Pyhton un path e tanımlı olması gerekiyor. Biz bunu pyhton yüklerken (Bir yeri tiklemiştik) yapmıştık.

Pyhton da flask kurulumu yaptık. Flask yükleme kodu:

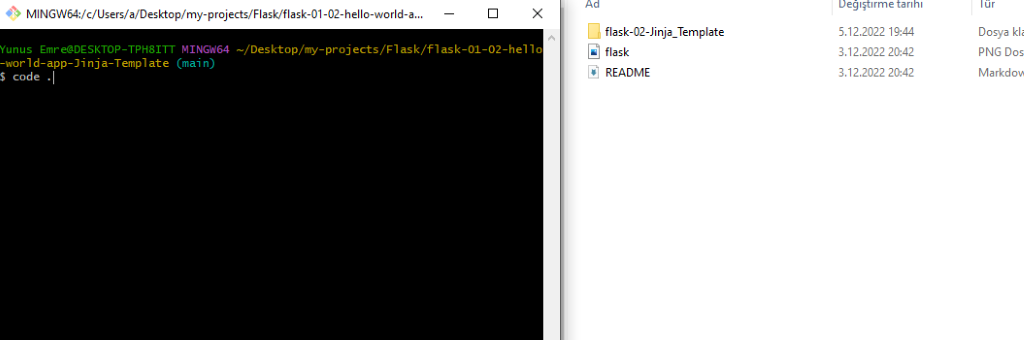
pip3 install flask

Yükledikten sonra yazan açıklamalarda güncelleme için kodu yazar. Orada yazan kodu girerek güncelliyoruz.

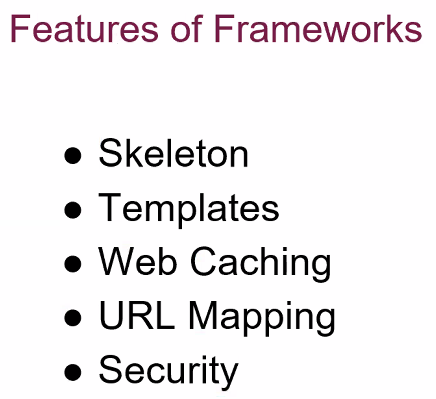
Şahsi Github repomuzda public bir My-projects izimli repo oluşturup bunu pc mize klonladık.

Yereldeki clarus-way reposunu git pull ile güncelledik ve içinde Pyhton-in-class hands on içindeki Flask klasörünü yereldeki my-project repomuzun içine kopyaladık. Daha sonra Flask klasörünü VSCode açtık:

Bir dosyayı VSCode da açmak için dosya içine girip sağ tıklayıp git bash here dedikten sonra "code ." Kodunu giriyoruz ve VSCode da açıyor:



Her web uygulamalarının geliştirilmesi için bir takım framework ler gereklidir. Flask de bir framework tür. İskelet yapıları olarak düşünmemiz gerekir. Sıfırdan bir iskelet kurmaktansa kurulu iskeletleri alıp onlar üzerinde çalışma yapıyoruz.



Frameworklerin özellikleri;

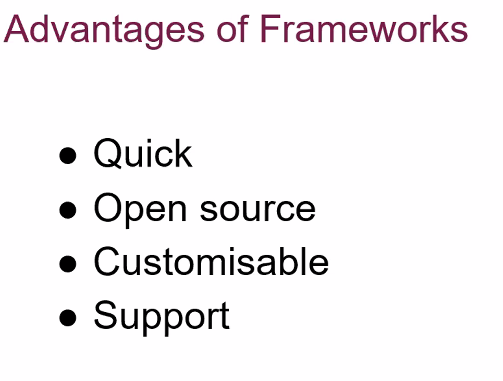
İskelet; bir web sitesi veya uygulama için hızlı ayarlanmış ana sütunları ifade der. Full stack ekibi bunu alır ve kendileri özel çerçeveler ve yapılar oluştururlar.

Şablonlar; bir sitenin hızlı bir şekilde hazırlanmasını sağlayana önceden tasarlanmış şablonlar.

Front-end kısmının daha hızlı yüklenmesini sağlamak ve back-end i de yormamak için çeşitli bilgilier cash ler (depolar)

URL Mapping; arama motoru optimizasyonuna yardımcı olaracak, arama motorları dizilimine kolaycak eklenmesini sağlaar.

Güvenlik açıklarına karşı siteleri koruyacak gereksinimlere sahiptir framewokler.



Frameworklerin avantajları:

En önemli avantajı hızlı olması. Tekrar tekrar kod yazma ihtiyacını ortadan kaldırıyor.

Serbestçe ve ücretsiz kullanılabilirler.

Ana iskeleti alıp üzerine kendine özgü bir iskelet oluştrulabilir.

Açık kaynak olduğu için birbirlerine hızlı bir şekilde yardımcı olan networkleri vardır.

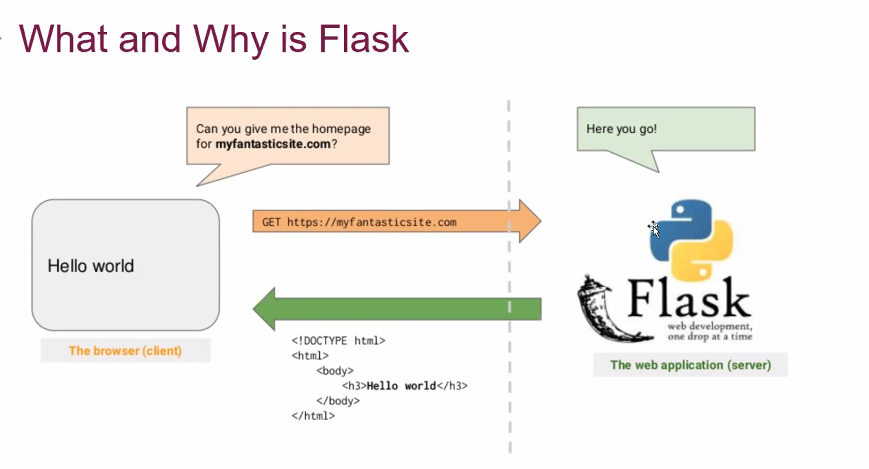
Neden  Flask?

Kullanımı daha basittir. Bir micro frameworktür. Kendinize bir araba tasarlıyorsunuz, ilk etapta motor şanzıman vites tekerlek gibi 4 ana parçaya ihtyiacınız var. Bir dükkana gidip önce bunları aldınız ve sonra ihtiyaç duydukça gidip ayna, kapı vs alıyorsunuz. Flask budur.

Ancak Django öyle değildir. İhtiyacın olan olmayan ne varsa en başta verir, bütün kütüphaneleri yükler. Bu da maliyet demektir.

Cloud engineer ların en önemli görevlerinden birisi resourceları etkin kullanarak maliyeti düşürmektir.

Biz devopsçular olarak flask te kod yazmayı bilmeyecez ancak önümüze gelen bir kodun Flask te mi yada Django da mı yazıldığını anlamamız gerekir.



Flaskte biz kod yazmayacağız, yazılan kodları anlayacak şekilde öğrenmemiz gerekir.

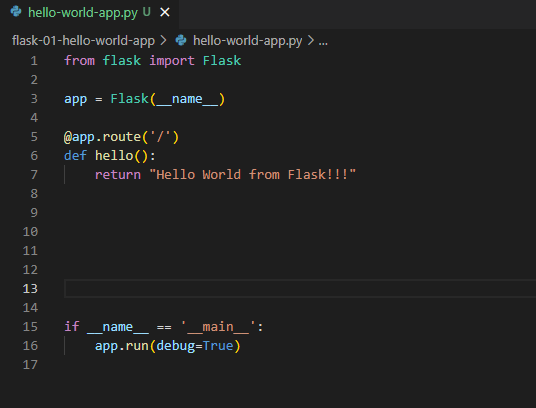
İlk başta şu kodları yazıyoruz: (Flask yazma farzları) ilki flask kütüphanesini çeker. Sonra app diye bir obje oluşturuyor. Ve en sonunda if ile bir kontrol mekanizması işletiyor. Eğer doğruysa çalıştır diyor:

from flask import Flask

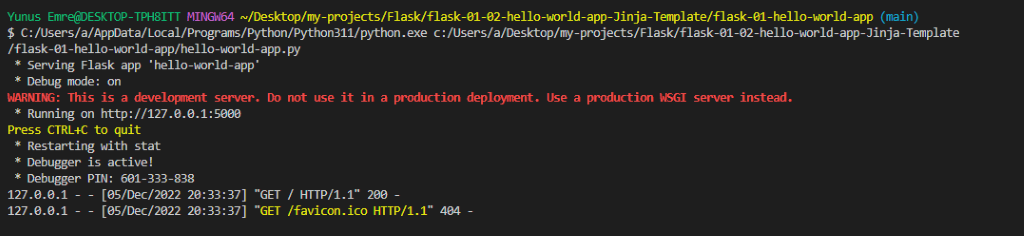
app = Flask(\_\_name\_\_)

if \_\_name\_\_ == '\_\_main\_\_':

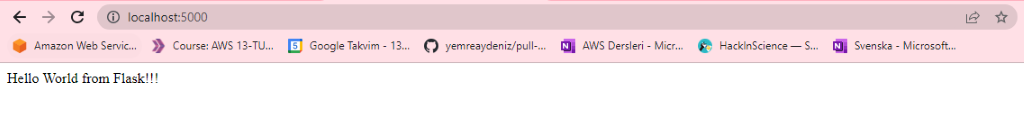
    app.run(debug=True)



Bu kodarı yazıp save edip çalıştırıyoruz. Çalıştırdıktan sonra terminalde çalıştığını görüyoruz:



Sonrasında web tarayıcısına gidip adres çubuğuna localhost:5000 yazıyoruz.





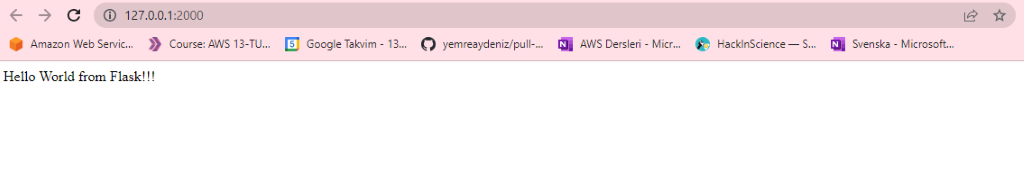
Burada belirtilen linkte web tarayıcısında bizi bu sayfaya götürür.

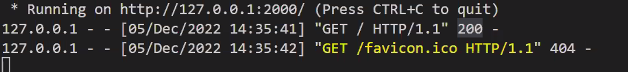
Port belirtmezsek 5000 portunda çalıştırır. Şimdi portu değiştirmek için terminale gidip ctrl+c ile çalışan kodu durduruyoruz ve portu 2000 e değiştiriyoruz:

En son satırdaki koda gidip True dan sonra virgüle ekleyip port=2000 diyerek değiştirdik. Boş olan port hangisi ise onu yazarız.



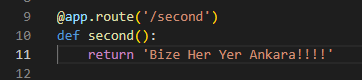
Tekrar save edip kodu çalıştırdık ve port=2000 de çalıştı:



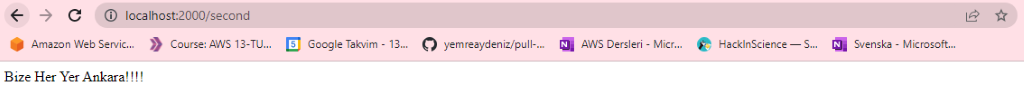


En altta iki satırda yazan 200 ve 404 ler HTML durum kodlarıdır. Bunarı seri olarak bilmemiz gerekir. (200 serisi nedir 400 serisi nedir şeklinde)

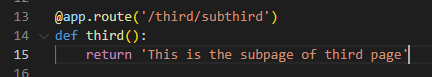
Şimdi yeni bir kod bloğu ekliyoruz:



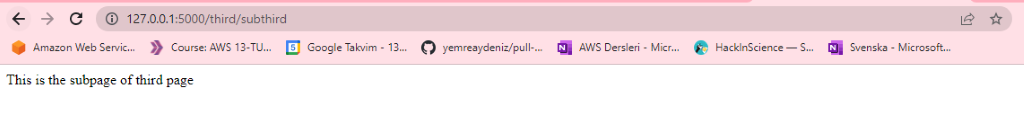
Burada / tan sonra second yazınca aşağıdaki yazı çıksın demektir bu. Web sayfaları bu şekilde dizayn edilir. Yani bu web sitesinin ikinci sayfası.



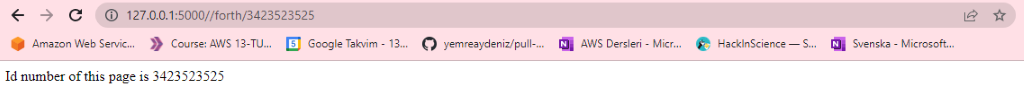
Şimdi yeni bir kod bloğu ekliyoruz: /third/subthird sayfası oldu bu.



Portu tekrar 5000 e aldım.



Yeni bir kod bloğu giriyoruz: Bu /forth/ tan sonra bir sayı girdiğimizde aşağıdaki yazıyı çıktı verecek:

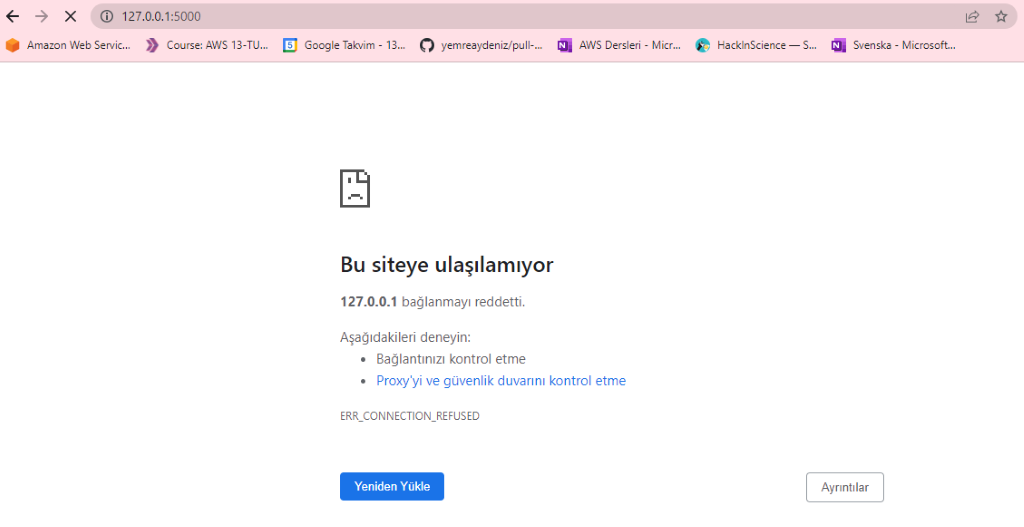




Burada debug mode olduğunu görüyoruz.

Debug mode hata ayıklama modudur. Syntax hatası olduğu zaman debug mode sayesinde hangi satırda hata olduğunu bize gösterir.

Ctrl+c ile çalışan kodu sonlandırıyoruz. Web tarayıcısında sayfayı yayınlamayacaktır:

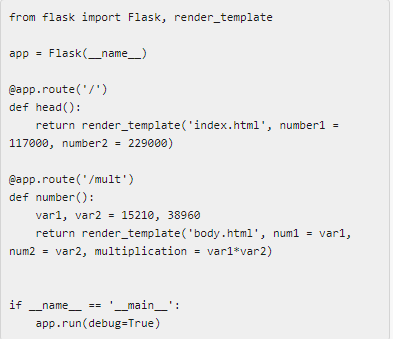


Flask-01 i tamamladık.

Şimdi Flask-02 ye geçiyroruz. Flask 02 klasörü içinde (templates dışında ama) jinja.py dosyası oluşturduk:



Sonra aşağıdaki kod bloğunu jinja.py nin içine yaptıştıralım:



Birinci satırda ihtiyaç duyduğu kütüphaneleri import ediyor. (render\_template modülünü) template kiralıyoruz.

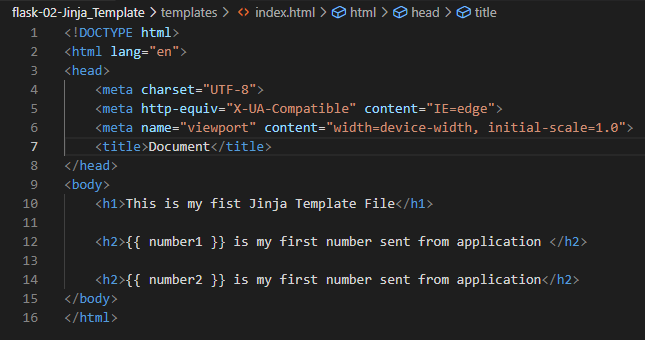
Sonra app diye bir obje oluşturuyor.

En sonunda debug=True modda ve port=5000 de run ediyor.

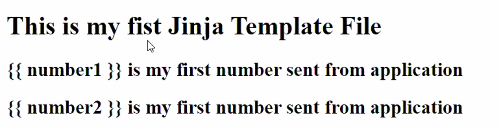
Ancak sayfayı modifiye edilmiş:

1. Satırıda ana sayfada index.html i kirala demiş.

Index.html de ne var:



Index.html çalıştığında şöyle bir template verecek:

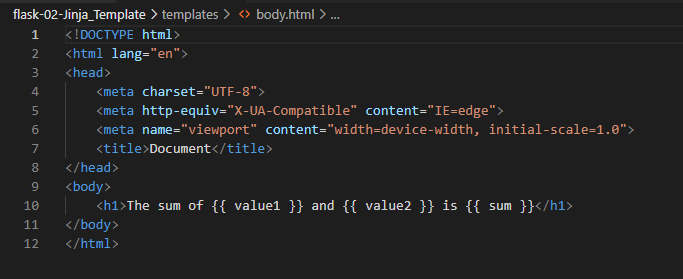


Orada number 1 değerine 117000 number 2 değerine de 229000 sayılarını ata demiş.

Şimdi codu run ettiğimizde port=5000 de bu numaraları görebiliyoruz:

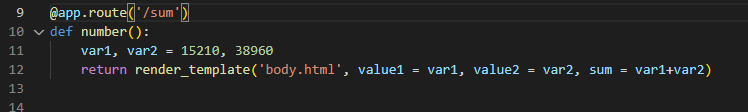


1. Satırdaki kod bloğuna bakalım: burada da body.html i kirala demiş. Body.html de ne var:

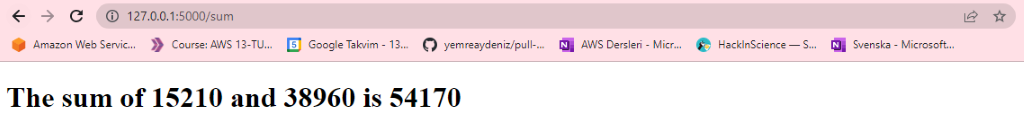


Toplama fonksiyonu var.süslü parantez içindeki üç tane değişken var. Bu üçüne değer veriyoruz:

Ve 9.satırdaki kod bloğunun son satırını şu şekilde değiştiyoruz:



Şimdi web tarayıcıya gidip /sum sayfasına bakıyoruz:



Verdiğimiz sayıları topladığını görüyoruz. Burada index.html ve body.html templateleri kiralanmış oluyor.

Özetle en alttaki port kısmında önce çalışan port bulacaz.

İkincisi kiraladığımız templatelerde hangi değişkenlere değer verecez. Bunları inceleyip bulup değer atayacaz.

Local repomuzda main branchtayken yaptığımız çalışmaları aşağıdaki komutlarla push ediyoruz:

git add .   
git commit -m "flask-01 and flask-02 added"   
git push

GitHub da repomuzu kontrol ettiğimizde remote repoda görüyoruz:

