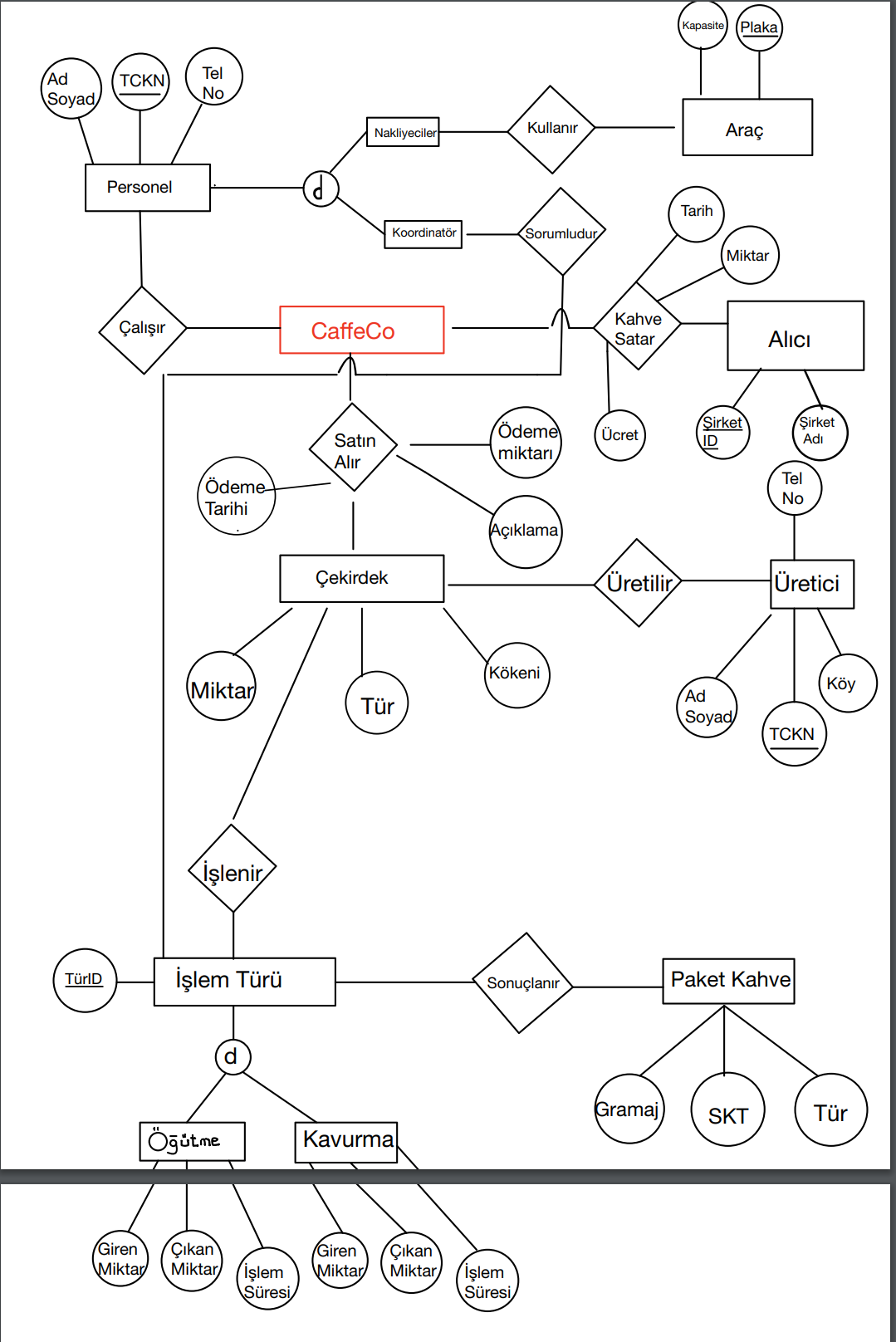
Select SQL:

"Select \* from public.\"Uretici\""

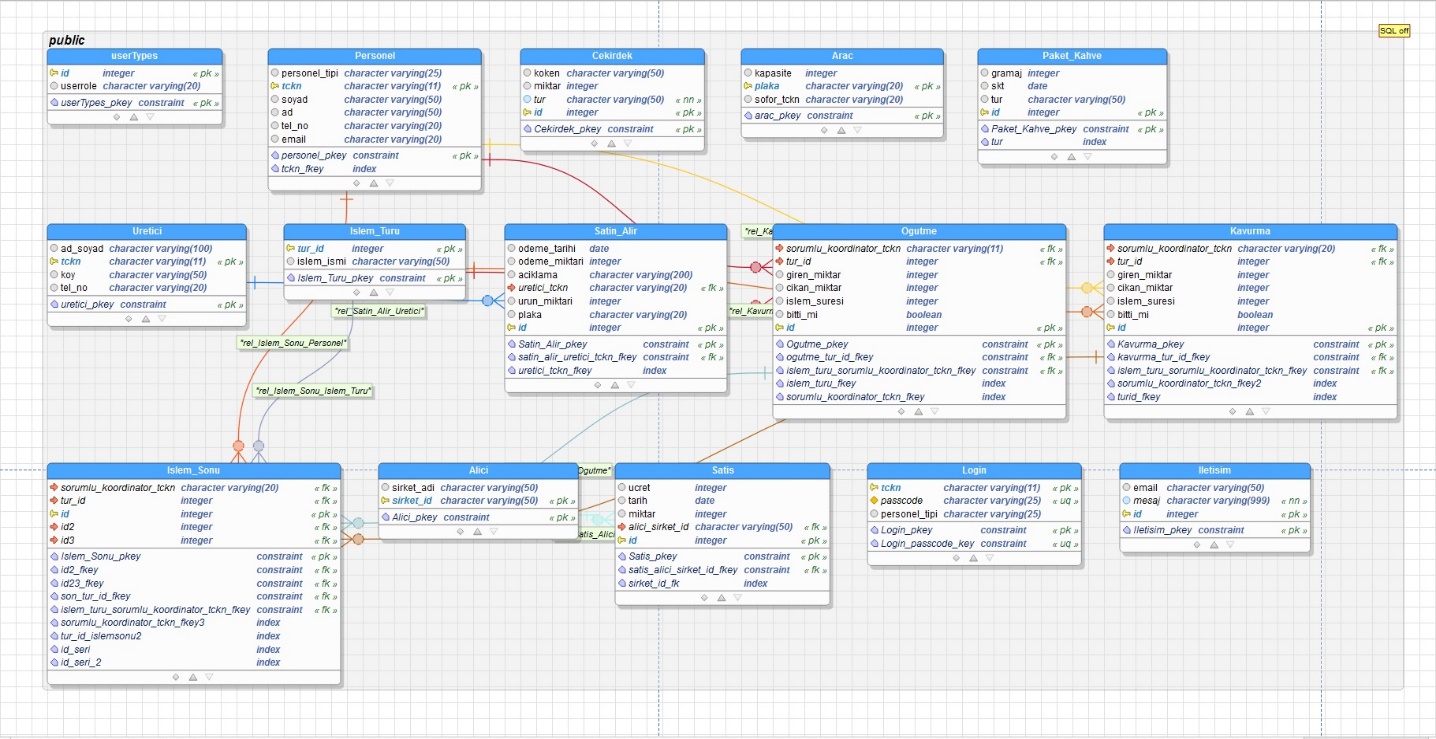


**2) Appendix**

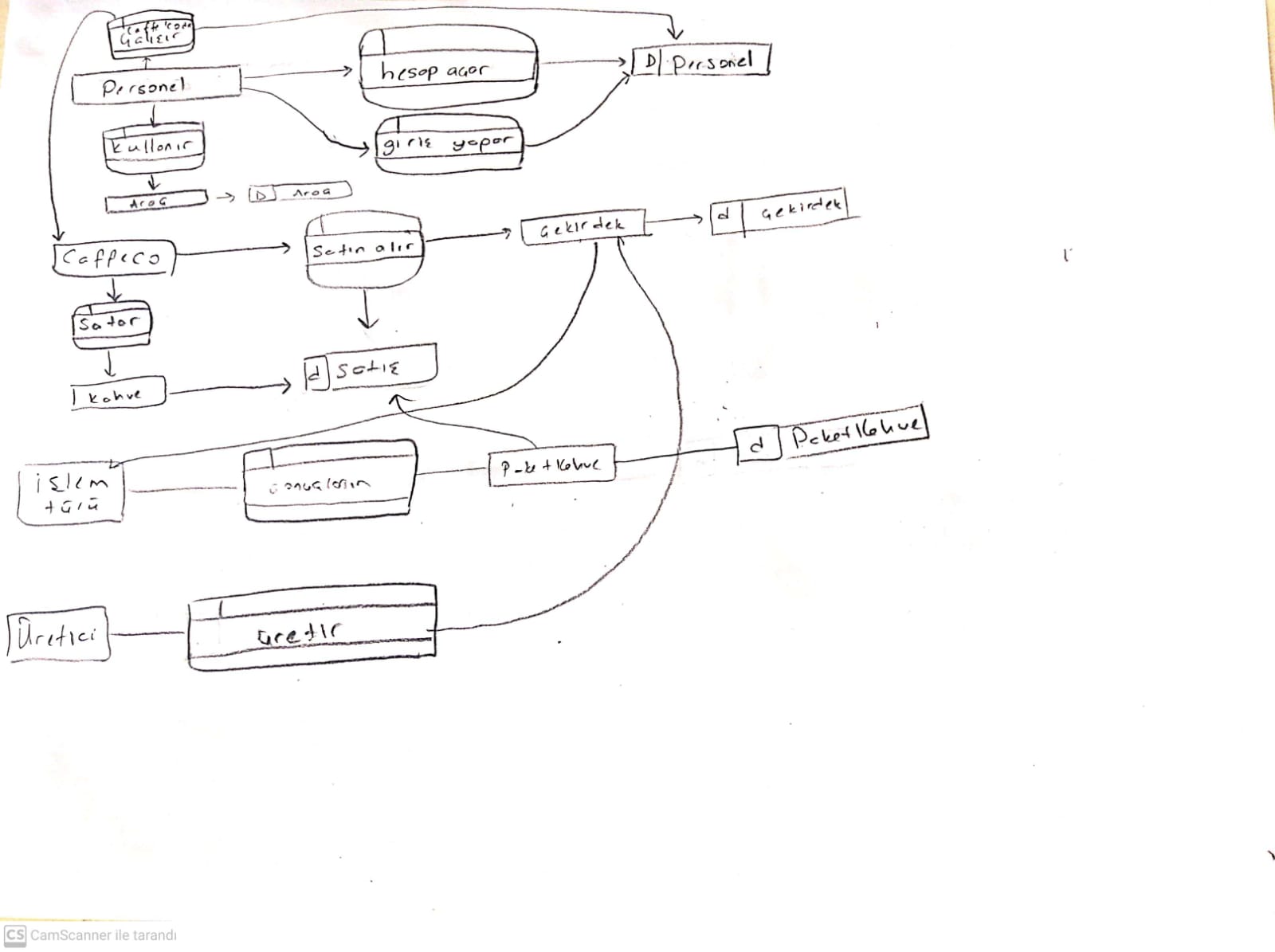
**2.1. ER Diyagramı**

****

**2.2. Relational Mapping**

****

**2.3. Data Flow Diagram (DFD)**

****

**2.4. Hipo Diagram**

Display Menu Screen

LOGIN

Register

Manage

Öğütme Data

Manage

Çekirdek

Manage Paket Kahve

Manage Kavurma List

Display

İşlem Sonu Table

Update

Register

Display

Display

Register

Update

Update

Register

Display

Update

Display

Register

Admin Home Page

Manage Üretici

Manage Satın Alım

Manage Alıcı

Update

Register

Display

Update Register

Display

Update

Register

Display

Manage İşlem Türü

Manage

Araçlar

Manage Personel

Manage Satış

Update

Register

Display

Update

Register

Display

Display Öğütme

Register

Update

Display

Kavurma

Register

Update