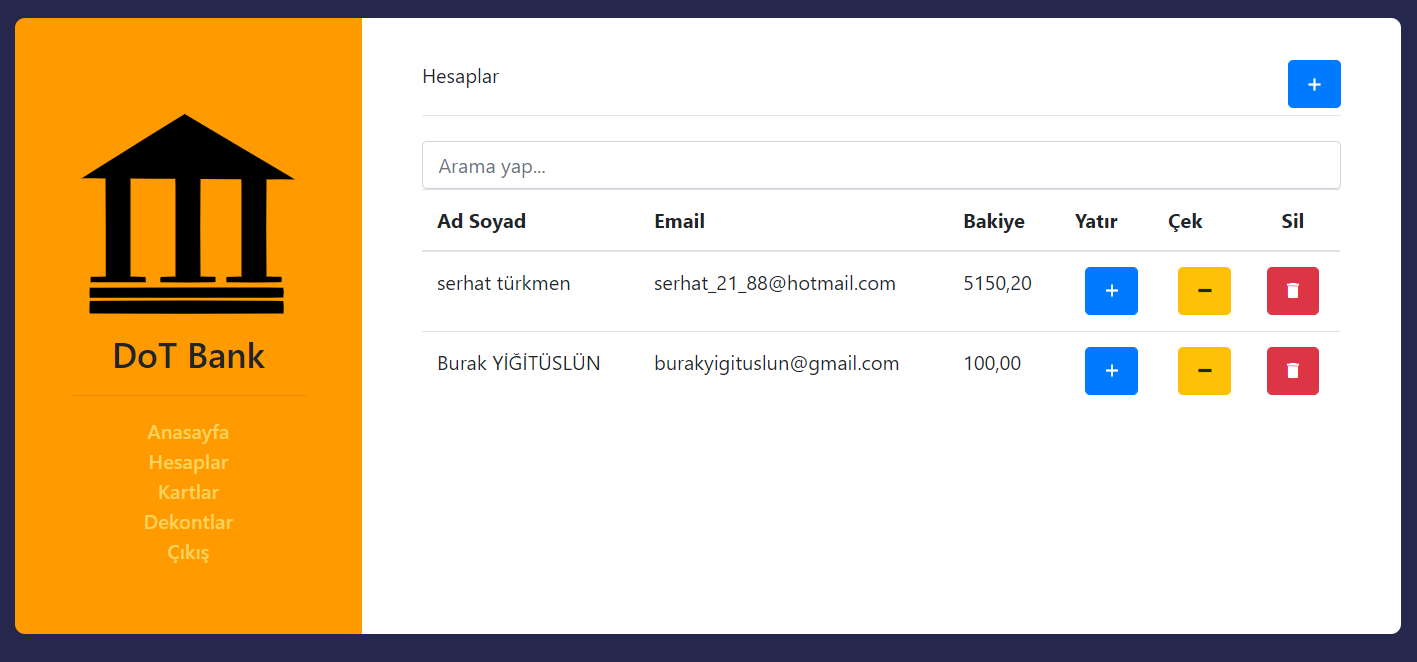
# Banka Yönetim Sistemi ve Sanal Pos Api

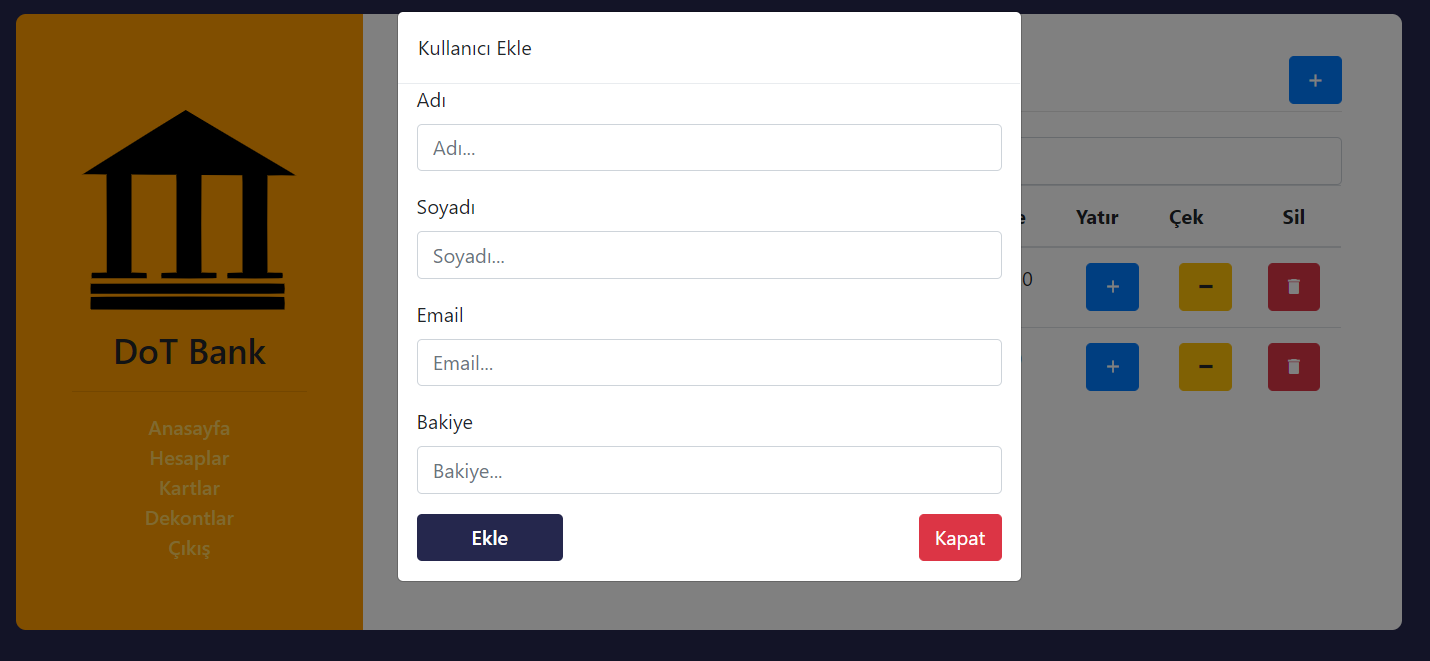
Proje Adı : Dot Bank;  
 Projede kullanılan teknoloji: Flask (Python Web Framework)  
 Projede yapılabilen işlemler ikiye ayrılmıştır.   
 Bunlar arayüz ve api’dir.  
 -> Sistem arayüzünde hesap, kart ve dekontlar bulunmaktadır.  
 -> Api desteğinde ise kart bilgileri girilerek kullanıcı doğrulama ve gelen cevaptaki karta sahip kullanıcının numarası ile ödeme alabilme mevcuttur.

Proje başka projede ödeme desteği için test edilmek üzere demo olarak yazıldığından başlagıç aşamasındadır.

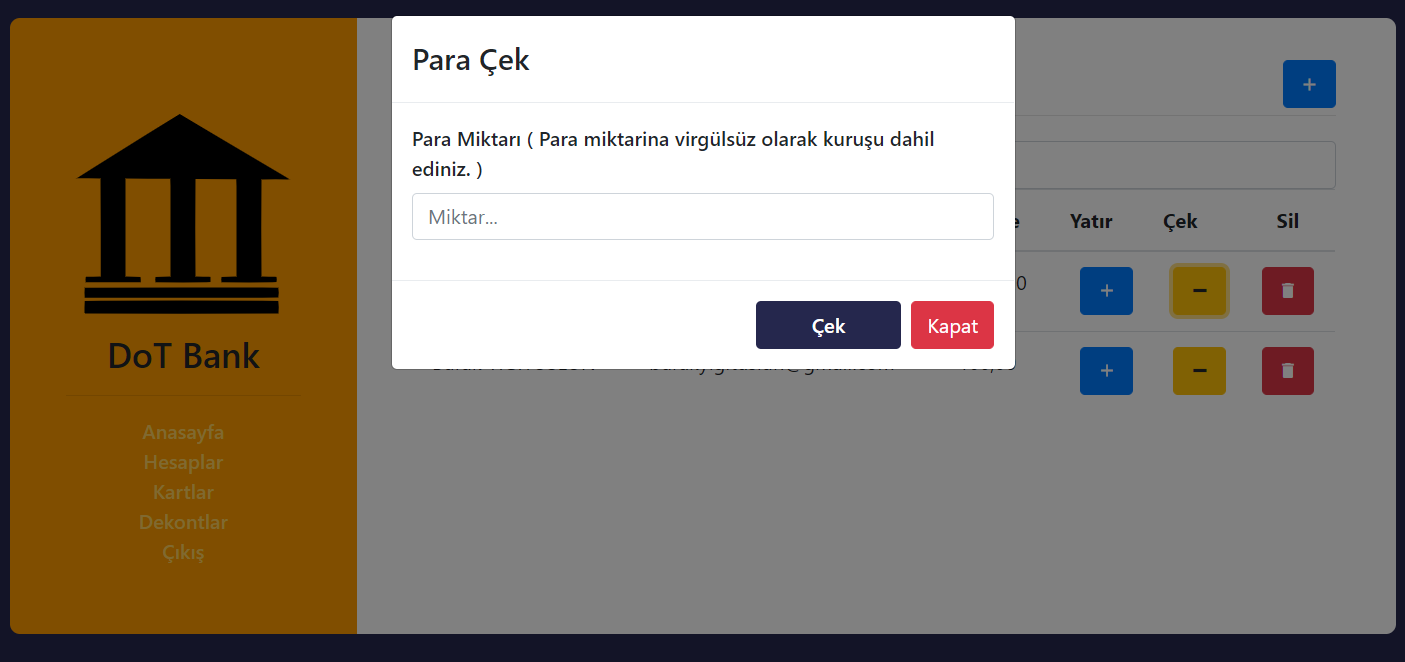
**Arayüz Ekranları**



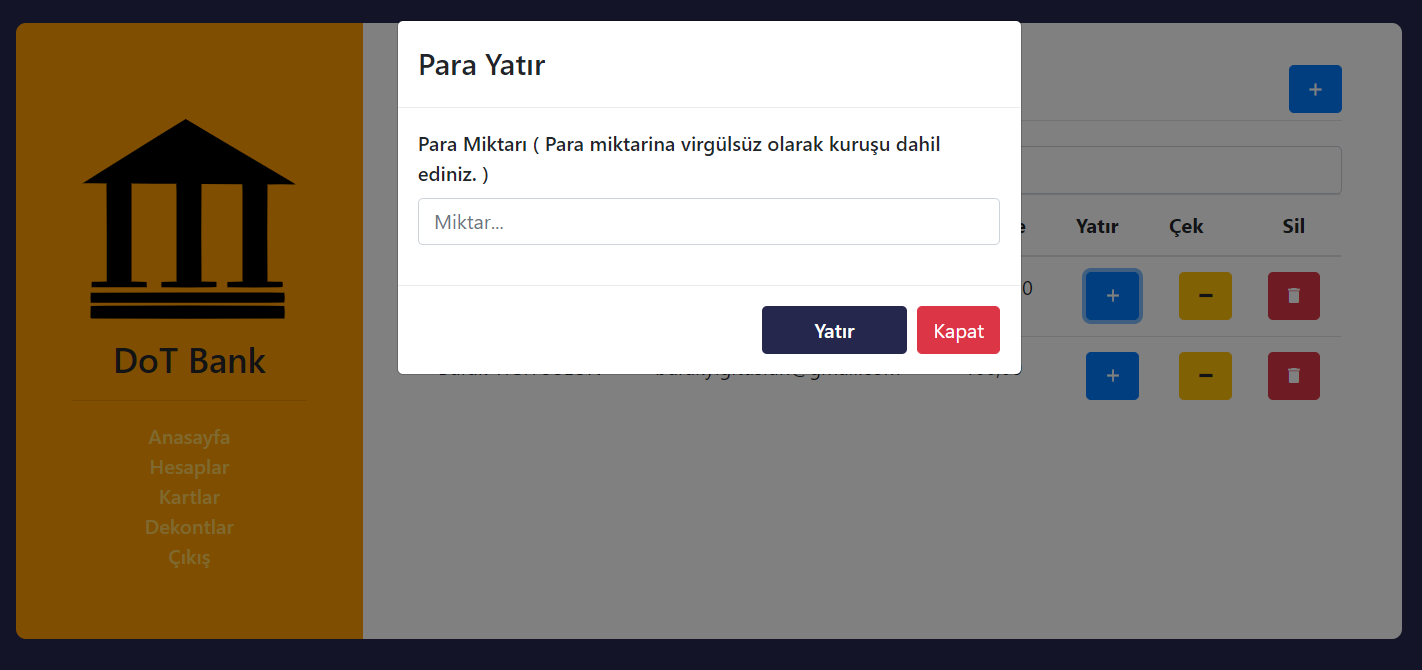
Hesaplar Sayfası



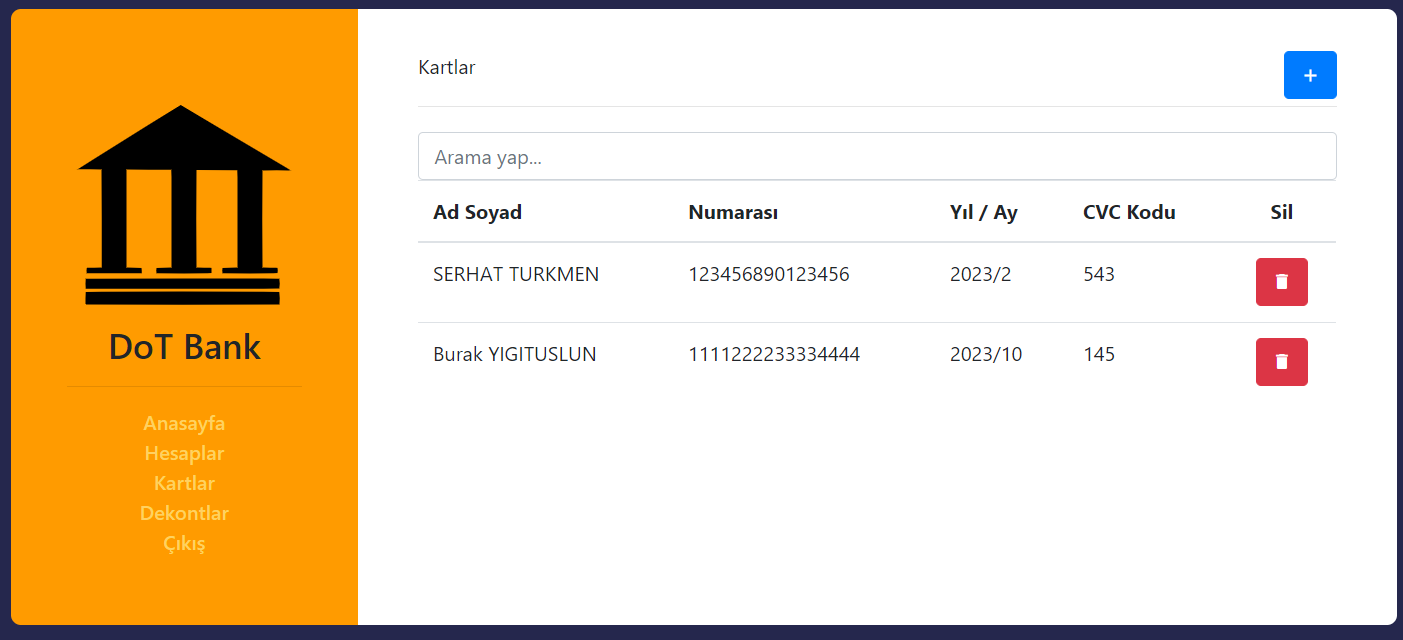
Hesaplar Sayfası -> Kullanıcı Ekleme açılır penceresi



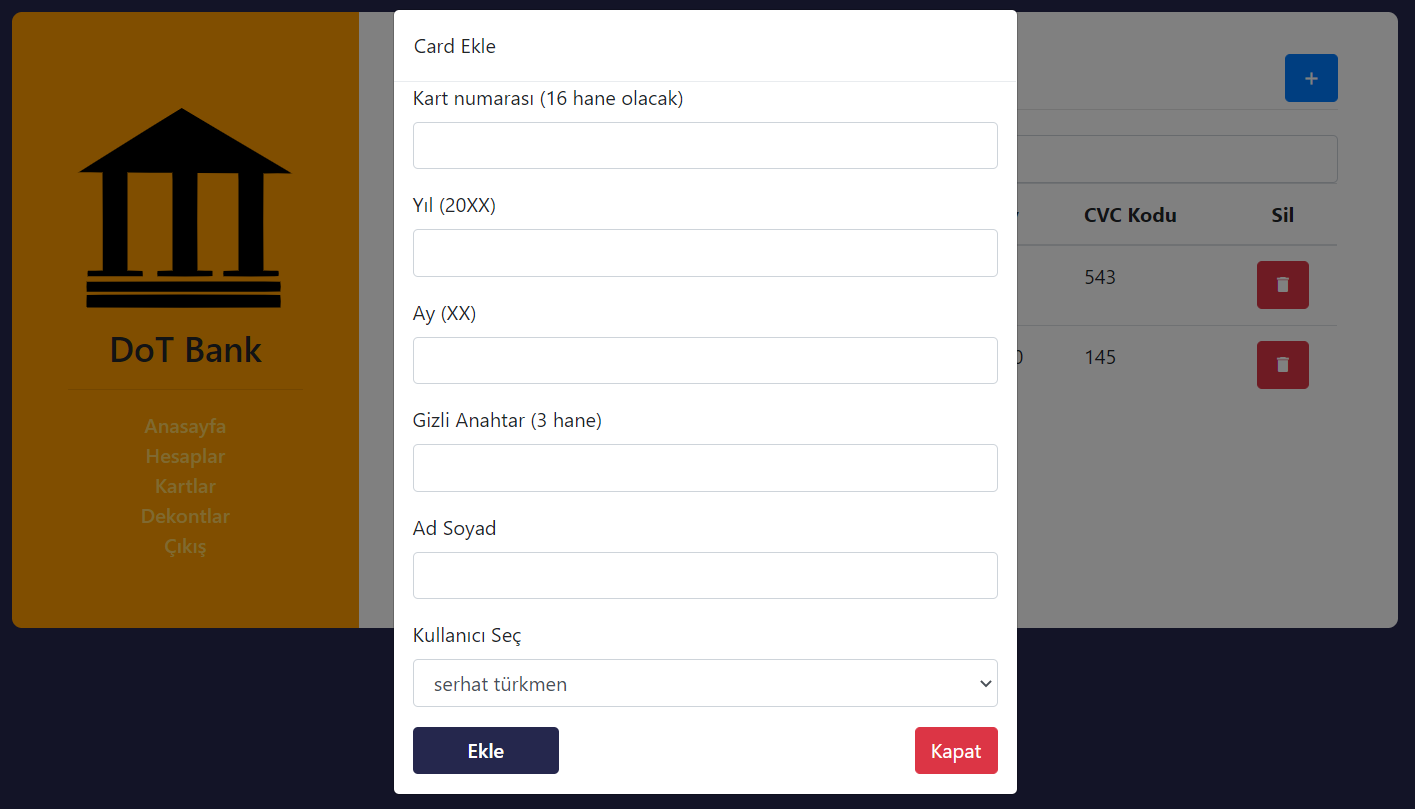
Hesaplar Sayfası -> Para Çekme açılır penceresi



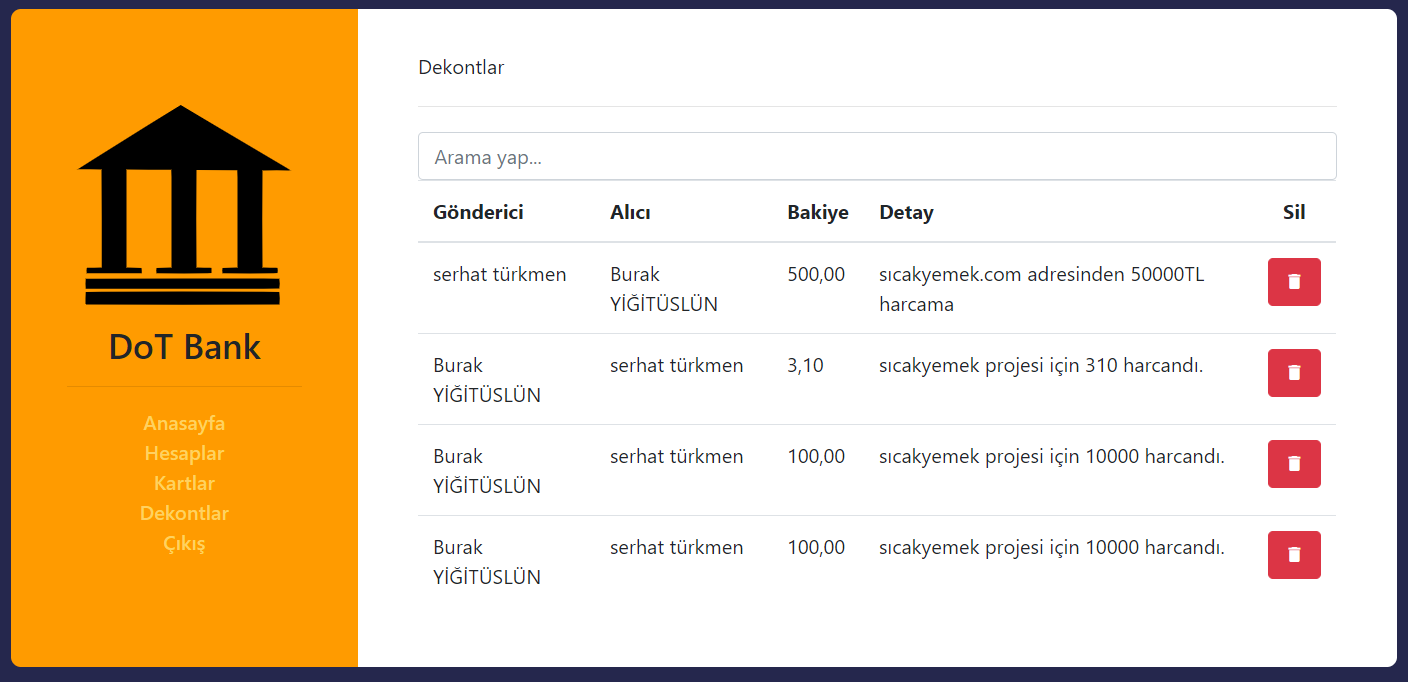
Hesaplar Sayfası -> Para Yatırma Açılır Sayfası



Kartlar Sayfası



Kartlar Sayfası -> Kart Ekle açılır penceresi



Dekontlar Sayfası

Şimdi de api kısmını biraz ele alalım. Api request(istek) ve response(cevap) üzerine kuruludur. Request’te GET veya POST işlemlerini göndererek gelen Response(Cevap) geri döndürülür.  
 Api olarak bir kart doğrulama bir de ödeme alma yönlendirmeleri tanımladım.

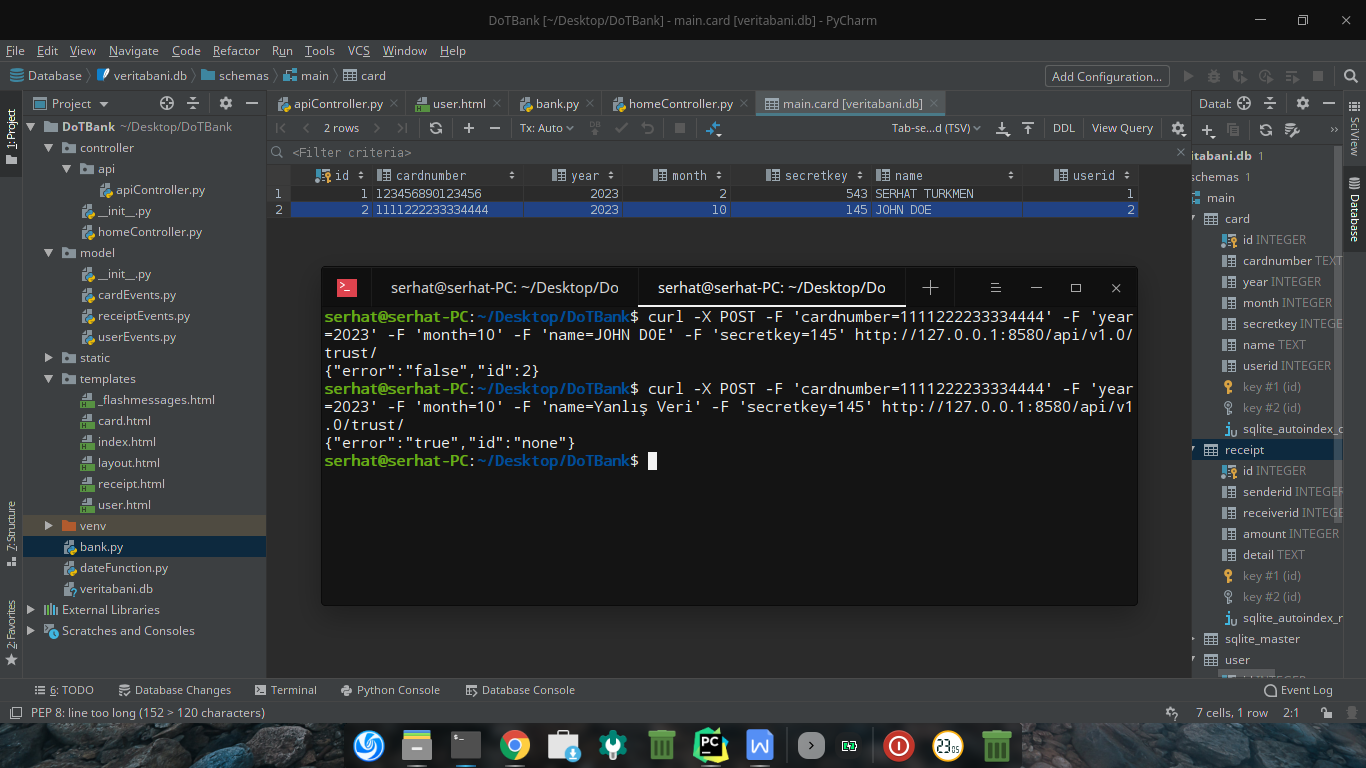
|  |  |
| --- | --- |
| Form Name Değeri | Form Value Değeri |
| cardnumber | 1111222233334444 |
| year | 2023 |
| month | 10 |
| name | JOHN DOE |
| secretkey | 145 |

Test için veritabanında kayıtlı olan bir kullanıcıyı kulanacağız.

Curl kütüphanesinin kurulu olduğu bir ortamda sistemi test edelim.

curl -X POST -F 'cardnumber=1111222233334444' -F 'year=2023' -F 'month=10' -F 'name=JOHN DOE' -F 'secretkey=145' http://127.0.0.1:8580/api/v1.0/trust/

Bur kodda bir form gönderiyoruz. Form içeriği üstteki tabloda mevcuttur. Bu form verilerini <http://127.0.0.1:8580/api/v1.0/trust/> bu adrese gönderiyoruz. Bu adres bize json formatında hata olup olmadığına ve userid yi geri döndürüyor. Bir istek atalım.



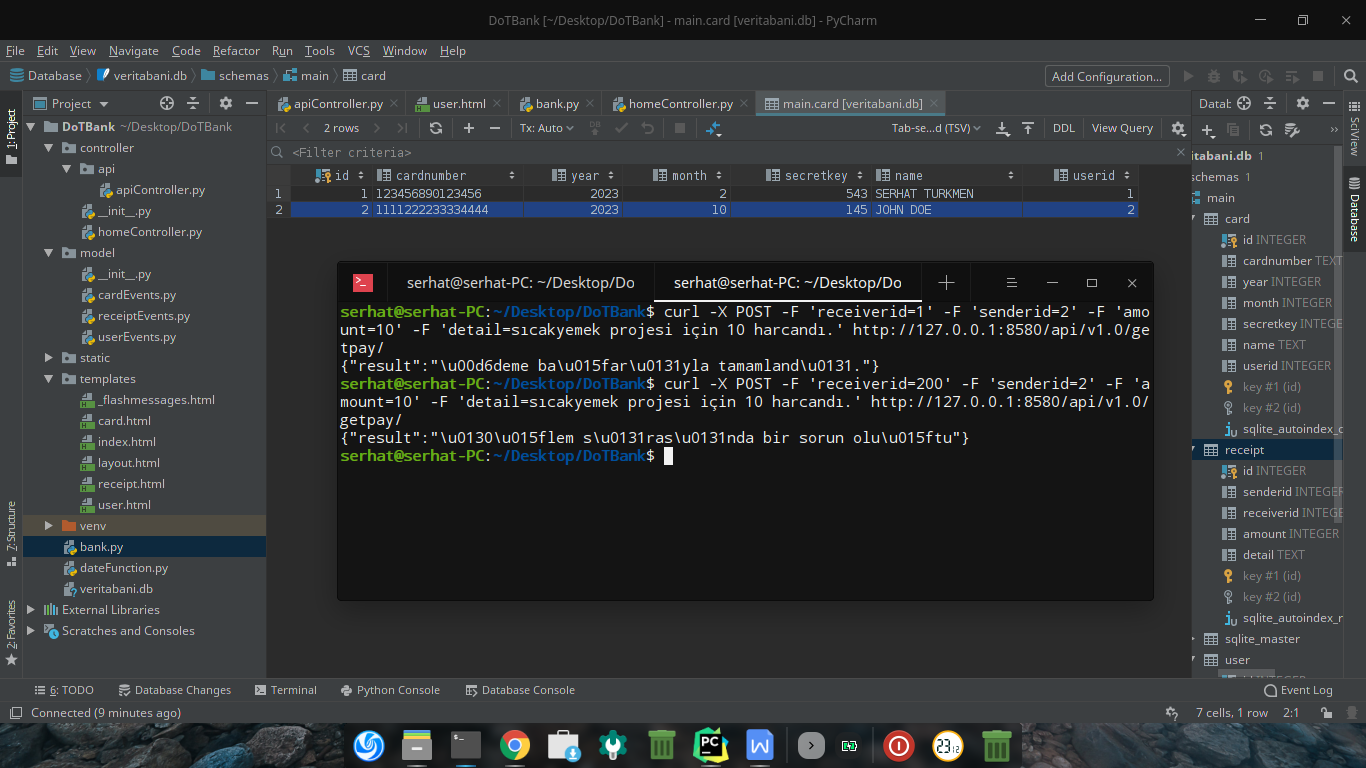
İki sorgu attım. Bunlardan ilki var olan veri üzerinden sorgu yapıldı. Bu sorgu sonucunda hatanın olmadığı ve kullanıcı id sinin de 2 olduğunu belirtti. İkinci sorguda ise bir hatalı veri söz konusu olduğu ve kullanıcı adının var olmadığını belirten bir response(cevap) verdi. Bu cevaplara göre işlem yapabilirsiniz.

Şimdi ise kart doğrulamanın başarılı bir şekilde yapıldığını varsayarak ödeme alma örneğini uygulayalım.

curl -X POST -F 'receiverid=1' -F 'senderid=2' -F 'amount=10' -F 'detail=sıcakyemek projesi için 10 harcandı.'

http://127.0.0.1:8580/api/v1.0/getpay/

Ödemede detail kısmına dekonttaki detay kısmına tekabül etmektedir. Para değeri (Amount) olarak ise virgülsüz şekilde para girişi sağlandı. Örneğin 10 girildiyse 10 kuruş olarak değerlendirilecektir. Receiverid(Alıcı id) bankada işletmenin id adresi olarak önceden tanımlı olacaktır. Ayrıca senderid(gönderici id) kart doğrulamada eriştiğimiz bir bilgi olduğundan bunu bir değişkende saklamak uygun olacaktır. İşlemi bir de test edelim.



İşlem sonucu json formatında olduğundan Türkçe karakterlerini ekranda yazdırırken bir görüntüleme sorunu yaşandı. Bu sorun api kullanırken sorun oluşturmamaktadır. İlk sonuçta “Ödeme başarıyla tamamlandı” yazıyor. Ve ikinci sonuçta “Ödeme sırasında bir sorun oluştu” yazmaktadır. Buna ek olarak statu ekledim. Eğer statu:True ise işlemin başarılı olduğunu, statu:False ise işlemin başarısız olduğunu anlayan bir geri dönüş sağlanıyor. Bu sayede kullanıcı ekranına result işlemlerin sürdürülebilmesi için ödeme onayını statu ile alabilmiş olacağız.  
  
 Sağlıcakla kalınız.