BASIC FLASK SET UP AND TESTING

Setting up Flask

Note: I invoked these via command line client in the directory where my python resides. See Screenshots

Install the Virtual Environment package:

python -m pip install virtualenv

```
C:\Users\damar\AppData\Local\Programs\Python\Python311>python -m pip install virtualenv
Collecting virtualenv
  Obtaining dependency information for virtualenv from https://files.pythonhosted.org/packages/07/4d/410156100224c5e2f00
11d435e477b57aed9576fc7fe137abcf14ec16e11/virtualenv-20.26.3-py3-none-any.whl.metadata
  Downloading virtualenv-20.26.3-py3-none-any.whl.metadata (4.5 kB)
Collecting distlib<1,>=0.3.7 (from virtualenv)
  Obtaining dependency information for distlib<1,>=0.3.7 from https://files.pythonhosted.org/packages/8e/41/9307e4f5f997
6bc8b7fea0b66367734e8faf3ec84bc0d412d8cfabbb66cd/distlib-0.3.8-py2.py3-none-any.whl.metadata
  Downloading distlib-0.3.8-py2.py3-none-any.whl.metadata (5.1 kB)
Collecting filelock<4,>=3.12.2 (from virtualenv)
  Obtaining dependency information for filelock<4,>=3.12.2 from https://files.pythonhosted.org/packages/ae/f0/48285f0262
fe47103a4a45972ed2f9b93e4c80b8fd609fa98da78b2a5706/filelock-3.15.4-py3-none-any.whl.metadata
  Downloading filelock-3.15.4-py3-none-any.whl.metadata (2.9 kB)
Collecting platformdirs<5,>=3.9.1 (from virtualenv)
  Obtaining dependency information for platformdirs<5,>=3.9.1 from https://files.pythonhosted.org/packages/68/13/2aa1f0e
1364feb2c9ef45302f387ac0bd81484e9c9a4c5688a322fbdfd08/platformdirs-4.2.2-py3-none-any.whl.metadata
  Downloading platformdirs-4.2.2-py3-none-any.whl.metadata (11 kB)
Downloading virtualenv-20.26.3-py3-none-any.whl (5.7 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 5.7/5.7 MB 4.0 MB/s eta 0:00:00
Downloading distlib-0.3.8-py2.py3-none-any.whl (468 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 468.9/468.9 kB 5.9 MB/s eta 0:00:00
Downloading filelock-3.15.4-py3-none-any.whl (16 kB)
Downloading platformdirs-4.2.2-py3-none-any.whl (18 kB)
Installing collected packages: distlib, platformdirs, filelock, virtualenv
Successfully installed distlib-0.3.8 filelock-3.15.4 platformdirs-4.2.2 virtualenv-20.26.3
```

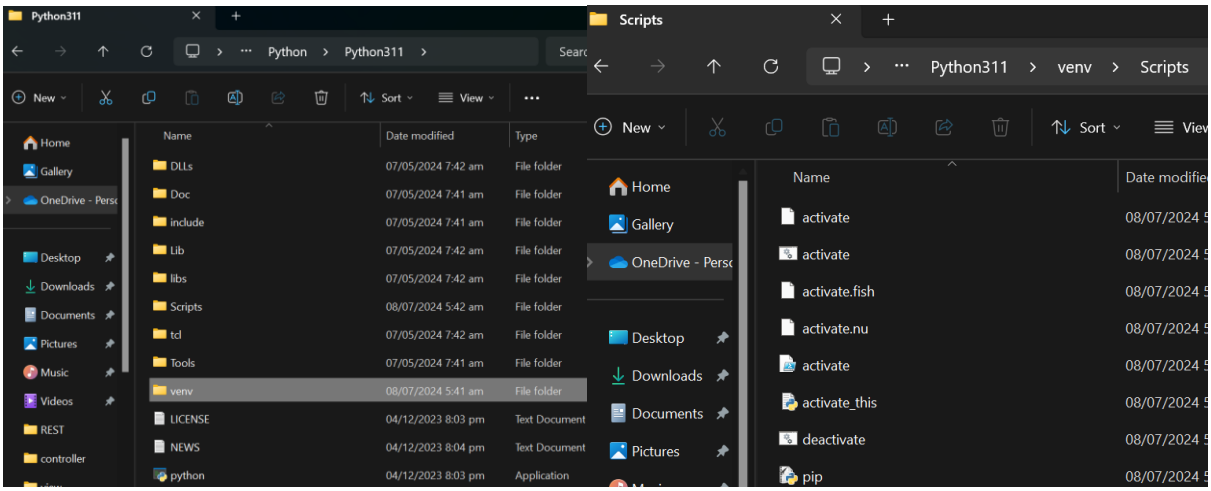
Create a Virtual Environment

virtualenv venv

```
C:\Users\damar\AppData\Local\Programs\Python\Python311>virtualenv venv
created virtual environment CPython3.11.7.final.0-64 in 6495ms
  creator CPython3Windows(dest=C:\Users\damar\AppData\Local\Programs\Python\Python311\venv, clear=False, no_vcs_ignore=F
alse, global=False)
  seeder FromAppData(download=False, pip=bundle, setuptools=bundle, wheel=bundle, via=copy, app_data_dir=C:\Users\damar\
AppData\Local\pypa\virtualenv)
    added seed packages: pip==24.1, setuptools==70.1.0, wheel==0.43.0
  activators BashActivator,BatchActivator,FishActivator,NushellActivator,PowerShellActivator,PythonActivator
```

BASIC FLASK SET UP AND TESTING

After invoking virtualenv venv, a ***folder named*** *"venv"* ***for the virtual environment will be generated in your python folder. The folder has the Scripts Folder to call and activate the virtual environment***



Activate your Virtual Environment

venv\Scripts\activate

```
C:\Users\damar\AppData\Local\Programs\Python\Python311>venv\Scripts\activate

C:\Users\damar\AppData\Local\Programs\Python\Python311>()

(venv) C:\Users\damar\AppData\Local\Programs\Python\Python311>
```

Then, pip install flask

python -m pip install flask

```
(venv) C:\Users\damar\AppData\Local\Programs\Python\Python311>python -m pip install flask
Collecting flask
  Downloading flask-3.0.3-py3-none-any.whl.metadata (3.2 kB)
Collecting Werkzeug>=3.0.0 (from flask)
  Downloading werkzeug-3.0.3-py3-none-any.whl.metadata (3.7 kB)
Collecting Jinja2>=3.1.2 (from flask)
  Downloading jinja2-3.1.4-py3-none-any.whl.metadata (2.6 kB)
Collecting itsdangerous>=2.1.2 (from flask)
  Downloading itsdangerous-2.2.0-py3-none-any.whl.metadata (1.9 kB)
Collecting click>=8.1.3 (from flask)
  Downloading click-8.1.7-py3-none-any.whl.metadata (3.0 kB)
Collecting blinker>=1.6.2 (from flask)
  Downloading blinker-1.8.2-py3-none-any.whl.metadata (1.6 kB)
Collecting colorama (from click>=8.1.3->flask)
  Downloading colorama-0.4.6-py2.py3-none-any.whl.metadata (17 kB)
Collecting MarkupSafe>=2.0 (from Jinja2>=3.1.2->flask)
  Downloading MarkupSafe-2.1.5-cp311-cp311-win_amd64.whl.metadata (3.1 kB)
Downloading flask-3.0.3-py3-none-any.whl (101 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 101.7/101.7 kB 3.0 MB/s eta 0:00:00
Downloading blinker-1.8.2-py3-none-any.whl (9.5 kB)
Downloading click-8.1.7-py3-none-any.whl (97 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 97.9/97.9 kB 5.5 MB/s eta 0:00:00
Downloading itsdangerous-2.2.0-py3-none-any.whl (16 kB)
Downloading jinja2-3.1.4-py3-none-any.whl (133 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 133.3/133.3 kB 4.0 MB/s eta 0:00:00
Downloading werkzeug-3.0.3-py3-none-any.whl (227 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 227.3/227.3 kB 2.7 MB/s eta 0:00:00
Downloading MarkupSafe-2.1.5-cp311-cp311-win_amd64.whl (17 kB)
Downloading colorama-0.4.6-py2.py3-none-any.whl (25 kB)
Installing collected packages: MarkupSafe, itsdangerous, colorama, blinker, Werkzeug, Jinja2, click, flask
Successfully installed Jinja2-3.1.4 MarkupSafe-2.1.5 Werkzeug-3.0.3 blinker-1.8.2 click-8.1.7 colorama-0.4.6 flask-3.0.3 itsdange
rous-2.2.0
```

BASIC FLASK SET UP AND TESTING

Create the Python Code for your Flask Instance

The Flask Code (My filename is myFlask.py)

```
from flask import Flask, redirect, url_for, request
app = Flask(__name__)


@app.route('/nakuha/<val>')
def nakuha(val):
   wew = ('''
        <html>
        <body>
        <center>
        <h1>My Sample Flask</h1>
        '''+'I got the value: %s' % val+
        '''
        </body>
        </center>
        </html>
        '''
        )

   return wew


@app.route('/test', methods=['POST', 'GET'])
def getData():
   if request.method == 'POST':
      valueGot = request.form['theKey']
      return redirect(url_for('nakuha', val=valueGot))
   else:
      valueGot = request.args.get('theKey')
      return redirect(url_for('nakuha', val=valueGot))


if __name__ == '__main__':
   app.run(host='192.168.1.17',port='80',debug=True)
```

```
from flask import Flask, redirect, url_for, request
app = Flask(__name__)


@app.route('/nakuha/<val>')
def nakuha(val):
    wew = ('''
            <html>
            <body>
            <center>
            <h1>My Sample Flask</h1>
            '''+'I got the value: %s' % val+
            '''
            </body>
            </center>
            </html>
            '''
            )

    return wew


@app.route('/test', methods=['POST', 'GET'])
def getData():
    if request.method == 'POST':
        valueGot = request.form['theKey']
        return redirect(url_for('nakuha', val=valueGot))
    else:
        valueGot = request.args.get('theKey')
        return redirect(url_for('nakuha', val=valueGot))


if __name__ == '__main__':
    app.run(host='192.168.1.17',port='80',debug=True)
```

Flask instance (app) is running at my PC having IP of 192.168.1.17 using 80 as port. Function “getData “ anchors *@app.route* to capture the POSTED request of my web application. Once captured, it redirects to another url associated with function “nakuha” and pass the value therein. Value is then returned and viewed as an HTML page showing the value

Run your Flask Server

```
C:\Users\damar\AppData\Local\Programs\Python\Python311>python C:\Users\damar\Desktop\FlaskBasic\myFlask.py
 * Serving Flask app 'myFlask'
 * 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 http://192.168.1.17:80
Press CTRL+C to quit
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 123-990-347
```

The Web App

The web app is a simple HTML File that uses form. Form action was set to POST in the specified IP of the Flask app. Submit sends the value of the array_key of the input text “theKey” to the Flask app

```
<html>
<center>
<h1>My Python Scripting via Flask</h1>
      <form action = "http://192.168.1.17:80/test" method = "post">
      <p>Enter VALUE:</p>
      <p><input type = "text" name = "theKey" /></p>
      <p><input type = "submit" value = "submit" /></p>
      </form>
</center>
</html>
```

Sample Run

My Python Scripting via Flask

Enter VALUE:

wakokok

submit

My Sample Flask

I got the value: wakokok

BASIC FLASK SET UP AND TESTING

Code Interaction

```python
from flask import Flask, redirect, url_for, request
app = Flask(__name__)


@app.route('/nakuha/<val>')
def nakuha(val):
    wew = ('''
            <html>
            <body>
            <center>
            <h1>My Sample Flask</h1>
            '''+'I got the value: %s' % val+
            '''
            </body>
            </center>
            </html>
            '''
            )

    return wew


@app.route('/test', methods=['POST', 'GET'])
def getData():
    if request.method == 'POST':
        valueGot = request.form['theKey']
        return redirect(url_for('nakuha', val=valueGot))
    else:
        valueGot = request.args.get('theKey')
        return redirect(url_for('nakuha', val=valueGot))


if __name__ == '__main__':
    app.run(host='192.168.1.17',port='80',debug=True)
```

```html
<html>
<center>
<h1>My Python Scripting via Flask</h1>
      <form action = "http://192.168.1.17:80/test" method = "post">
      <p>Enter VALUE:</p>
      <p><input type = "text" name = "theKey" /></p>
      <p><input type = "submit" value = "submit" /></p>
      </form>
</center>
</html>
```

In my test example, both Flask File and Web app file are in a desktop folder