

DEBIAN SUNUCUSU ÜZERİNDE FLASK İLE POSTGRESQL VERİTABANINA KAYIT YAPAN UYGULAMA

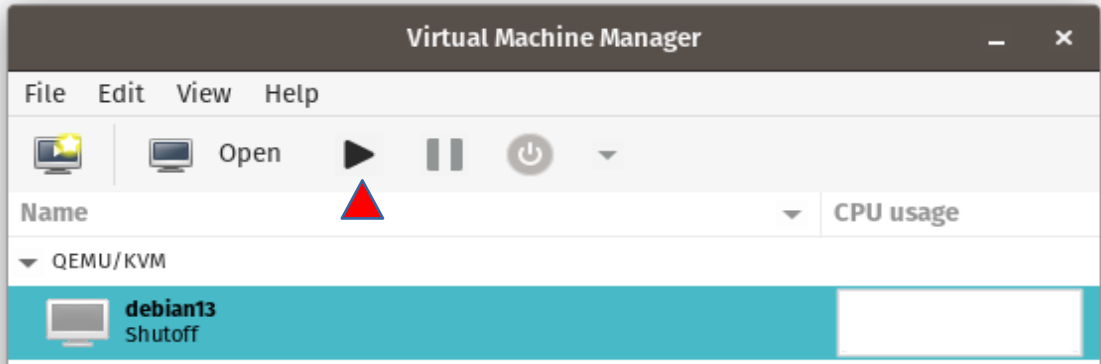
ÖNÜR DOĞAN

Gereklilikler:

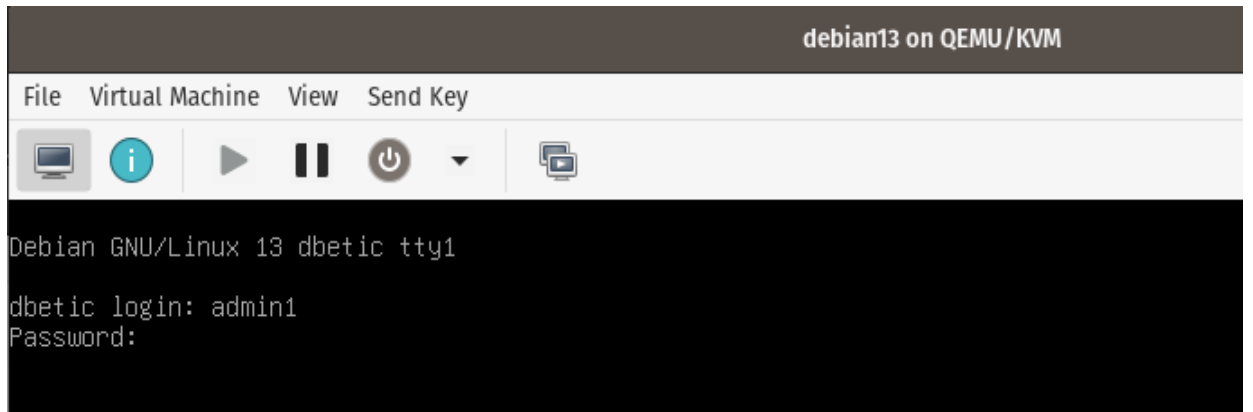
- 1. Python Virtual Enviroment:** Sunucumuzun işletim sisteminde herhangi bir değişikliğe sebep olmaması maksadıyla proje klasörümüzde .venv adında (gizli) bir klasör oluşturacağız, Linux işletim sisteminde “.” ile başlayan klasörler gizli oldukları anlamına gelir.
- 2. PostgreSQL:** Açık kaynak kodlu veritabanı sistemidir. Projemizde girilen veya kullanılacak bazı verilerin kaydedilmesi ve çağırılması maksadıyla kullanacağız.
- 3. Flask:** Python ile yazılmış hafif bir web uygulama çatısıdır (framework). Yani HTML formlarını, sayfaları, veritabanı bağlantılarını, kullanıcı girişlerini yönetmemizi sağlar biz PostgreSQL’e veri kaydetmek için kullanacağız.

ADIM ADIM KURULUM

- 1. Öncelikle linux sistemimizde VMM (Virtual Machine Manager) ile oluşturduğumuz sanal Debian tabanlı sistemimizi başlatacağız.**



- 2. Açılan debian terminaline kendi kullanıcı adımızı ve parolamızı girerek sistemi açıyoruz.**



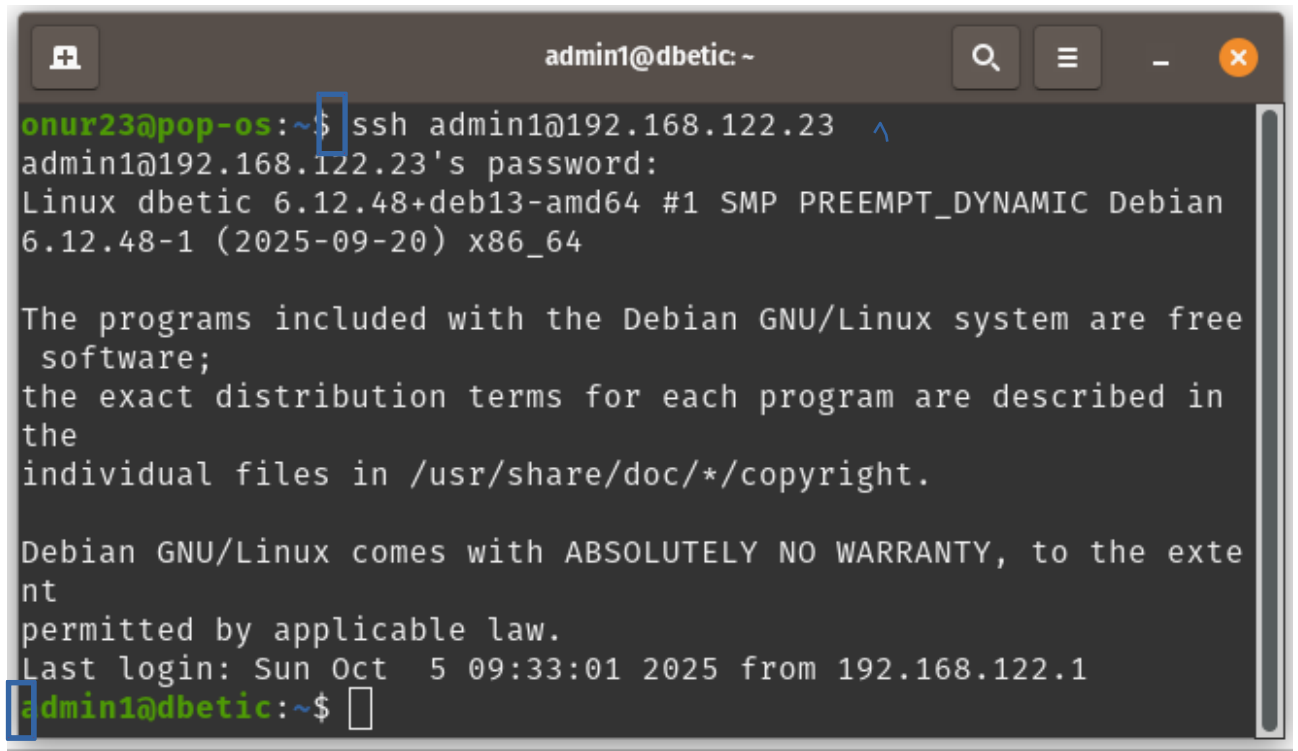
3. Ana işletim sistemimizin terminalini açarak SSH ile Debian sunucumuza bağlanarak tüm işlemlerimizi kendi terminalimizden tamamlayacağız.SSH bağlantısı uzaktan sunucunun terminaline bağlanmak için kullanılır.

Önemli not: SSH Client çoğu linux dağıtımına yüklü olarak gelir. SSH server ise sonradan yüklenir, eğer sunucu tarafınızda SSH server yok ise

```
sudo apt install openssh-server  
sudo systemctl enable ssh  
sudo systemctl start ssh
```

Komutlarıyla yükleyebilirsiniz yükleme komutları linux dağıtımına göre değişiklik gösterebilir.

Karşıya bağlanma komutumuz: `ssh kullanıcıadı@localipaddress`



```
admin1@dbetic: ~  
onur23@pop-os:~$ ssh admin1@192.168.122.23  
admin1@192.168.122.23's password:  
Linux dbetic 6.12.48+deb13-amd64 #1 SMP PREEMPT_DYNAMIC Debian  
6.12.48-1 (2025-09-20) x86_64  
  
The programs included with the Debian GNU/Linux system are free  
software;  
the exact distribution terms for each program are described in  
the  
individual files in /usr/share/doc/*/copyright.  
  
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the exte  
nt  
permitted by applicable law.  
Last login: Sun Oct 5 09:33:01 2025 from 192.168.122.1  
admin1@dbetic:~$
```

Eğer sunucumuzun ip adresini bilmiyorsak sunucumuzun terminalinden ip addr komutunu çalıştırmamız yeterli 122. ile başlayan adres işletim sistemimiz ile sunucumuz arasında kullanılacak olan adrestir.

4. Bağlantımız sağlandığına göre devam edelim. Şimdi sunucumuza Postgresql veritabanını ve eklentilerini yüklememiz gerekiyor. Yükleme komutu

sudo apt install -y postgresql postgresql-contrib

```
admin1@dbetic: ~  
permitted by applicable law.  
Last login: Sun Oct 5 09:33:01 2025 from 192.168.122.1  
admin1@dbetic:~$ sudo apt install -y postgresql postgresql-contrib  
[sudo] password for admin1:  
Note, selecting 'postgresql' instead of 'postgresql-contrib'  
Installing:  
  postgresql  
  
Installing dependencies:  
  libcommon-sense-perl postgresql-17
```

Hemen ardından veritabanımızın aktif olup olmadığını kontrol ediyoruz

Komutu: **sudo systemctl status postgresql**

```
admin1@dbetic: ~  
● postgresql.service - PostgreSQL RDBMS  
   Loaded: loaded (/usr/lib/systemd/system/postgresql.service; enabled; preset: enabled)  
   Active: active (exited) since Sun 2025-10-05 11:53:17 EDT; 54s ago  
  Invocation: 90c8832b403a40a7b1e2bd353fb00748  
    Main PID: 2592 (code=exited, status=0/SUCCESS)  
   Mem peak: 1.8M  
      CPU: 10ms  
  
Oct 05 11:53:17 dbetic systemd[1]: Starting postgresql.service - PostgreSQL RDBMS...  
Oct 05 11:53:17 dbetic systemd[1]: Finished postgresql.service - PostgreSQL RDBMS.  
~  
~  
lines 1-10/10 (END)
```

Önemli not: Eğer sizin veritabanınızda active kısmı farklıysa yani henüz çalışmıyorsa **sudo systemctl start postgresql** komutuyla aktif edebilir, **sudo systemctl enable postgresql** komutuyla sistem her açıldığında otomatik başlamasını sağlayabilirsiniz.

5.Şimdi veritabanımıza girerek yeni bir kullanıcı oluşturacağız, ardından uygulamamızda kullanacağımız veritabanını ve tabloları oluşturacağız.

PostgreSQL giriş komutumuz:

sudo -i -u postgres Sistemdeki postgres adlı kullanıcı hesabına geç” demektir. postgres hesabı veritabanının root hesabıdır.

psql ise veritabanının terminaline geçiren komuttur.

\l -- Tüm veritabanlarını listele

\du -- Kullanıcıları listele

\q -- Çık

```
admin1@dbetic: ~  
admin1@dbetic:~$ sudo -i -u postgres  
postgres@dbetic:~$ psql  
psql (17.6 (Debian 17.6-0+deb13u1))  
Type "help" for help.  
  
postgres=#
```

Şimdi yapacağımız uygulama için yeni bir kullanıcı, veritabanı ve tabloları ekleyeceğiz. Sırasıyla şu komutları çalıştıracğız.

CREATE USER admin1 WITH PASSWORD '**';** \\ Yeni kullanıcı oluştur

CREATE DATABASE anketdb OWNER admin1; \\ admin1 Kullanıcısı için anketdb veritabanını oluştur.

GRANT ALL PRIVILEGES ON DATABASE anketdb TO admin1; \\admin1 için tüm yetkileri ver

\c anketdb \\databasesinin içine gir

**CREATE TABLE anketler (
id SERIAL PRIMARY KEY,
full_name VARCHAR(100) NOT NULL,
email VARCHAR(120) NOT NULL,
hizmet_kalite VARCHAR(50) NOT NULL,
personel_ilgi VARCHAR(50) NOT NULL,
fiyat_memnuniyet VARCHAR(50) NOT NULL,
tekrar_tercih VARCHAR(50) NOT NULL,
yorum TEXT,
olusturulma TIMESTAMP NOT NULL DEFAULT now()
);** \\belirtilen tabloları oluştur.

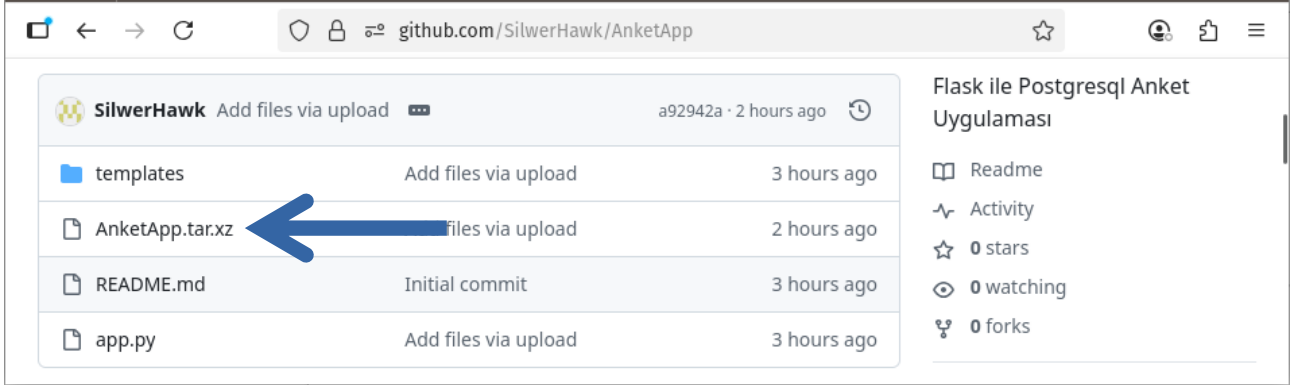
\d anketler \\ Anketler veritabanında hangi sütunlar var listele

Sonucun aşağıdaki şekilde gelmesi gerekiyor

```
admin1@dbetic: ~  
Table "public.anketler"  
+-----+-----+-----+-----+-----+  
| Column | Type | Collation | Nullable | Default |  
+-----+-----+-----+-----+-----+  
| id | integer | | not null | nextval('anketler_id_seq'::regclass) |  
| full_name | character varying(100) | | not null | |  
| email | character varying(120) | | not null | |  
| hizmet_kalite | character varying(50) | | not null | |  
| personel_ilgi | character varying(50) | | not null | |  
| fiyat_memnuniyet | character varying(50) | | not null | |  
| tekrar_tercih | character varying(50) | | not null | |  
| yorum | text | | | |  
| olusturulma | timestamp without time zone | | not null | now() |  
+-----+-----+-----+-----+-----+  
Indexes:  
"anketler_pkey" PRIMARY KEY, btree (id)  
(END)
```

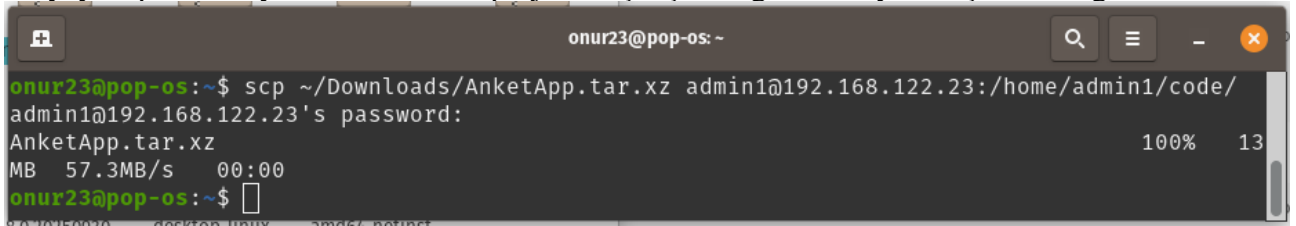
Ekranı gördüysek şuan için veritabanı kısmıyla işimiz tamamlandı demektir. Ancak veritabanı adımızı kullanıcıyı ve şifresini bir kenara not alalım bu daha sonra işimize yarayacak.

5. Şimdi kullanacağımız uygulamayı github üzerinden tar.gz olarak indirip sunucumuza atıp projemizin klasörüne yükleyeceğiz.

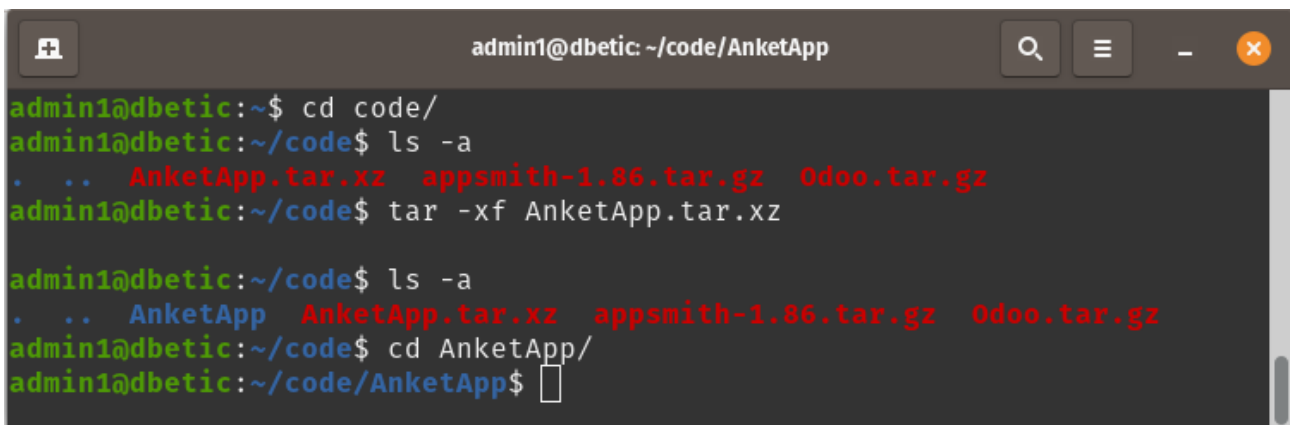


Hedefe tıklayınca başka bir klasör seçilmediyse ana makinemizin Downloads klasörüne indirecektir.

6. Şimdi exit komutuyla PostgreSQL ve SSH bağlantısından çıkacağız, ardından indirdiğimiz dosyayı scp komutuyla sunucumuzda projemizi çalıştıracığımız dosyanın içine atacağız.



Tekrar SSH bağlantısı ile sunucumuza bağlanacağız ve sıkıştırılmış dosyamızı tar -xf komutu ile dışarı çıkaracağız.

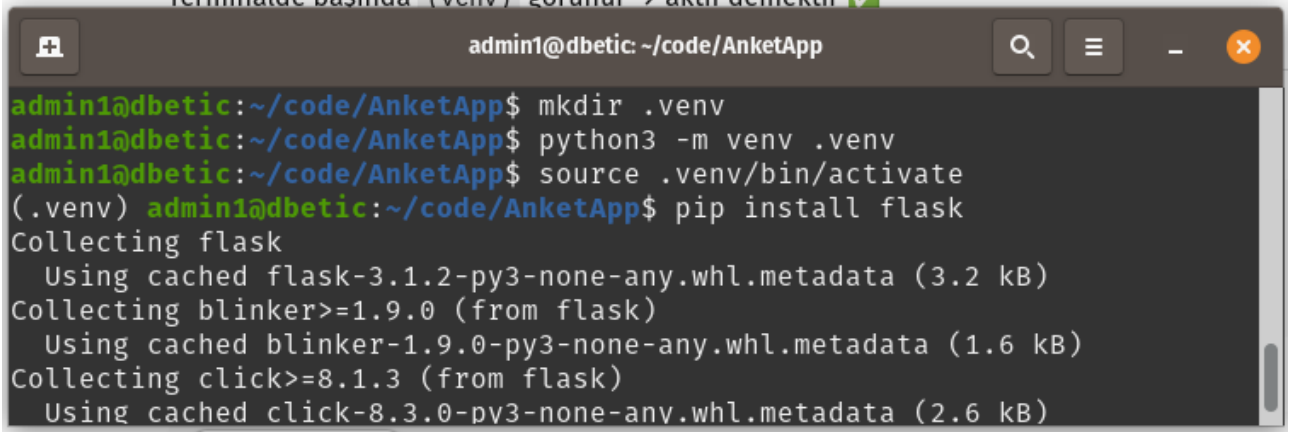


7. Dosyamızın içine girerek mkdir komutuyla .venv adında bir klasör oluşturacağız, bu klasör . ile başladığı için gizlidir. Linuxta . ile başlayan klasörler gizli olduğunu belirtir. Oluşturduğumuz bu klasörün içine Python Virtual Enviroment ve Flask Web geliştirme eklentisini kuracağız.

`mkdir .venv` - Virtual enviromenti kuracağımız klasörü oluştur

`python3 -m venv .venv` - .venv klasörüne virtual enviroment kurrulumu yapıyoruz.

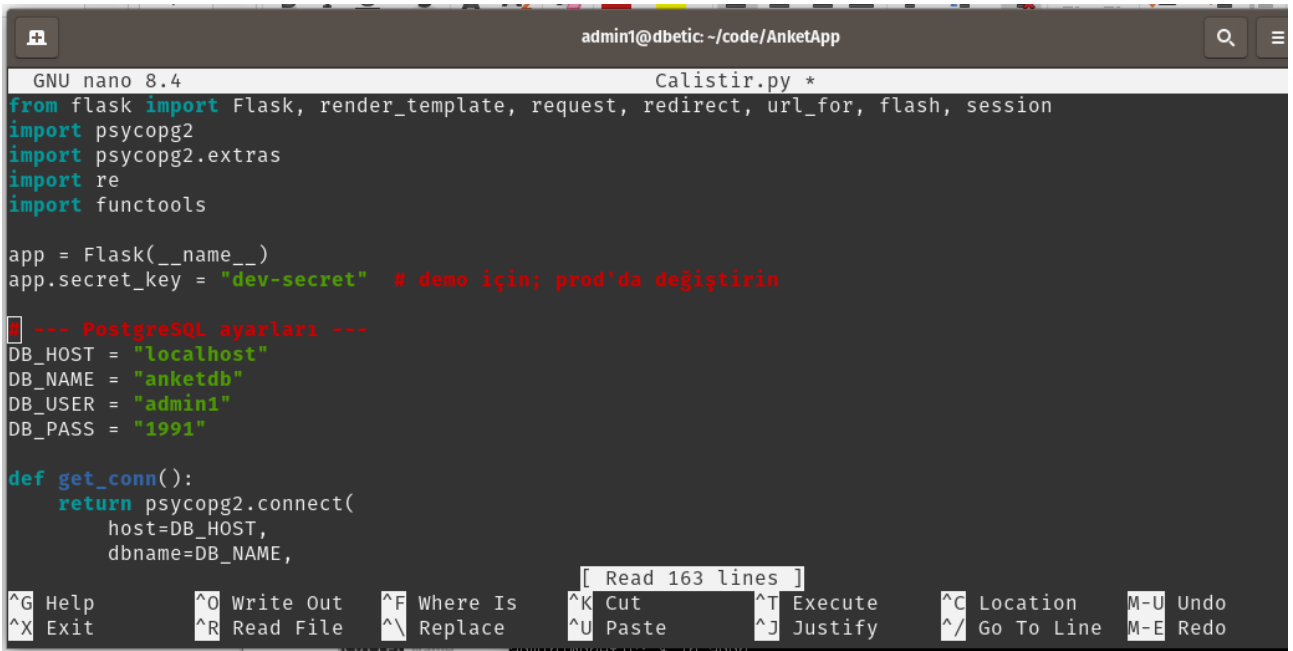
`Source .venv/bin/activate` - sanal ortamı aktif et



```
admin1@dbetic: ~/code/AnketApp
admin1@dbetic:~/code/AnketApp$ mkdir .venv
admin1@dbetic:~/code/AnketApp$ python3 -m venv .venv
admin1@dbetic:~/code/AnketApp$ source .venv/bin/activate
(.venv) admin1@dbetic:~/code/AnketApp$ pip install flask
Collecting flask
  Using cached flask-3.1.2-py3-none-any.whl.metadata (3.2 kB)
Collecting blinker>=1.9.0 (from flask)
  Using cached blinker-1.9.0-py3-none-any.whl.metadata (1.6 kB)
Collecting click>=8.1.3 (from flask)
  Using cached click-8.3.0-py3-none-any.whl.metadata (2.6 kB)
```

Bu işlemleri tamamladıktan sonra deactivate komutu ile sanal ortamdan çıkabiliriz.

8. Projemizin klaörüne geri döndük şimdi projemizin database ayarlarını yapacağız. Bu ayarlar app.py dosyasının içinde. Komut satırımıza nano app.py yazıyoruz. Bu komut her türlü dosyayı metin editörü ile açmaya yarar. Açılan dosyadan veritabanını kendi kullanıcı adımızı veritabanı adımızı ve şifremizi yazacağız. Veritabanı kendi bilgisayarımızda çalıştığı için Localhost olarak kalacak.



```
GNU nano 8.4 Calistir.py *
from flask import Flask, render_template, request, redirect, url_for, flash, session
import psycopg2
import psycopg2.extras
import re
import functools

app = Flask(__name__)
app.secret_key = "dev-secret" # demo için; prod'da değiştirin

--- PostgreSQL ayarları ---
DB_HOST = "localhost"
DB_NAME = "anketdb"
DB_USER = "admin1"
DB_PASS = "1991"

def get_conn():
    return psycopg2.connect(
        host=DB_HOST,
        dbname=DB_NAME,
```

C+X'e basarak kaydedip çıkıyoruz.

9. Tüm işlemlerimiz bitti şimdi tekrar source `.venv/bin/activate` komutuyla `.venv` moduna geçeceğiz ardından `app.py` dosyamızı bu modda başlatacağız.

Önemli not: burada `import psycpg2` hatası alırsanız `.venv` modundayken `pip install psycpg2-binary` yazarak modülü yüklememiz gerekmektedir.

```
admin1@dbetic: ~/code/AnketApp
drwxrwxr-x 5 admin1 admin1 4096 Oct  5 13:13 .venv
(.venv) admin1@dbetic:~/code/AnketApp$ python3 app.py
Traceback (most recent call last):
  File "/home/admin1/code/AnketApp/app.py", line 2, in <module>
    import psycpg2
ModuleNotFoundError: No module named 'psycpg2'
(.venv) admin1@dbetic:~/code/AnketApp$ pip install psycpg2-binary
Collecting psycpg2-binary
  Using cached psycpg2_binary-2.9.10-cp313-cp313-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (4.9 kB)
Using cached psycpg2_binary-2.9.10-cp313-cp313-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (3.0 MB)
Installing collected packages: psycpg2-binary
Successfully installed psycpg2-binary-2.9.10
(.venv) admin1@dbetic:~/code/AnketApp$ python3 app.py
* Serving Flask app 'app'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on all addresses (0.0.0.0)
* Running on http://127.0.0.1:5000
* Running on http://192.168.122.23:5000
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 139-216-018
```

Artık sistemimize `http://127.0.0.1:5000` adresinden (Debian üzerinden)
Ana bilgisayarımızdan ise `http://192.168.122.25:5000` adresinden ulaşılabilir olduğu yazıyor.

Ana bilgisayarımızdan `http://192.168.122.25:5000` açıp deneyelim.

Hizmet Memnuniyeti Anketi

Kişisel bilgilerinizi girin

Onur DOĞAN

onurdgn23@gmail.com

Hizmet değerlendirme

1) Hizmet kalitesinden memnun musunuz?

Çok memnunum

2) Personelin ilgisini nasıl değerlendirirsiniz?

Harika

3) Fiyat/performans açısından memnun musunuz?

Çok memnunum

4) Hizmetimizi tekrar tercih eder misiniz?

Evet

[Deneme](#)

[Geri](#) [Gönder](#)

Anket Uygulaması Anket İstatistik Admin Girişi

Teşekkür ederiz ❤️

Geri bildiriminiz bizim için çok değerli.

[Anasayfaya dön](#)

Anket Uygulaması Anket İstatistik Admin [Çıkış](#)

Giriş başarılı.

Admin Paneli

ID	Ad Soyad	E-posta	Hizmet Kalitesi	Personel İlgisi	Fiya Men
1	Onur DOĞAN	onurdgn23@gmail.com	Çok memnunum	Harika	Çok men

Onur DOĞAN