

Uygulamalı Laboratuvar: Flask Kullanarak Bir Web Uygulaması Oluşturma ve Dağıtma



Giriş

Bu laboratuvara, matematiksel fonksiyonların temel bir uygulamasını oluşturuyor ve bunu Flask kullanarak bir web arayüzü üzerinden dağıtıyoruz. Amaç, şimdije kadar kursta edinilen tüm bilgileri birleştirmek ve uygulama geliştirme ve dağıtım adımlarını uygulamalı olarak görmek.

Gerekli tahmini süre: **30** dakika

Hedefler

Bu görevde şunları yapacaksınız:

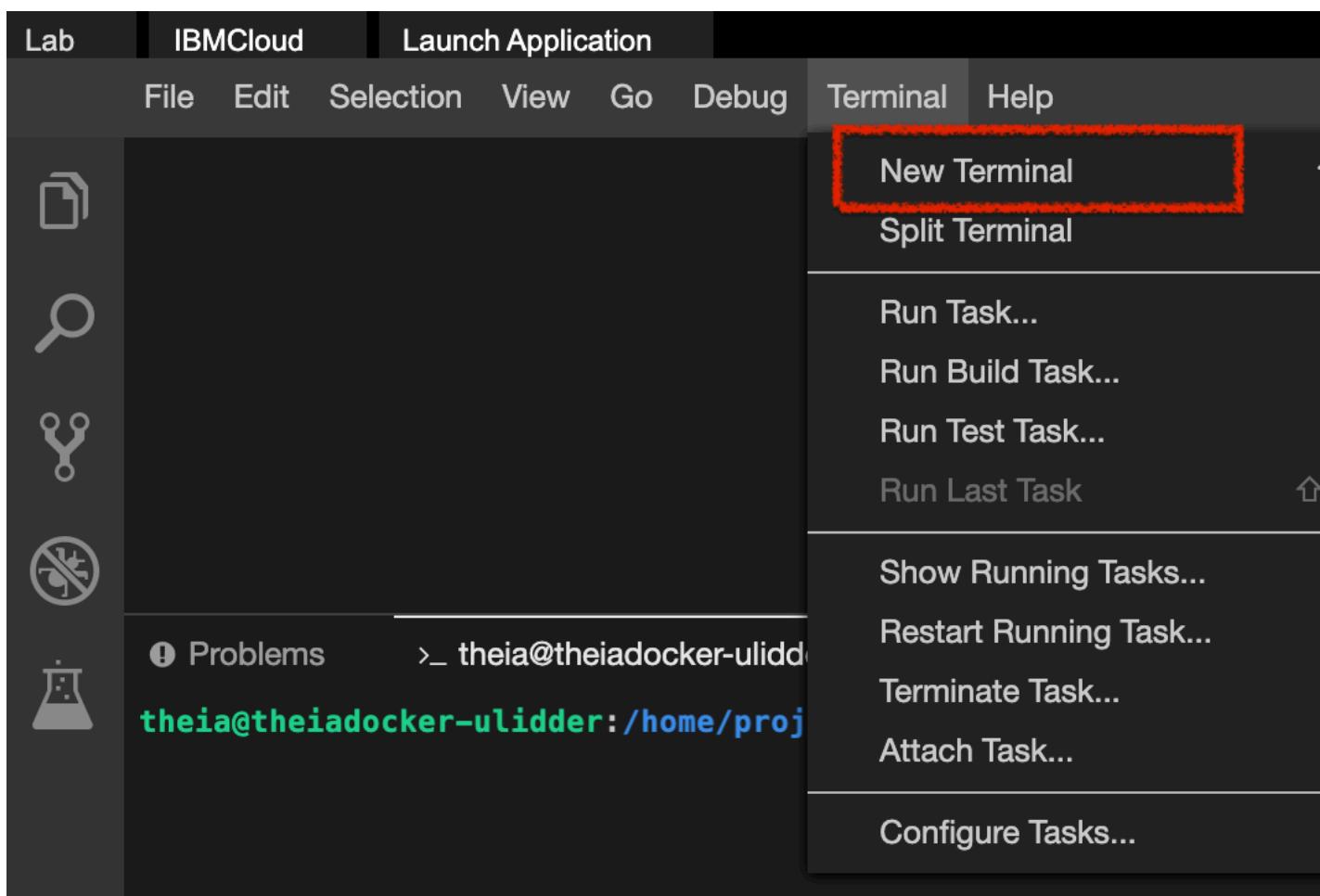
- Görev 1: Matematiksel fonksiyonları oluşturun.
- Görev 2: Fonksiyonları paketleyin ve paketi test edin.
- Görev 3: Flask kullanarak uygulama paketinin web dağıtımını gerçekleştirin.

Görev 1: Matematiksel Fonksiyonları Yaz

Bu görevde, iki değeri toplamak, çıkarmak ve çarpmak için fonksiyonlar içeren bir betik yazmanız gerekmektedir. Bu betiği `mathematics.py` adını verelim.

Bu görev için adımları takip edin.

1. Editördeki menüyü kullanarak bir terminal penceresi açın: Terminal > Yeni Terminal.



2. Proje ana dizinine gidin.

```
cd /home/project
```

3. Ön çalışma laboratuvarında kopyaladığınız klon URL'sinden proje dizinini Git ile klonlamak için aşağıdaki komutu çalıştırın.

```
git clone https://github.com/ibm-developer-skills-network/hjbsk-build_deploy_app_flask
```

4. `practice_project` klasörüne geçin.

```
cd /home/project/hjbsk-build_deploy_app_flask
```

5. `Maths` adında bir klasör oluşturun ve o dizine geçin.

```
mkdir Maths  
cd Maths
```

6. Gezginde `Maths` dizinine gidin ve `mathematics.py` adında yeni bir dosya oluşturun.

7. `mathematics.py` dosyasına `a` ve `b` sayı argümanlarını alan **summation** fonksiyonunu ekleyin.

▼ Çözüm için buraya tıklayın

```
'''python  
def summation(a, b):  
    result = a + b  
    return result  
'''
```

8. `mathematics.py` dosyasına `a` ve `b` sayı argümanlarını alan **subtraction** fonksiyonunu ekleyin.

▼ Çözüm için buraya tıklayın

```
'''python  
def subtraction(a, b):  
    result = a - b  
    return result  
'''
```

9. `mathematics.py` dosyasına `a` ve `b` sayı argümanlarını alan **multiplication** fonksiyonunu ekleyin.

▼ Çözüm için buraya tıklayın

```
```python
def multiplication(a, b):
 result = a * b
 return result
```

```

```
mathematics.py X server.py ↓

Maths > mathematics.py > multiplication

1 def summation(a, b):
2     result=a+b
3     return result
4
5 def subtraction(a, b):
6     result=a-b
7     return result
8
9 def multiplication(a, b):
10    result=a*b
11    return result
12
```

Görev 2: Fonksiyonları Paketle

1. Maths dizininde `__init__.py` dosyasını oluşturun.

2. `__init__.py` dosyasına `mathematics.py` dosyasını aktarın.

```
from . import mathematics
```

3. `server.py` dosyasında Maths paketini içe aktarın.

```
from Maths.mathematics import summation, subtraction, multiplication
```

4. `server.py` dosyasında, `/` uç noktası için `index.html` dosyasını render eden bir yöntem uygulayın.

```
@app.route("/")
def render_index_page():
    return render_template('index.html')
```

- server.py dosyasında /sum uç noktası için sağlanan alanda, önceki bölümde oluşturduğunuz paketten uygun toplama fonksiyonunu kullanan bir fonksiyon uygulayın. Fonksiyon, istek parametrelerinden num1 ve num2'yi float girdi olarak almalıdır. Daha sonra, sonucu `is_integer()` yöntemi ile tam sayı olup olmadığını kontrol etmelidir. Eğer öyleyse, sonucu bir string olarak döndürmeden önce tam sayıya dönüştürmelidir.
- server.py dosyasında /sub uç noktası için sağlanan alanda, önceki bölümde oluşturduğunuz paketten uygun çıkarma fonksiyonunu kullanan bir fonksiyon uygulayın. Fonksiyon, istek parametrelerinden num1 ve num2'yi float girdi olarak almalıdır. Daha sonra, sonucu `is_integer()` yöntemi ile tam sayı olup olmadığını kontrol etmelidir. Eğer öyleyse, sonucu bir string olarak döndürmeden önce tam sayıya dönüştürmelidir.
- server.py dosyasında /mul uç noktası için sağlanan alanda, önceki bölümde oluşturduğunuz paketten uygun çarpma fonksiyonunu kullanan bir fonksiyon uygulayın. Fonksiyon, istek parametrelerinden num1 ve num2'yi float girdi olarak almalıdır. Daha sonra, sonucu `is_integer()` yöntemi ile tam sayı olup olmadığını kontrol etmelidir. Eğer öyleyse, sonucu bir string olarak döndürmeden önce tam sayıya dönüştürmelidir.

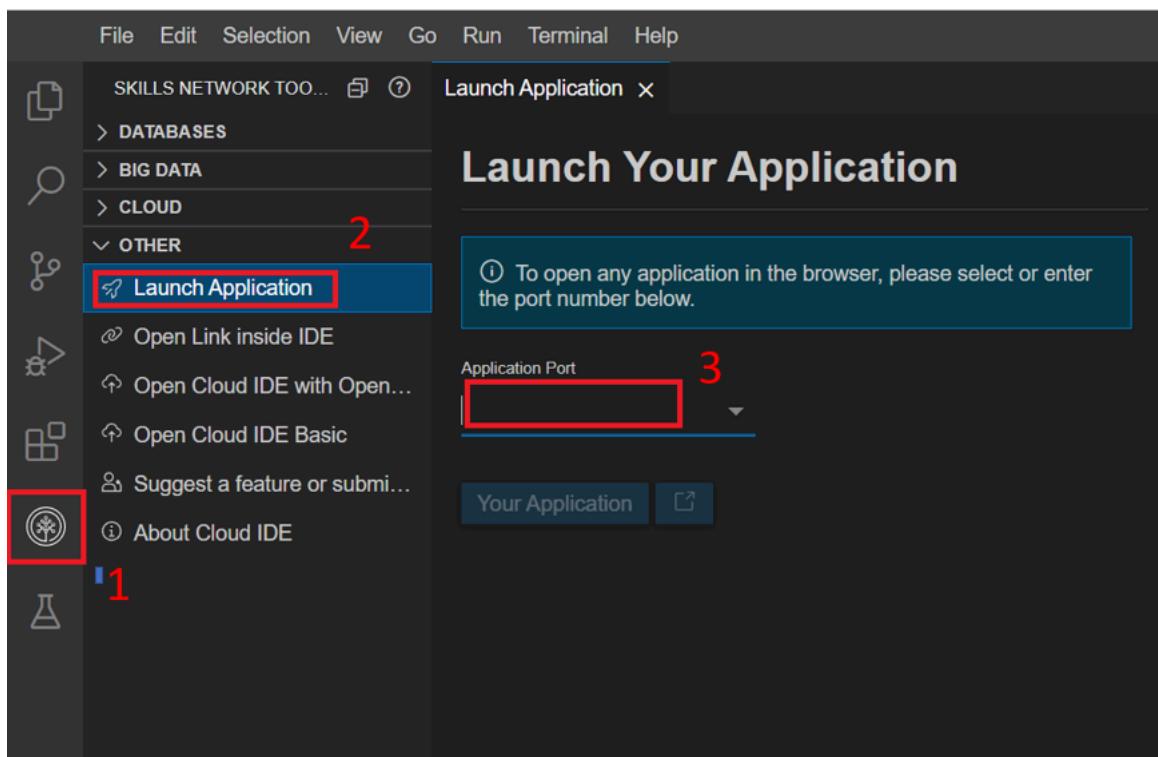
Görev 3: Uygulama paketinin Flask kullanarak Web Dağıtımı

- Terminalde mevcut dizini `hbbsk-build_deploy_app_flask` dizinine değiştirin ve sunucuya terminalden çalıştırın.

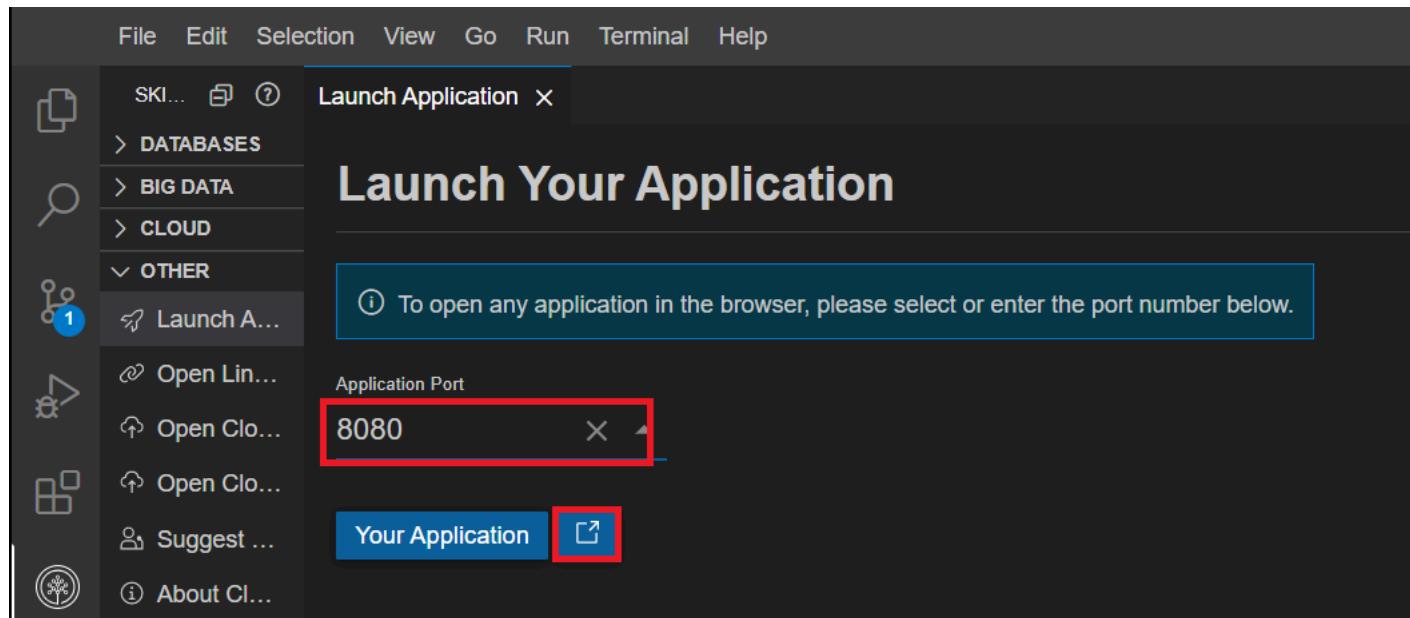
```
cd /home/project/hbbsk-build_deploy_app_flask && python3.11 server.py
```

- Sunucunun 8080 portunda başladığını göreceksiniz.

- Soldaki Skills Network butonuna tıklayın, bu Skills Network Toolbox'u açacaktır. Ardından Diğer, sonra Uygulamayı Başlata tıklayın. Buradan port numarasını girebilmeniziniz.



8080 portuna bağlanın ve Başlat butonuna tıklayın.



4. Aşağıda gösterildiği gibi bir indeks sayfasıyla yeni bir tarayıcı penceresi açılır.

Basic Mathematical /

Number 1

Number 2

Add

Subtract

Multiply

Response from :

Uygulamanızı istenen çıktılar için test edin. Aşağıda bazı örnekler gösterilmektedir.

Basic Mathematical /

Number 1

10

Number 2

20

Add

Subtract

Multiply

Response from :

Result: 30

Basic Mathematical /

Number 1

20

Number 2

10

Add

Subtract

Multiply

Response from :

Result: 10

Basic Mathematical /

Number 1

20

Number 2

10

Add

Subtract

Multiply

Response from :

Result: 200

(İsteğe Bağlı) Uygulama Alıştırması

İlgilenen öğrenciler, bu dağıtılmış uygulamaya hata yönetimi yeteneği eklemeyi deneyebilirler. Örneğin, arayüz matematiksel işlemler için sayısal olmayan girişler alındığında sistemin tepkisi ne olmalıdır?

Sonuç

Tebrikler! Bu projenin görevlerini tamamladınız.

Bu laboratuvarın sonunda şunları başardınız:

1. Matematiksel işlemler gerçekleştiren fonksiyonlar oluşturuldu.
2. Bu fonksiyonlar için bir paket oluşturuldu.
3. Flask kullanarak bu paketi kullanan uygulama localhost'ta dağıtıldı.