

FLASK PYTHON WEB APPLICATION IN WINDOWS

WEB APPLICATION USING PYTHON

- Web application can be developed using python language
- To create a web application in python, we need **a web framework**
- Web framework is a built-in module / library which makes web development much easier

Popular Python Web Frameworks

- Django
- Flask
- Bottle
- Pyramid, etc,...

FLASK

- It is a lightweight web framework (micro web framework) which is used for creating dynamic web application in python language.
- Flask is called as **micro framework** because it **doesn't directly support the features like form validation, database abstraction, authentication** and so on
- Such features are obtained using flask extensions
- It provides small set of features like **URL routing and page rendering** but extended with many plugins like CouchDB, MangoDB, SQLAlchemy, Babel, etc,...

ROUTE (Connection between URL and View Function)

- In flask, route is used to store the web pages and it is called by the object of Flask class
- Every flask web application has several routes (URL)
- It is mainly used to **convert / map a web page (web link) to a python function** which is called as **view function** (It creates the connection route (URL) and view function)
- **Whatever function (python view function) returns will be shown to the user in the web browser.**

Example

- For example, <http://127.0.0.1:5000/> is the **main route** which will be used to display an index page or home page
- <http://127.0.0.1:5000/Contacts> may be **another route used for displaying contact page** that will provide the contact information about the website.
- It is an important to note that, **separate route should be used for each web page.**

View Function

- Every route should have a corresponding python function which will be converted to view function.
- Unlike normal function, It should return the value.
- The **return type of this function should be a string, dict, list, tuple with headers or status**, Response instance, or WSGI callable.

STEPS FOR CREATING HELLO WORLD FLASK WEB APPLICATION

Step 1: Create a new directory and navigate your CMD path or VSC path to that newly created directory

Step 2: Create a virtual environment directory for flask application

Syntax

```
python -m venv <user-defined folder-name>
```

Example

```
python -m venv env
```

Where,

env is a user defined folder (newly created directory by user)

Step 3: Activate the newly created virtual environment using the command below

Syntax

```
<environment folder name>\Scripts\activate
```

Example

```
env\Scripts\activate
```

Where,

env is a user defined folder.

- If the command above failed, **activate the Execution Policy** in windows using the command below
 - Open the CMD in windows with **administrator Mode** and run the command for activation of Execution Policy

```
powershell Set-ExecutionPolicy RemoteSigned
```

Step 4: Install Flask library using the pip command

```
pip install flask
```

Step 5: Create a new python file in the current directory and add the file to flask app using the command below

Syntax

```
set FLASK_APP=filename.py
```

Example

```
set FLASK_APP=helloweb.py
```

Step 6: Write a code for the newly created python file above

Step 7: Set the environment Settings before run the application

Syntax

```
$env:FLASK_APP="filename.py"
```

Example

```
$env:FLASK_APP = "helloweb.py"
```

Where,

- **env** is a reserved keyword.

NOTE

- Here python file must be submitted within double quotes otherwise interpreter will display the error message.

Step 8: Run the application by using the command

```
flask run
```

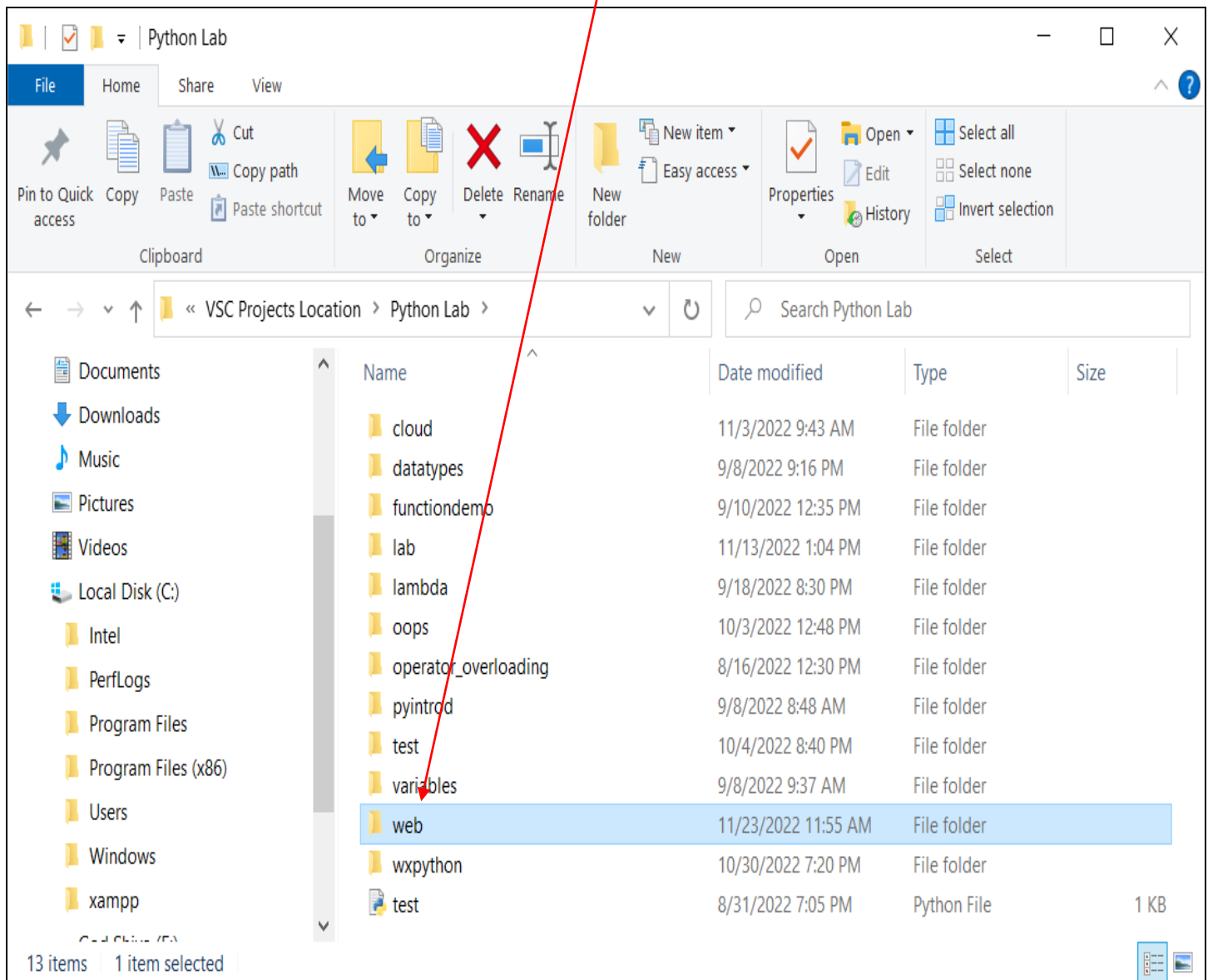
- After executing the command above, you will get a link for running the web application
- Copy the link and paste it into your web browser
- Done and you will get a result.

I. CREATING FLASK WEB APPLICATION IN WINDOWS

Application Type	:	Web Application
Web Framework	:	Flask
Language	:	Python (Python 3)
Python SDK	:	3.10.5
Tools Used	:	VSC Editor
Tested OS	:	Windows 10

Step 1:

- Create a new directory named **web** (either visually or in VSC editor) and navigate your CMD path or VSC path to that newly created directory.



Navigate the path to the newly created directory **web** using **cd** command

```
PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL JUPYTER

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

PS F:\VSC Projects Location\Python Lab> cd web
```

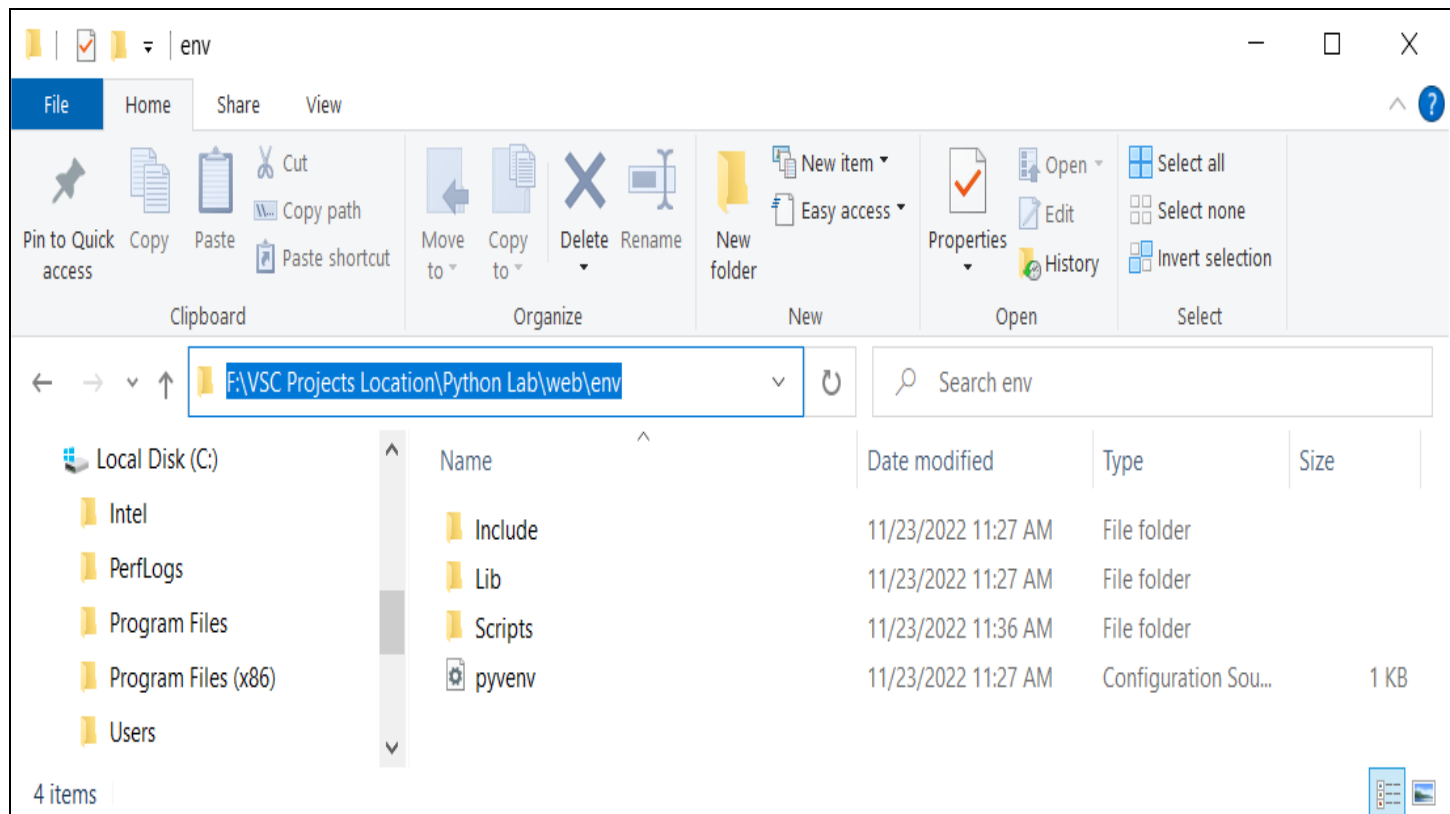
Step 2: Create a virtual environment directory named **env** for flask application using command **py -m venv <folder-name>**

```
PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL JUPYTER

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

PS F:\VSC Projects Location\Python Lab> cd web
PS F:\VSC Projects Location\Python Lab\web> py -m venv env
```

Contents of Virtual Environment Directory (env)



Step 3: Activate the newly created virtual environment using the command below

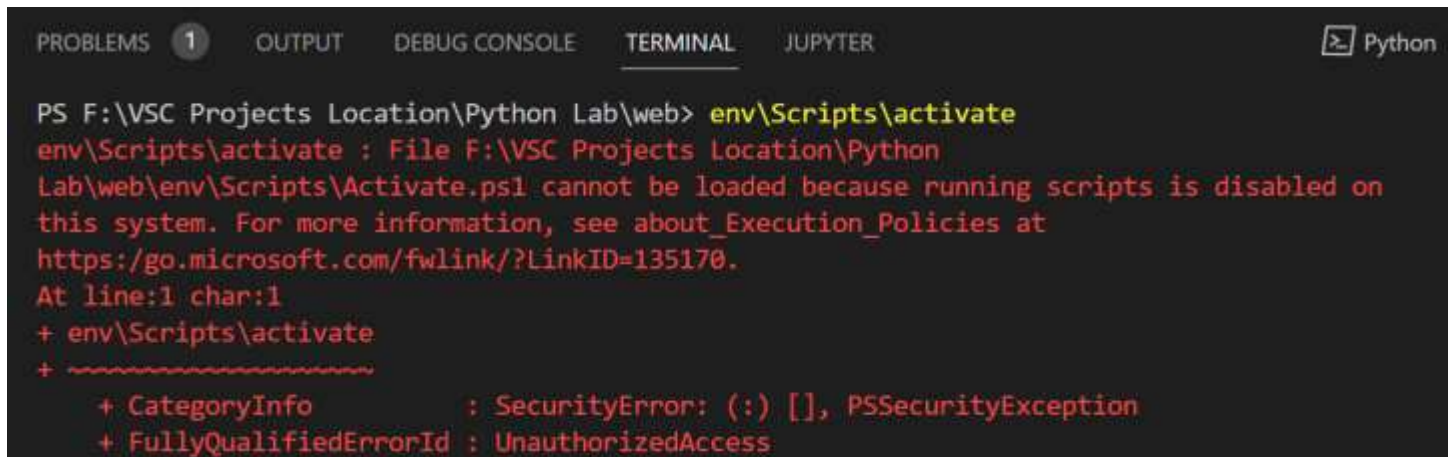
Syntax

```
<environment folder name>\Scripts\activate
```

Example

```
env\Scripts\activate
```

Error Screenshot



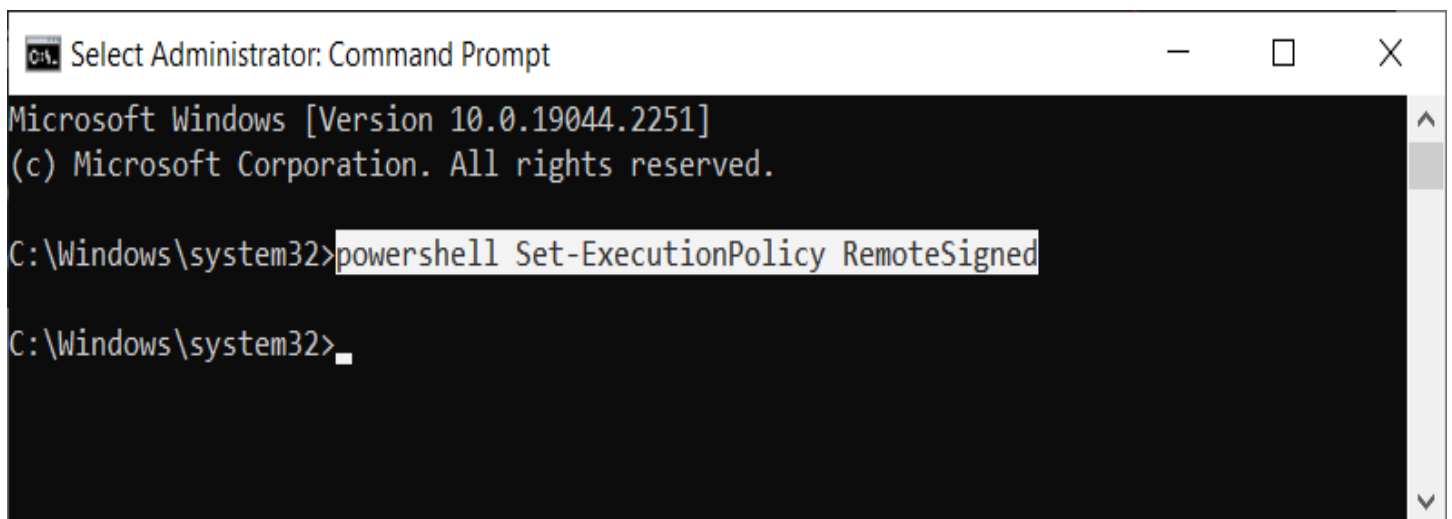
The screenshot shows a terminal window with the following text:

```
PS F:\VSC Projects Location\Python Lab\web> env\Scripts\activate
env\Scripts\activate : File F:\VSC Projects Location\Python
Lab\web\env\Scripts\Activate.ps1 cannot be loaded because running scripts is disabled on
this system. For more information, see about_Execution_Policies at
https://go.microsoft.com/fwlink/?LinkID=135170.
At line:1 char:1
+ env\Scripts\activate
+ ~~~~~
+ CategoryInfo          : SecurityError: (:) [], PSSecurityException
+ FullyQualifiedErrorId : UnauthorizedAccess
```

- If the command above failed, **activate the Execution Policy** in windows using the command below
 - Open the CMD in windows with **administrator Mode** and run the command for activation of Execution Policy

```
powershell Set-ExecutionPolicy RemoteSigned
```

Screenshot



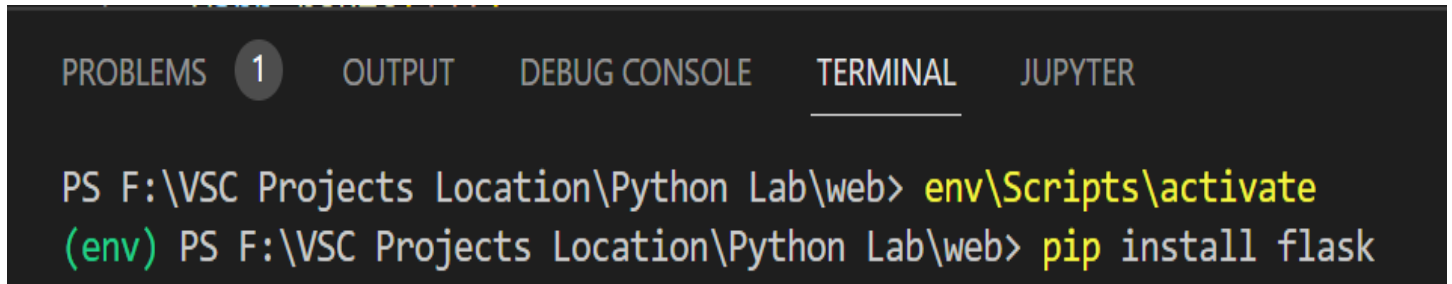
The screenshot shows a Windows Command Prompt window titled "Select Administrator: Command Prompt". The text inside the window is:

```
Microsoft Windows [Version 10.0.19044.2251]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\system32> powershell Set-ExecutionPolicy RemoteSigned

C:\Windows\system32> _
```

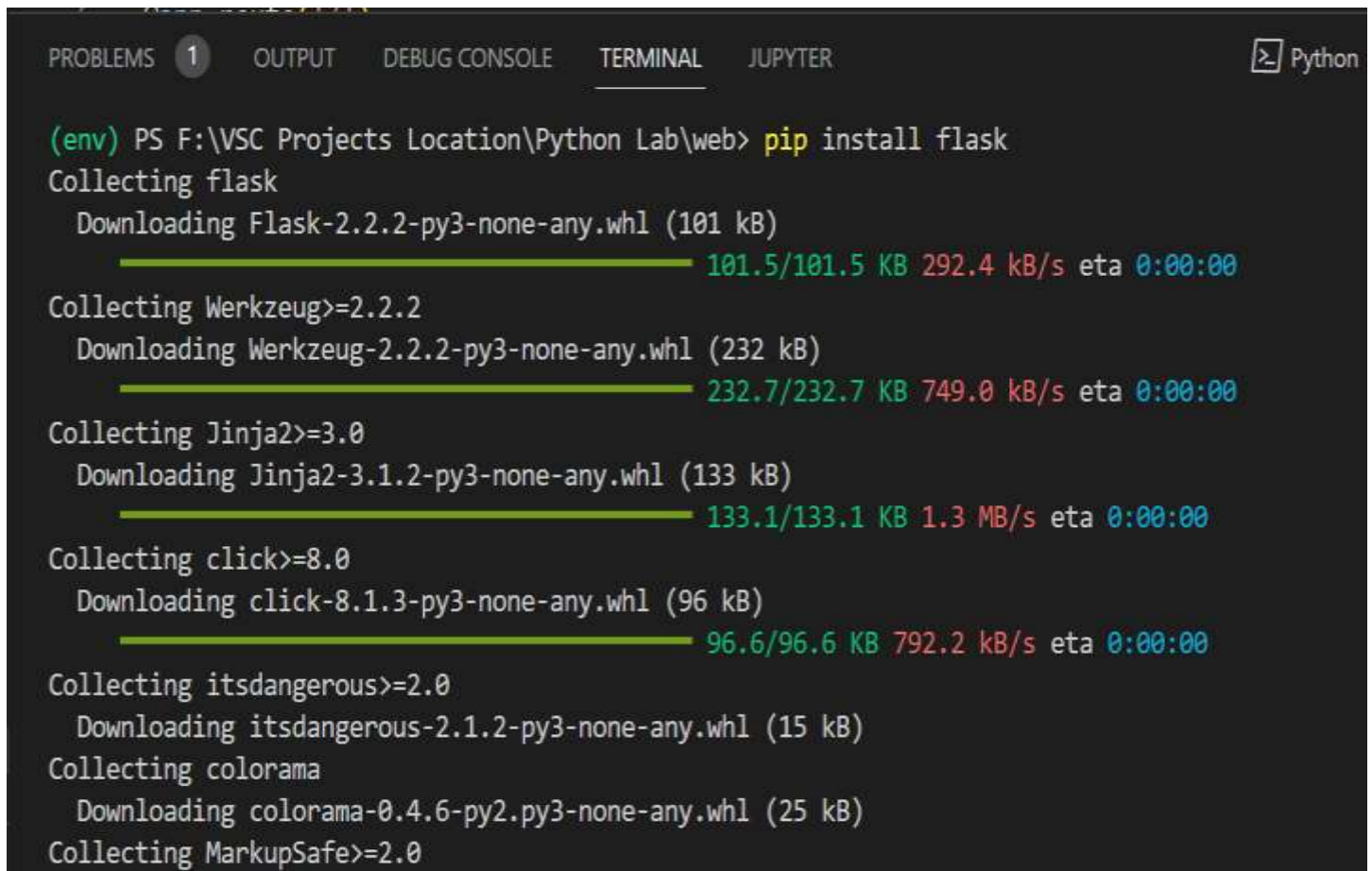

Success Status Screenshot



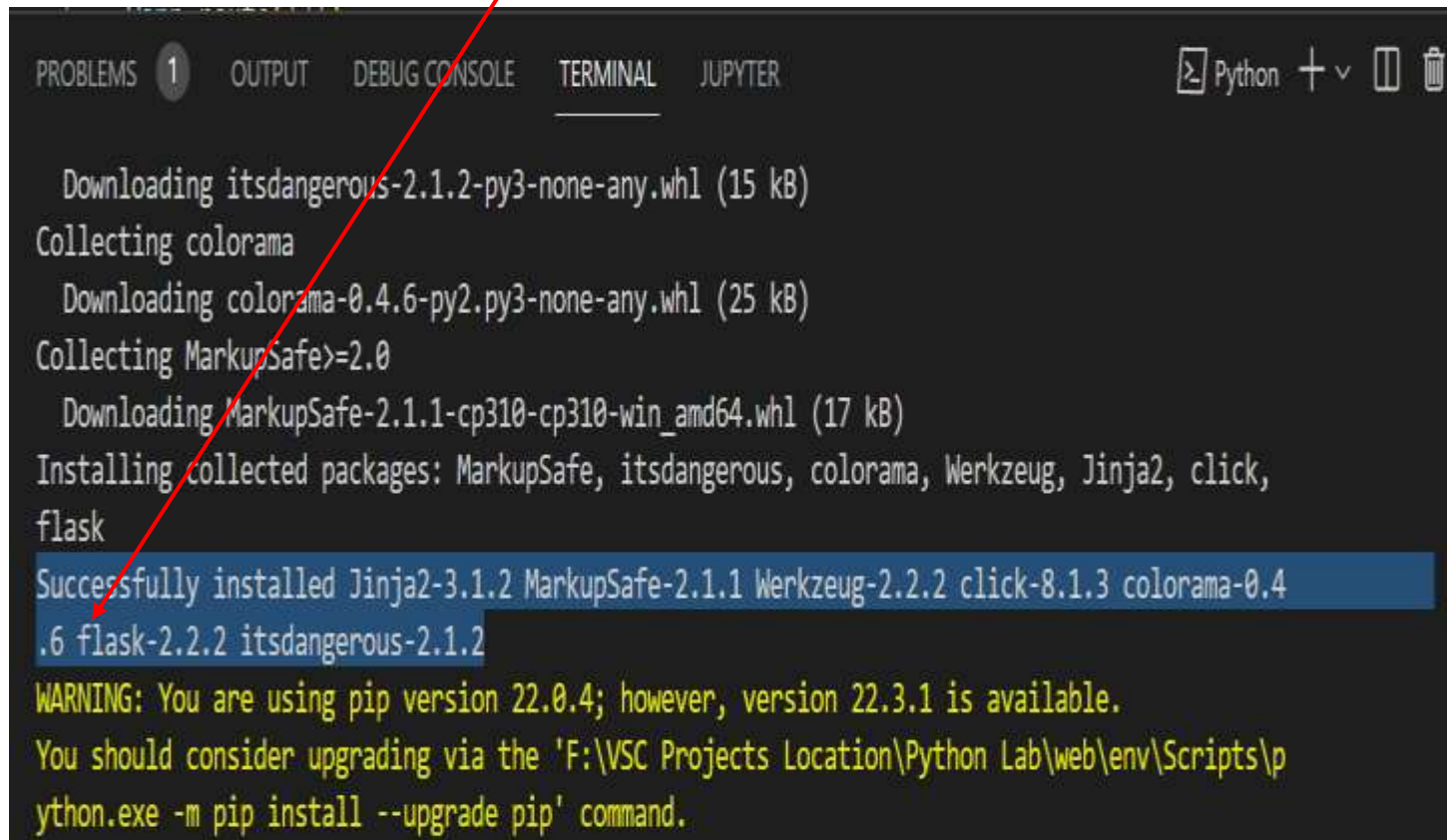
Step 4: Install Flask library using the pip command

pip install flask

Screenshot 1



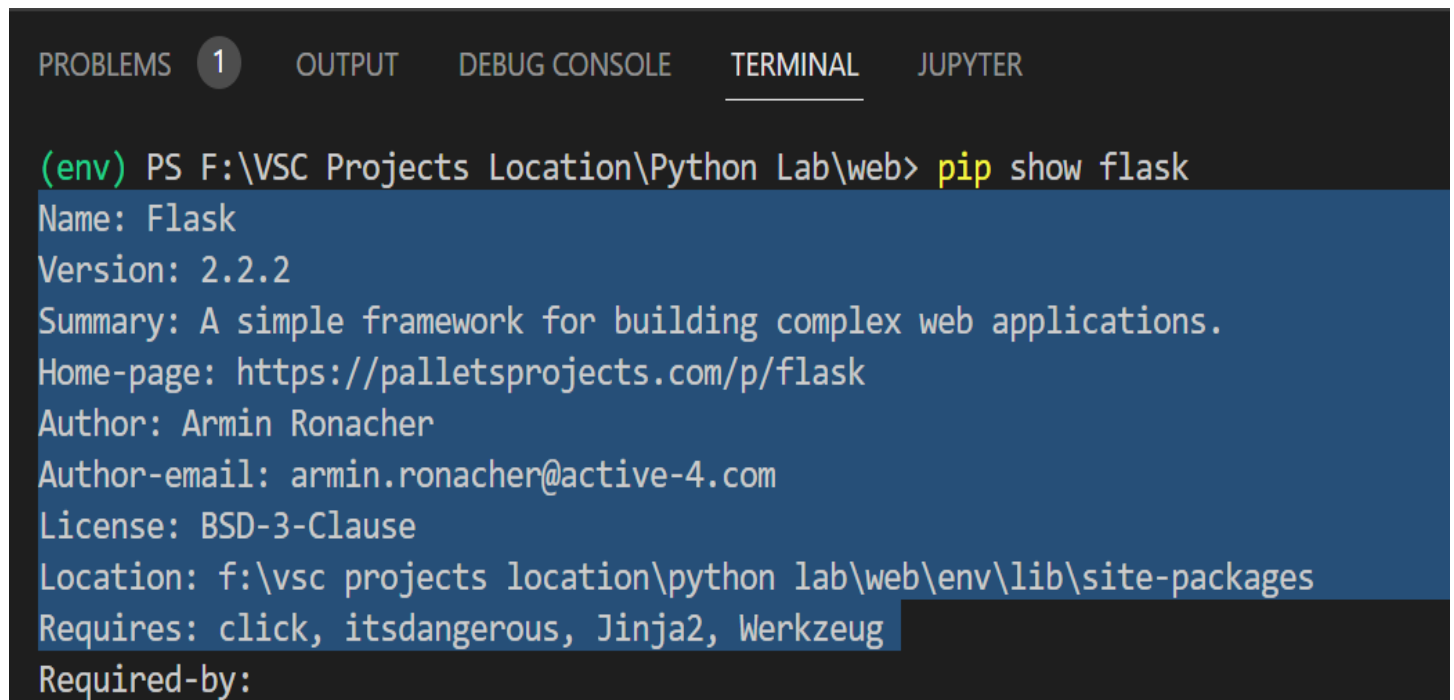
Screenshot 2 (pip install flask)



```
PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL JUPYTER Python + v [trash icon]

Downloading itsdangerous-2.1.2-py3-none-any.whl (15 kB)
Collecting colorama
  Downloading colorama-0.4.6-py2.py3-none-any.whl (25 kB)
Collecting MarkupSafe>=2.0
  Downloading MarkupSafe-2.1.1-cp310-cp310-win_amd64.whl (17 kB)
Installing collected packages: MarkupSafe, itsdangerous, colorama, Werkzeug, Jinja2, click, flask
Successfully installed Jinja2-3.1.2 MarkupSafe-2.1.1 Werkzeug-2.2.2 click-8.1.3 colorama-0.4.6 flask-2.2.2 itsdangerous-2.1.2
WARNING: You are using pip version 22.0.4; however, version 22.3.1 is available.
You should consider upgrading via the 'F:\VSC Projects Location\Python Lab\web\env\Scripts\python.exe -m pip install --upgrade pip' command.
```

Flask Details



```
PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL JUPYTER

(env) PS F:\VSC Projects Location\Python Lab\web> pip show flask
Name: Flask
Version: 2.2.2
Summary: A simple framework for building complex web applications.
Home-page: https://palletsprojects.com/p/flask
Author: Armin Ronacher
Author-email: armin.ronacher@active-4.com
License: BSD-3-Clause
Location: f:\vsc projects location\python lab\web\env\lib\site-packages
Requires: click, itsdangerous, Jinja2, Werkzeug
Required-by:
```

Step 5: Create a new python file in the current directory and add the file to flask app using the command below

Syntax

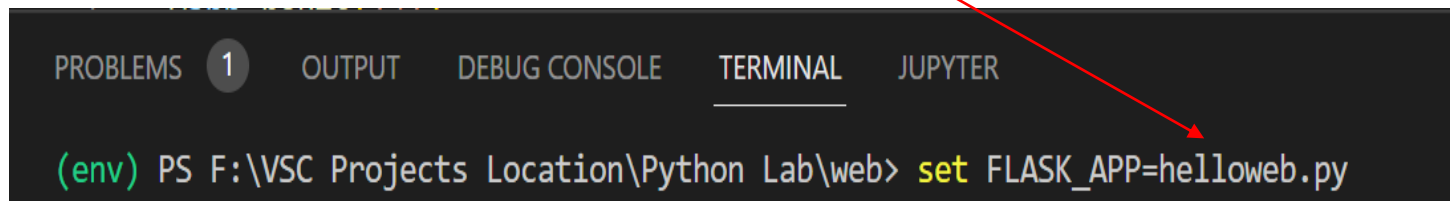
```
set FLASK_APP=filename.py
```

Example

```
set FLASK_APP=helloweb.py
```

Where,

helloweb.py is a newly created python file in the current directory.



```
PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL JUPYTER
(env) PS F:\VSC Projects Location\Python Lab\web> set FLASK_APP=helloweb.py
```

Step 6: Write a code for the newly created python file ab

1. SOURCE CODE

(helloweb.py)

load flask module

```
from flask import Flask
```

create an object for Flask class

```
app=Flask(__name__)
```

create a route (URL) for home page using Flask object

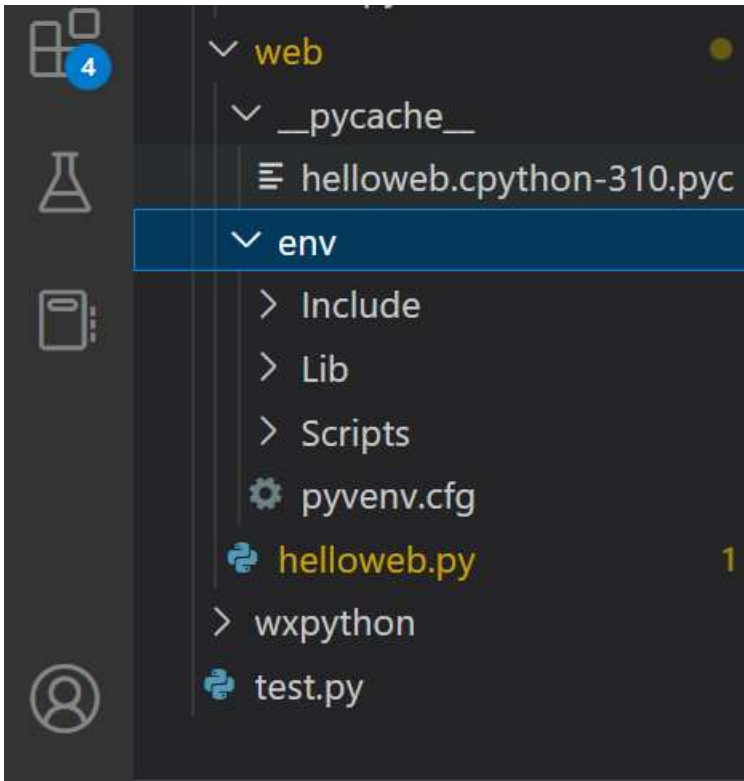
```
@app.route('/')
def disp():
```

```
    return "<h1>Welcome to Flask Python</h1>"
```

NOTE

- The **return value** of python view function named **disp()** must be a **string, dict, list, tuple with headers or status**, Response instance, or WSGI callable.

Project Structure in VSC Editor



Step 7: Set the environment Settings before run the application

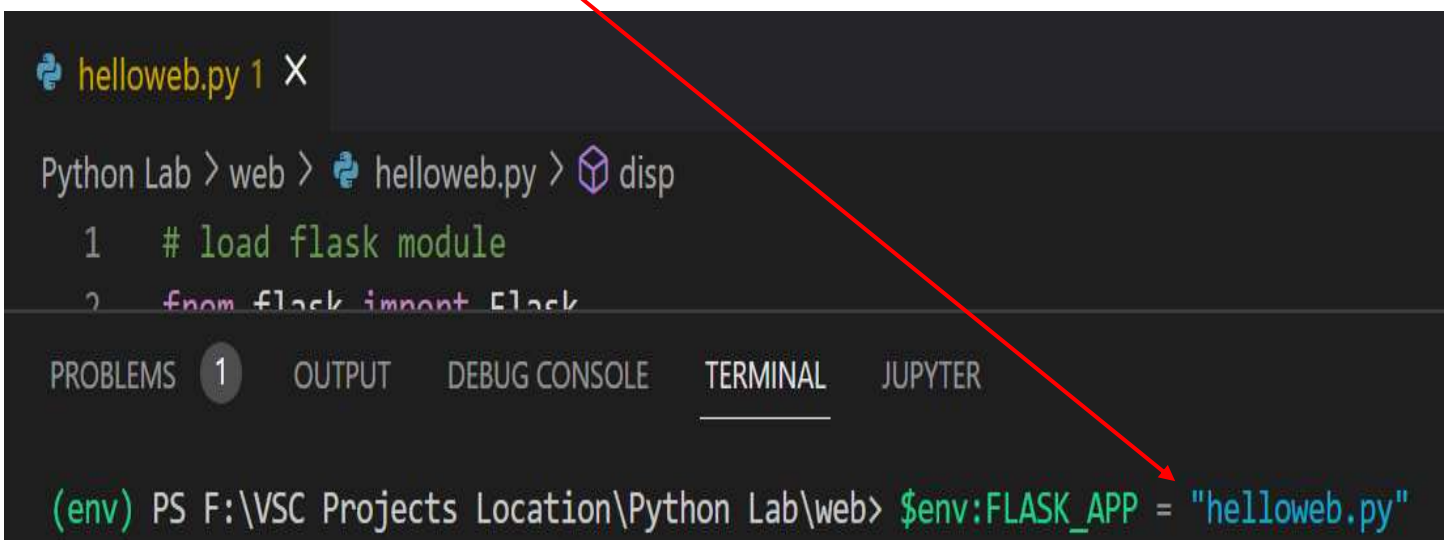
Syntax

```
$env:FLASK_APP=filename.py
```

Example

```
$env:FLASK_APP = "helloworld.py"
```

Screenshot

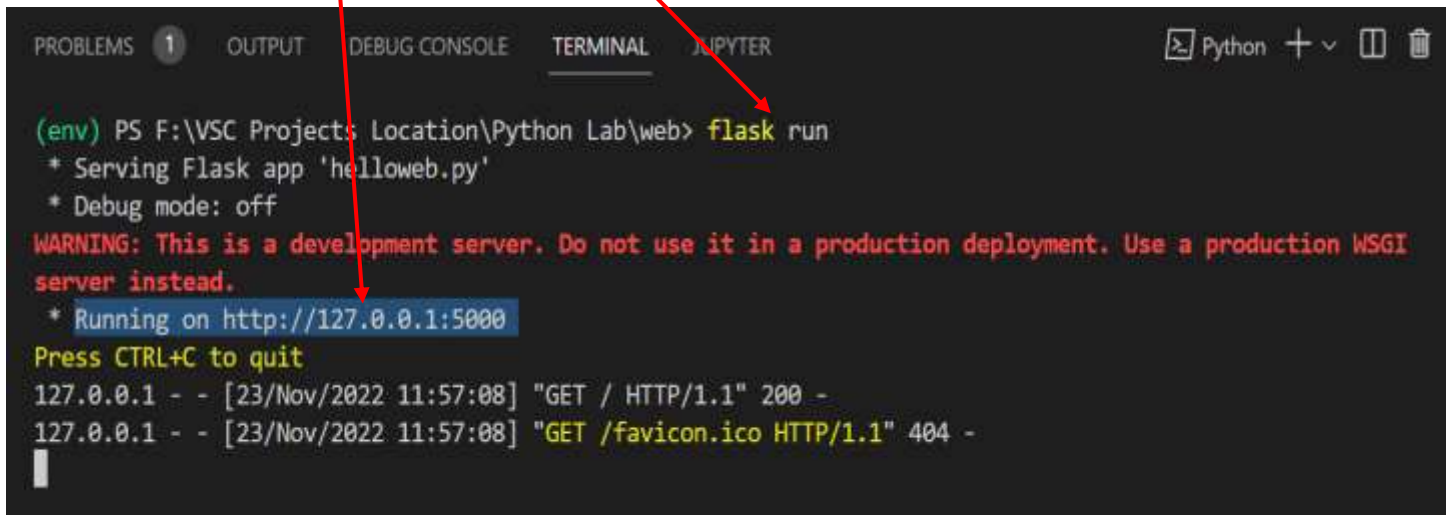


Step 8: Run the application by using the command

```
flask run
```

- After executing the command above, you will get a link for running the web application
- Copy the link and paste it into your web browser
- Done and you will get a result.

Screenshot



```
(env) PS F:\VSC Projects Location\Python Lab\web> flask run
* Serving Flask app 'helloweb.py'
* Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI
server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
127.0.0.1 - - [23/Nov/2022 11:57:08] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [23/Nov/2022 11:57:08] "GET /favicon.ico HTTP/1.1" 404 -
```

2. OUTPUT

