

PYTHON & FLASK

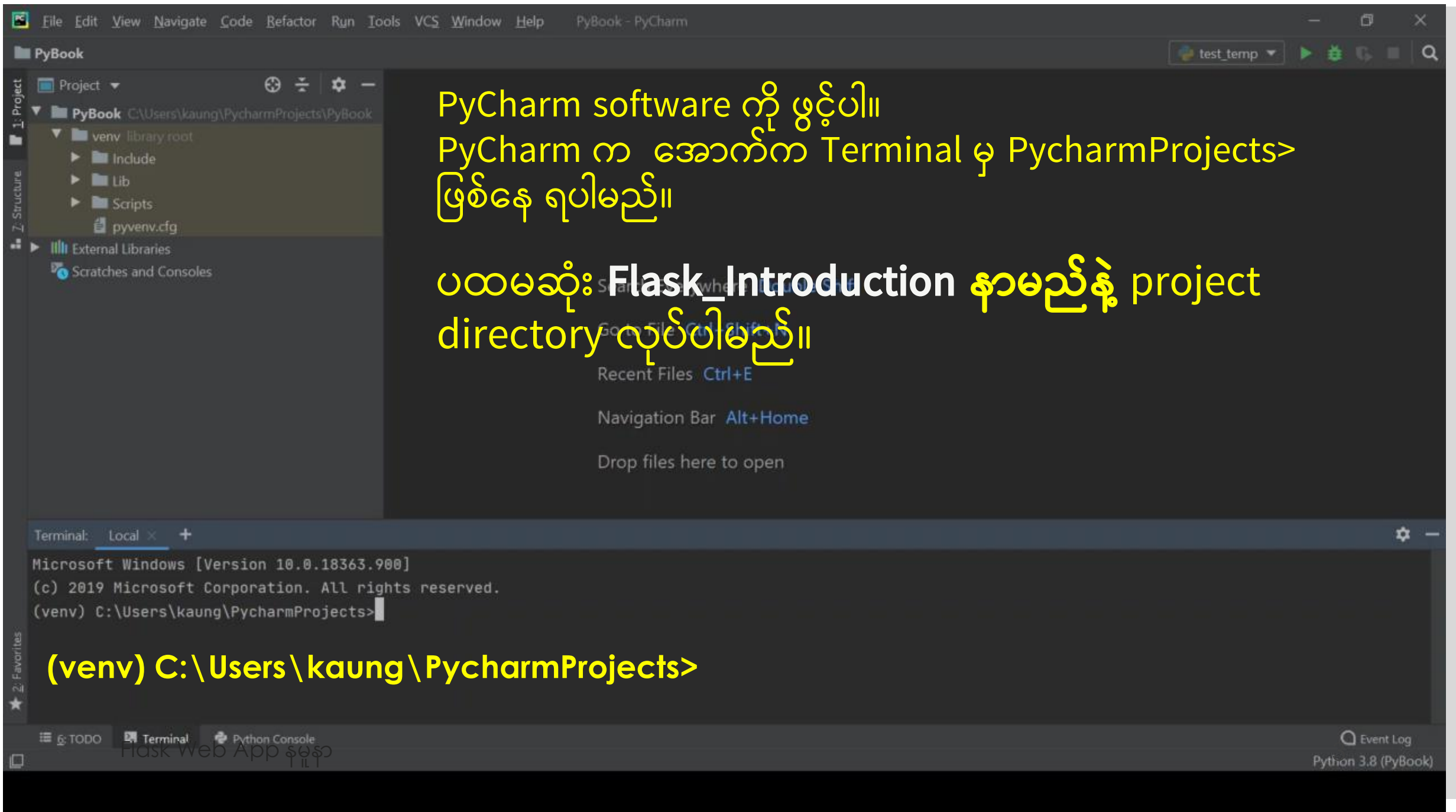
အပိုင်း(၁)

Flask Introduction

By Kaung Htat Nyunt

မာတိကာ

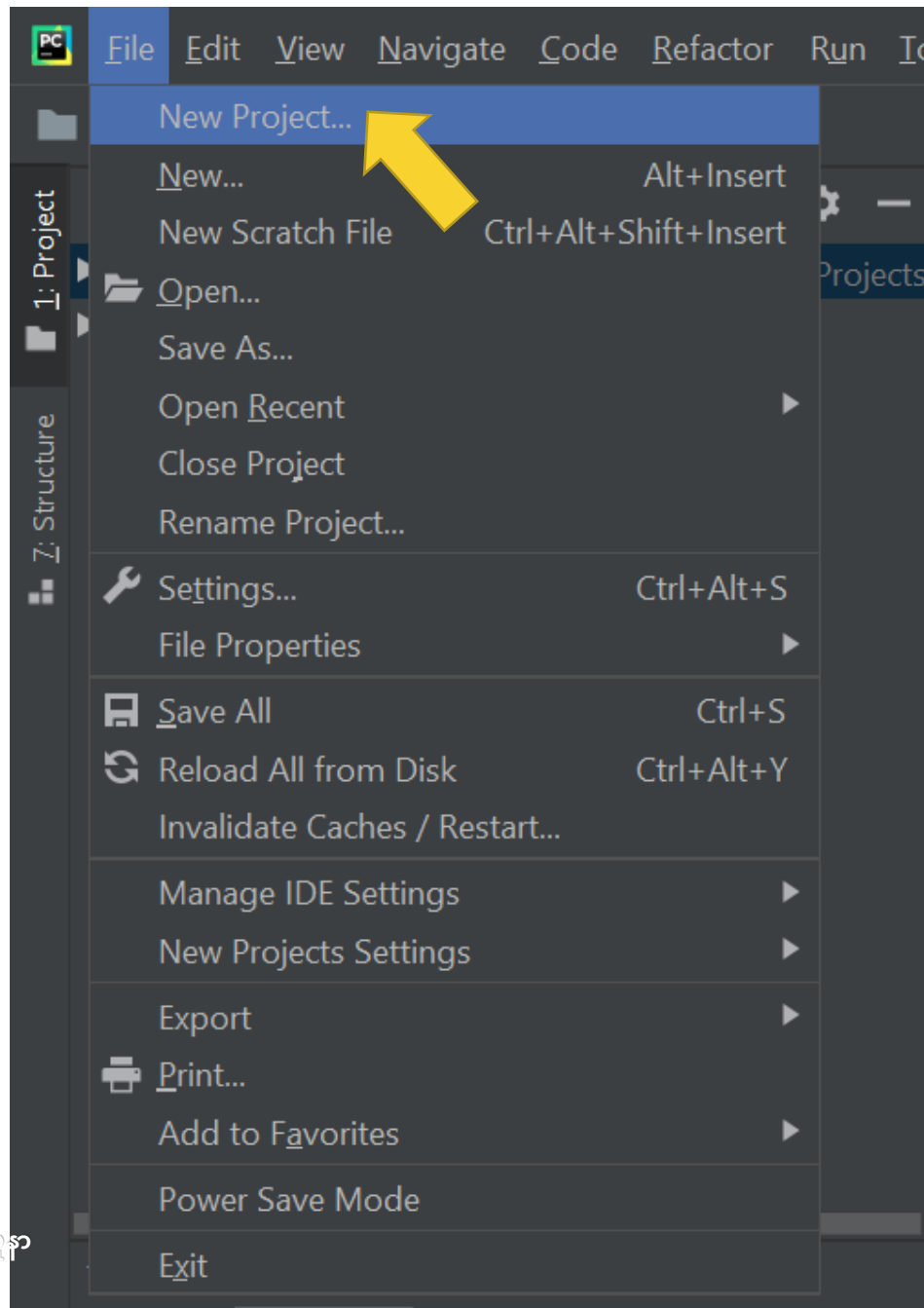
- Flask_Introduction ဆိုတဲ့ Python Project တစ်ခု တည်ဆောက် ပါမည်။
- app.py file တည်ဆောက်ပါမည်။
- app.py file နှင့် index.html တို့ ချိတ်ဆက် အလုပ်လုပ်ပုံ
- app.py file, index.html နှင့် base.html ချိတ်ဆက် အလုပ်လုပ်ပုံ တို့ကို ရှင်းပြသွားမည်။



PyCharm software ကို ဖွင့်ပါ။
PyCharm က အောက်က Terminal မှ PycharmProjects>
ဖြစ်နေ ရပါမည်။

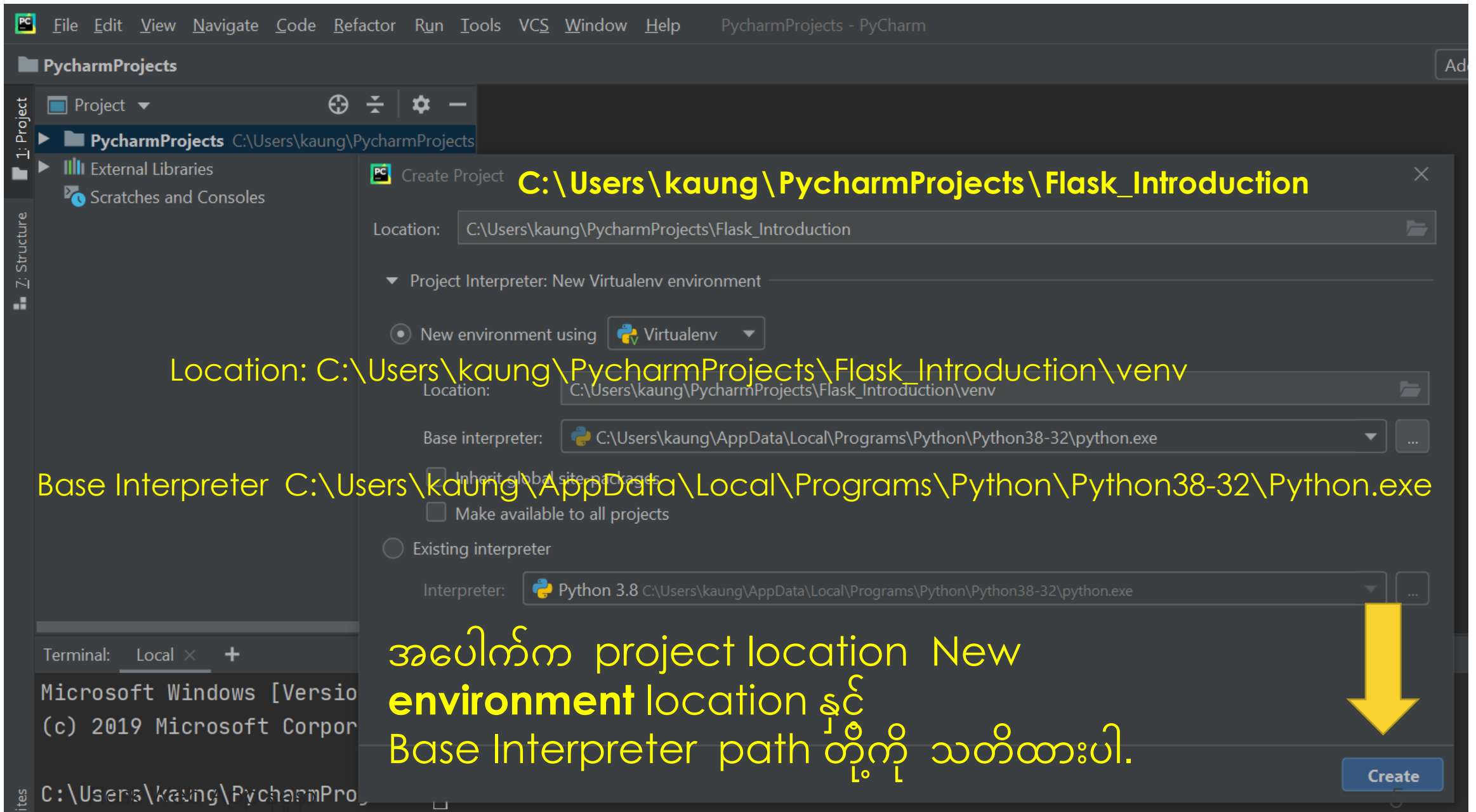
ပထမဆုံး **Flask_Introduction** နာမည်နဲ့ project
directory လုပ်ပါမည်။

(venv) C:\Users\kaung\PycharmProjects>



Flask_Introduction
နာမည်နဲ့ project director
လုပ်ရန်

File > New Project
ကို ကလစ်ပါ။



Successfully installed message and Flask installation

Creating Virtual Environment

Virtual Environment Successful မဖြစ်ရင် python တွေ run လို့ မရပါဘူး။

Successfully installed pip-20.1.1 setuptools-47.3.1

Terminal: Local x +

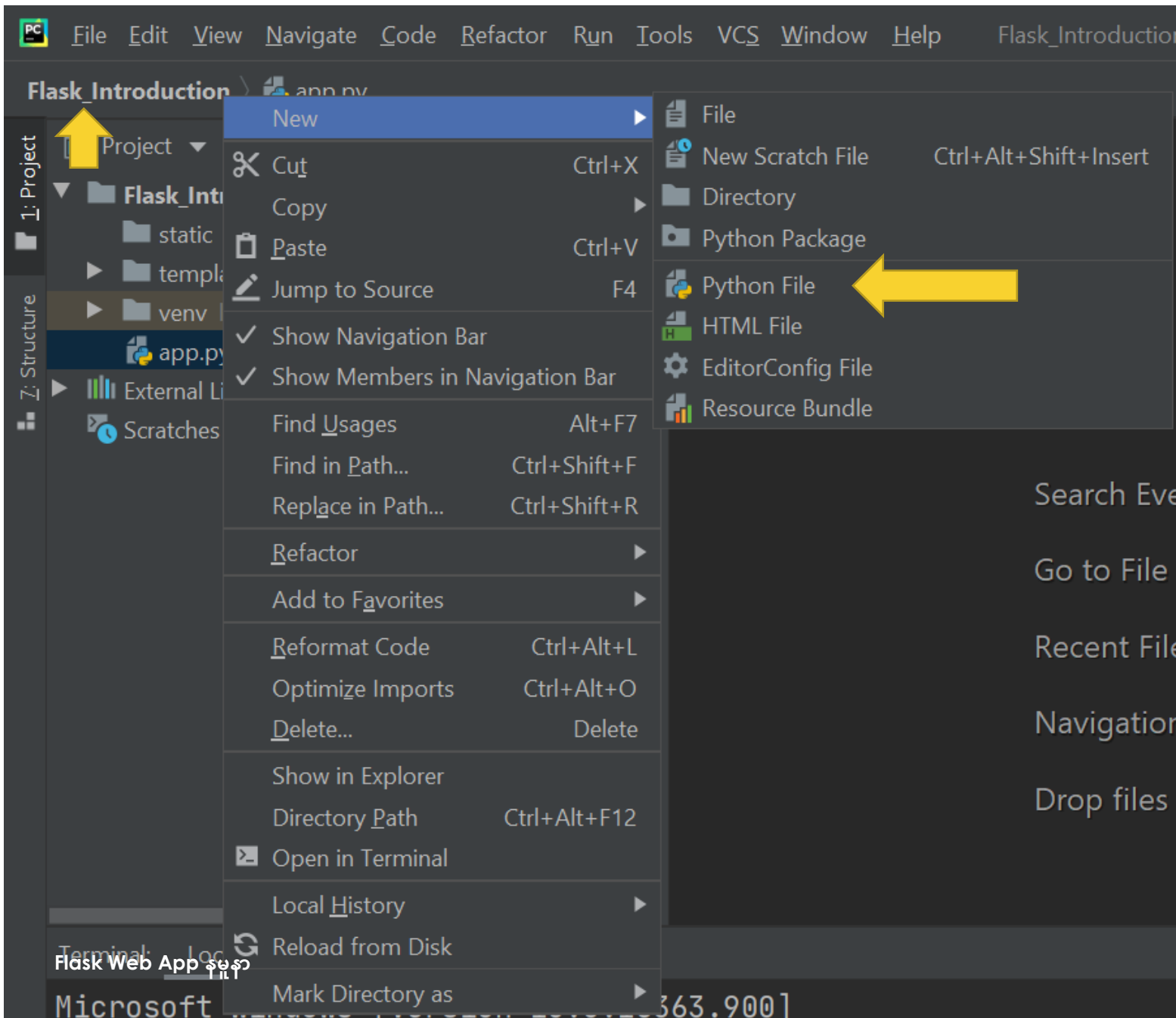
Terminal မှ

pip install Flask

```
(venv) C:\Users\kaung\PycharmProjects\Flask_Introduction>pip install Flask
Requirement already satisfied: Flask in c:\users\kaung\pycharmprojects\flask_introduction\venv\lib\site-packages (1.1.2)
Requirement already satisfied: click>=5.1 in c:\users\kaung\pycharmprojects\flask_introduction\venv\lib\site-packages (from Flask 1.1.2)
Requirement already satisfied: itsdangerous>=0.24 in c:\users\kaung\pycharmprojects\flask_introduction\venv\lib\site-packages (from Flask 1.1.2)
Requirement already satisfied: Werkzeug>=0.15 in c:\users\kaung\pycharmprojects\flask_introduction\venv\lib\site-packages (from Flask 1.1.2)
Requirement already satisfied: Jinja2>=2.10.1 in c:\users\kaung\pycharmprojects\flask_introduction\venv\lib\site-packages (from Flask 1.1.2)
Requirement already satisfied: MarkupSafe>=0.23 in c:\users\kaung\pycharmprojects\flask_introduction\venv\lib\site-packages (from Jinja2>=2.10.1->Flask) (1.1.1)
```

Flask install မလုပ်ရသေးရင် install လုပ်ပေးပါလိမ့်မည်။ Install

လုပ်ပြီးရင် Requirement already satisfied ပြမှာပါ။



App.py

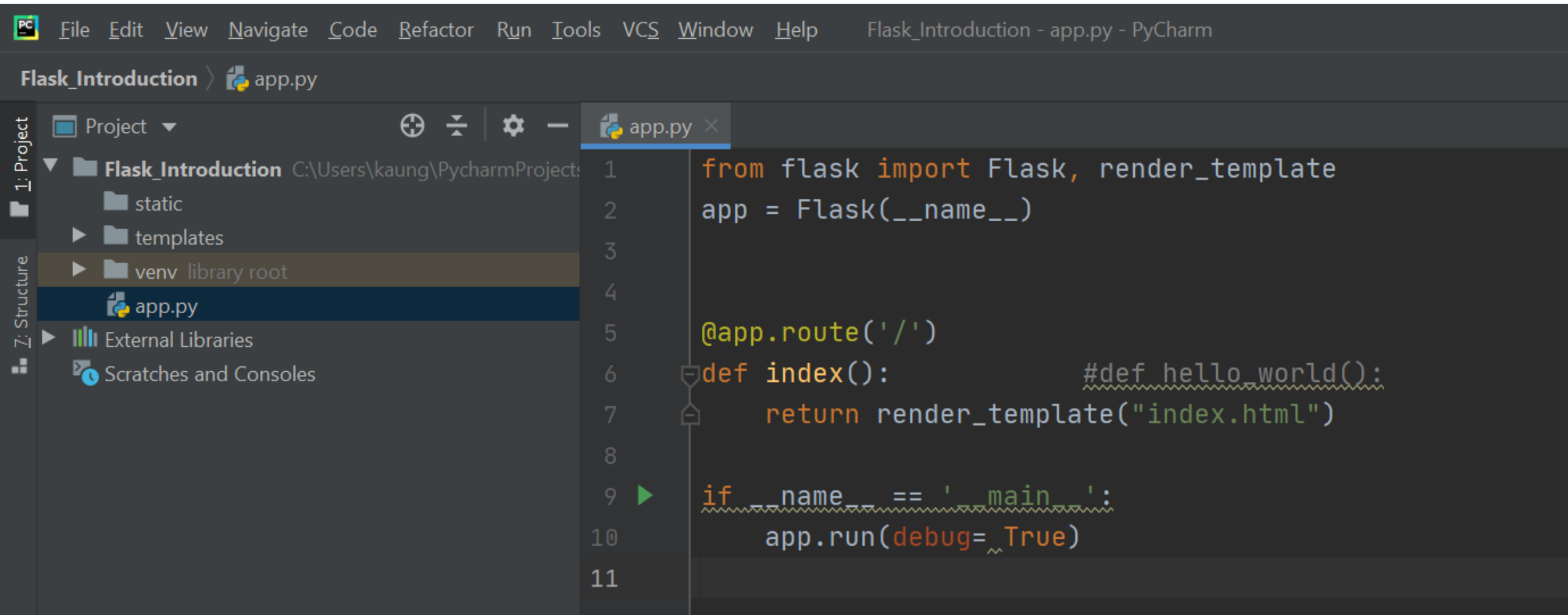
Flask_Introduction
ကို right click ခေါက်ပါ

New >

File >

Python File ကို ရွေးပါ။

app.py



The image shows a PyCharm IDE window titled "Flask_Introduction - app.py - PyCharm". The left sidebar displays the project structure for "Flask_Introduction", which includes a "static" folder, a "templates" folder, a "venv library root" folder, and the "app.py" file. The main editor area shows the code for "app.py" with line numbers 1 through 11. The code imports Flask and render_template, creates a Flask app, defines a route for the root path with an index function that renders "index.html", and runs the app in debug mode.

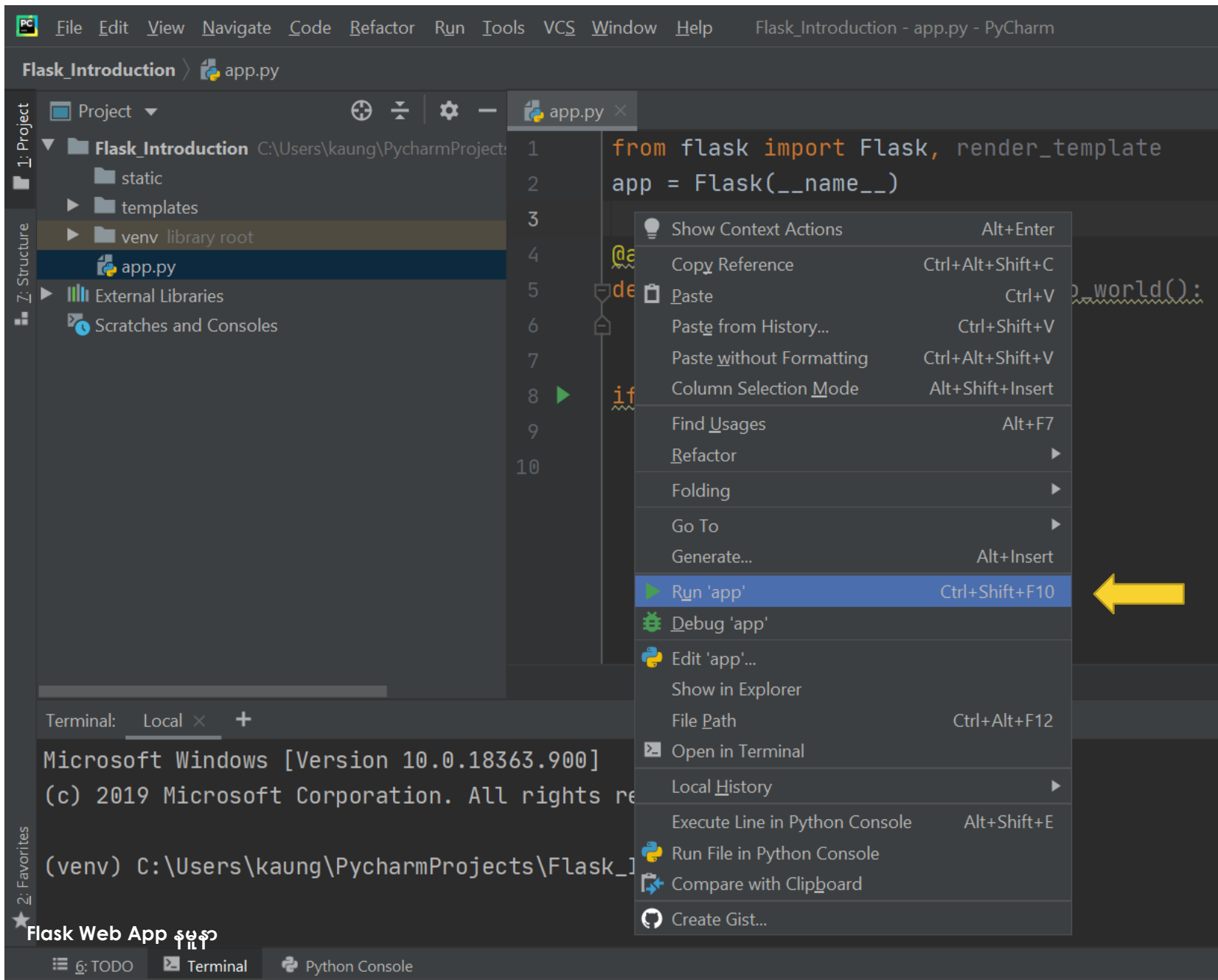
```
1 from flask import Flask, render_template
2 app = Flask(__name__)
3
4
5 @app.route('/')
6 def index():                                #def hello_world():
7     return render_template("index.html")
8
9
10 if __name__ == '__main__':
11     app.run(debug=True)
```

app.py

```
from flask import Flask, render_template
app = Flask(__name__)

@app.route('/')
def index():
    return "Hello, World"

if __name__ == '__main__':
    app.run(debug=True)
```

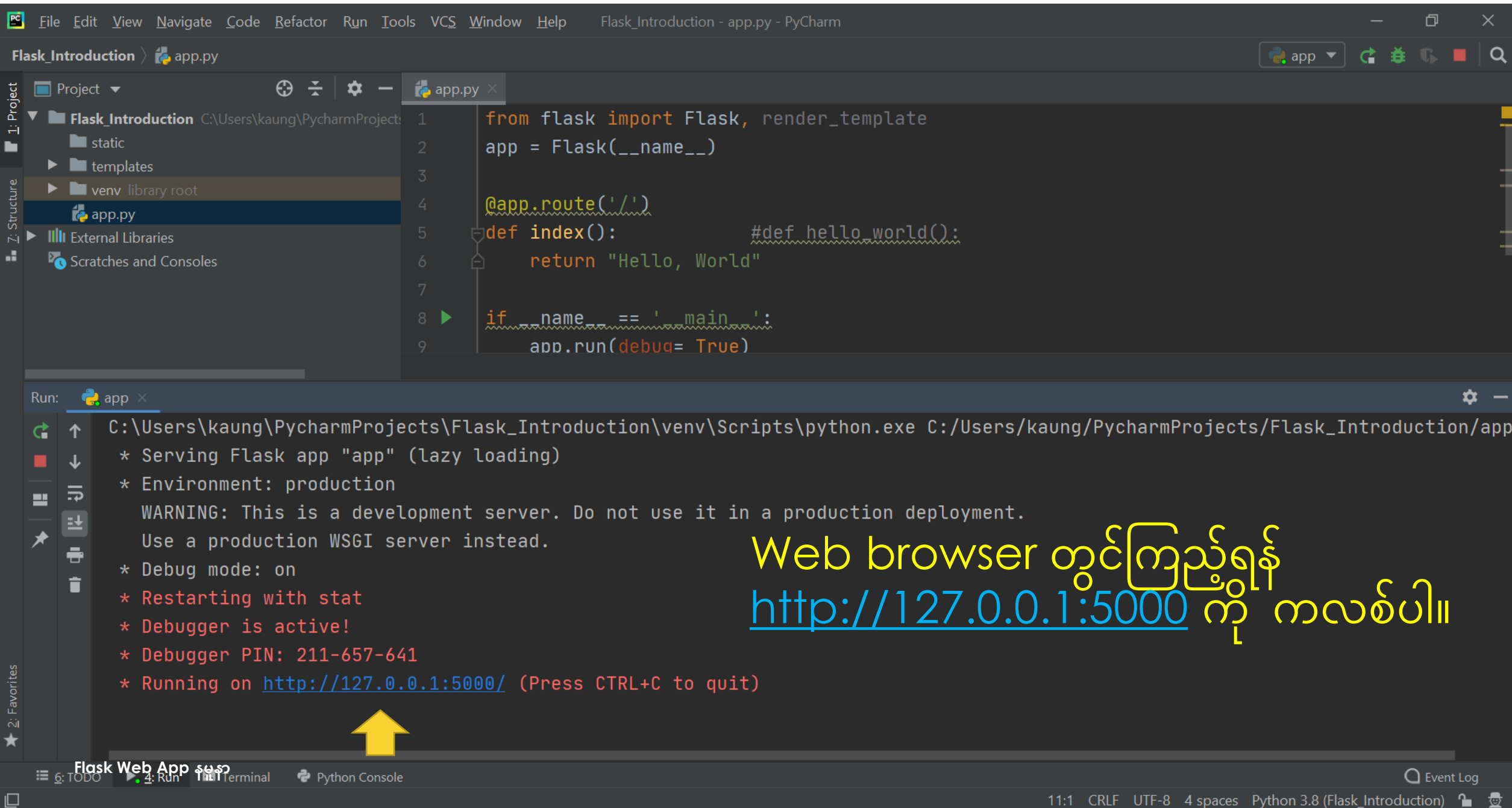


app.py ဖိုင်ကို
run ရန်

right click >
Run 'app'

သို့မဟုတ်

Ctrl+Shift+F10



Project

- Flask_Introduction
 - static
 - templates
 - venv library root
 - app.py
- External Libraries
- Scratches and Consoles

```
1 from flask import Flask, render_template
2 app = Flask(__name__)
3
4 @app.route('/')
5 def index(): #def hello_world():
6     return "Hello, World"
7
8 if __name__ == '__main__':
9     app.run(debug= True)
```

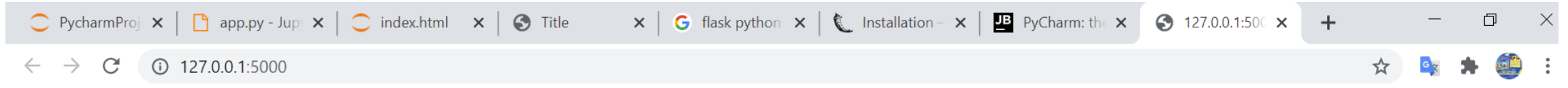
Run: app

C:\Users\kaung\PycharmProjects\Flask_Introduction\venv\Scripts\python.exe C:/Users/kaung/PycharmProjects/Flask_Introduction/app

- * Serving Flask app "app" (lazy loading)
- * Environment: production
- WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
- * Debug mode: on
- * Restarting with stat
- * Debugger is active!
- * Debugger PIN: 211-657-641
- * Running on <http://127.0.0.1:5000/> (Press CTRL+C to quit)

Web browser တွင်ကြည့်ရန် <http://127.0.0.1:5000> ကို ကလစ်ပါ။

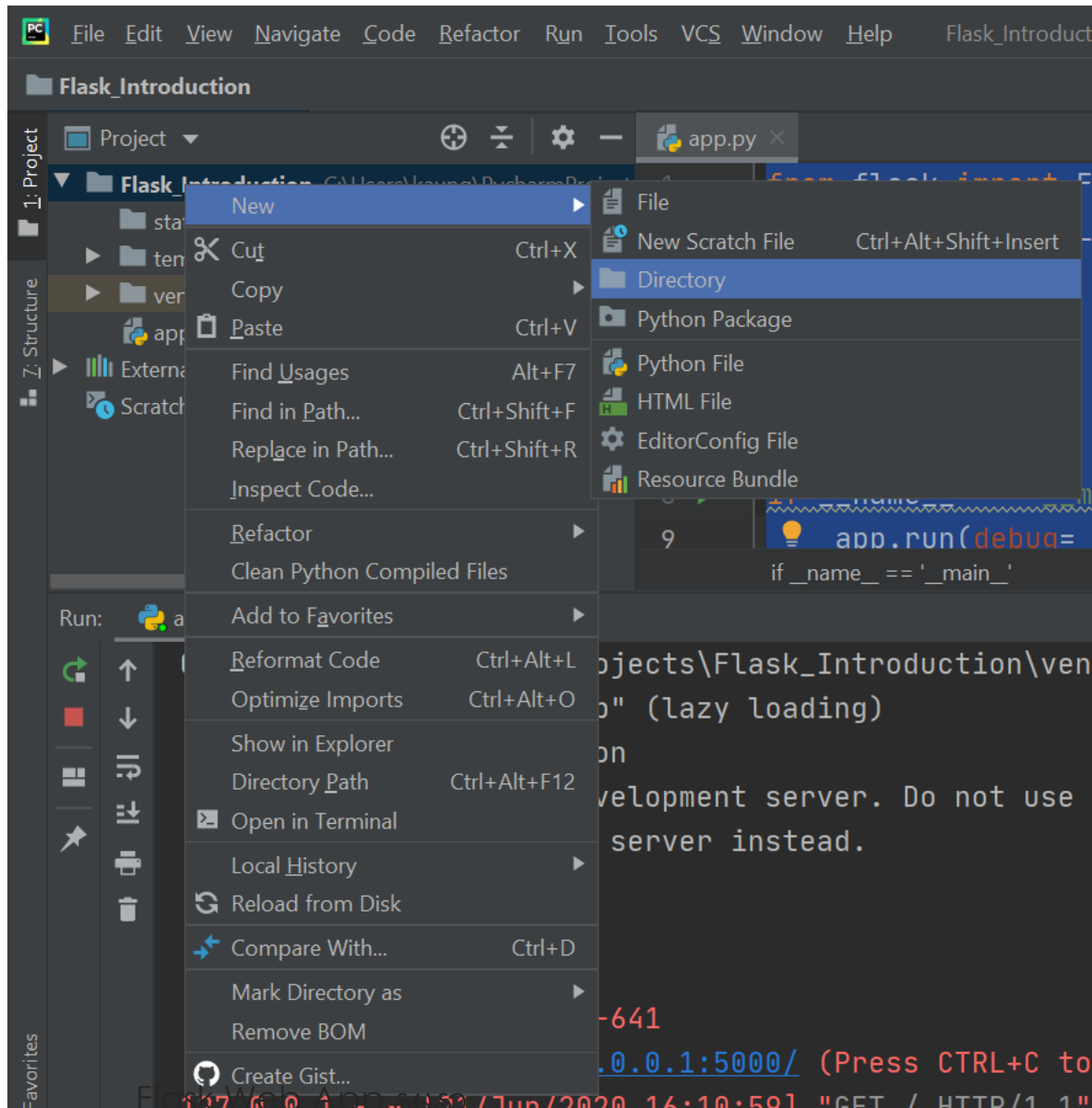
Flask Web App



ပထမအဆင့် အောင်မြင်သွားပါပြီ။

ဒုတိယအဆင့် အနေဖြင့် app.py file နှင့် index.html တို့ ချိတ်ဆက်အလုပ်လုပ်ပုံကို ဆက်ပါမည်။

static folder နှင့် templates folder နှစ်ခုကို ပြုလုပ်ပါမည်။

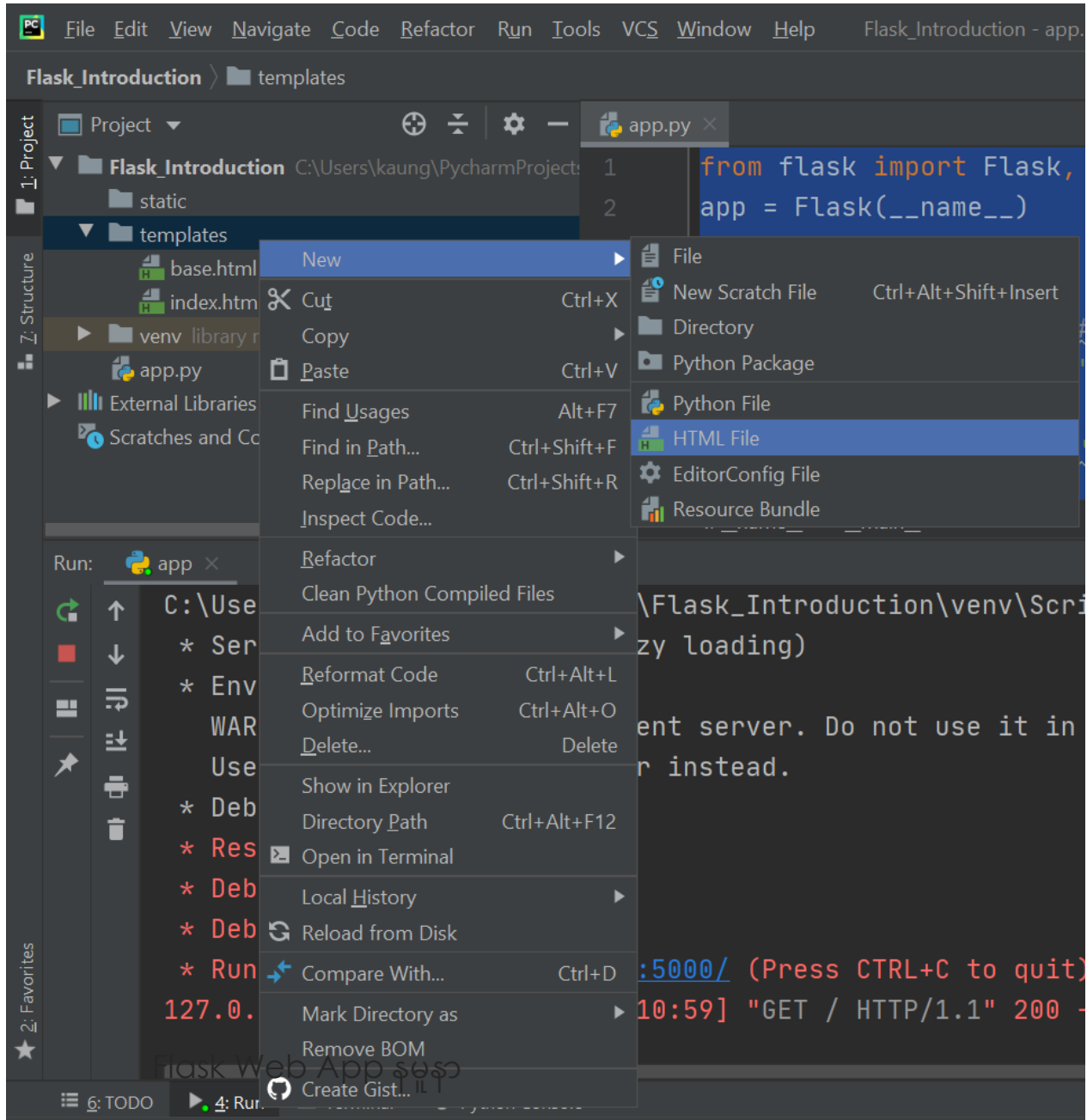


static folder နှင့် templates folder နှစ်ခုကို ပြုလုပ်ရန်

Flask_Introduction ဆိုတဲ့ project name အပေါ်တွင် right click ခေါက်ပါ။

New > Directory ကို click ပါ။

New Directory တွင် static folder နှင့် templates folder နာမည်ထည့်ပါ။



index.html ဖိုင် ပြုလုပ်ရန်
templates folder အပေါ်တွင်
right click
ခေါက်ပါ။

New >
HTML File ကို click ပါ။

HTML File ကို ရွေးပြီး
index.html နာမည်ထည့်ပါ။

index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Title</title>
  <h1> Python and Machine Learning for Myanmar Engineers"
    Facebook Page. </h1>
</head>
<body>
<p>This is an example paragraph.</p>
</body>
</html>
```

index.html နှင့် app.py တို့ချိတ်ရန် app.py ဖိုင်မှ ကုဒ်
(၁) ကြောင်း ပြင်ပါမည်။

```
from flask import Flask, render_template
app = Flask(__name__)
```

← render_template

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

```
    return
```

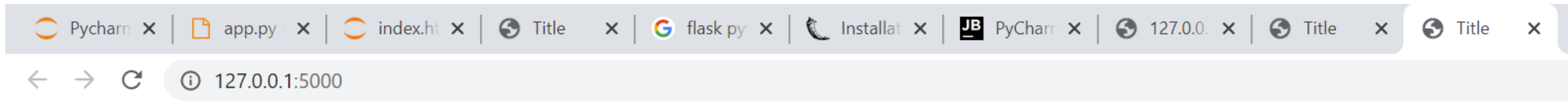
```
    render_template("index.html")
```

← Index.html ဖိုင် နှင့်ချိတ်ရန်

```
if __name__ == '__main__':
```

```
    app.run(debug=True)
```

Index.html ကိုလည်း လုပ်ပြီးပြီး app.py ကိုလည်း
ပြင်ပြီးပြီဆိုရင် run လို့ရပါပြီ။



Python and Machine Learning for Myanmar Engineers" Facebook Page.

This is an example paragraph.

အပေါ်ကလို ပေါ်တယ်ဆိုရင်တော့ ဒုတိယအဆင့်
အောင်မြင်သွားပါပြီ။

တတိယအဆင့် အနေတဲ base.html ဖိုင် index.html ဖိုင် နှင့် app.py ဖိုင် တွေကို ချိတ်ဆက်မှာပါ။

- base.html ဖိုင် နှင့် index.html ဖိုင် တို့က templates folder အောက်မှာ ရှိရပါမည်။
- အခု base.html ကို create လုပ်ပါမည်။
- base.html ဖိုင် ကို အခြေခံအဖြစ်ထားပြီးတော့ တခြားဖိုင်တွေက ဖော်ပြချင်တာတွေ သွားထည့်မှာပါ။ အခု ဥပမာမှာ index.html ဖိုင် ထဲက ဖော်ပြချင်တာတွေ သွားထည့်မှာပါ။
- app.py ထဲက `return render_template("index.html")` ကုဒ်ကို run တိုင်း index.html ဆီသွားမှာပါ။ index.html ကလည်း base.html ဖိုင် ထဲမှာ ရှိတာတွေကို အခြေခံထားပြီး သူ့ဖိုင်ထဲက ဖော်ပြချင်တာ(`display လုပ်ချင်တာ`) တွေကို web browser ပေါ်မှာ ဖော်ပြမှာပါ။

base.html

```
<!DOCTYPE html>
<html lang="en">
  <head>
<title> Python and Machine Learning for Myanmar Engineers" Facebook Page </title>
  </head>

  <body>
    {% block head%}  {% endblock%}
    <p>Hello "Python and Machine Learning for Myanmar Engineers" Facebook Page</p>

    {% block paragraph%}  {% endblock%}
  </body>
</html>
```

base.html ဖိုင်နှင့် Index.html ဖိုင်

```
<body>
{% block head%} {% endblock%}
<p>Hello "Python and Machine Learning for Myanmar Engineers" Facebook Page</p>

{% block paragraph%} {% endblock%}
```

```
{% block head%}           {% endblock%}
{% block paragraph%}      {% endblock%}
```



Flask Web App မှာ index.html ဖိုင်မှာ စားသားများ ဝင်လာပါမည်။

index.html ဖိုင်ကို base.html ဖိုင်နှင့် ချိတ်ရန် ပြင်ရမည်။

```
{% extends 'base.html' %}
```

```
{% block head%
```

```
<h1> Python and Machine Learning for Myanmar Engineers" Facebook Page. This is from block head</h1>
```

```
{% endblock%
```

```
{% block paragraph%
```

```
<p>This is an example paragraph. This is from block paragraph.</p>
```

```
{% endblock%
```

<h1> Python and Machine Learning for Myanmar Engineers" Facebook Page. This is from block head</h1> သည်

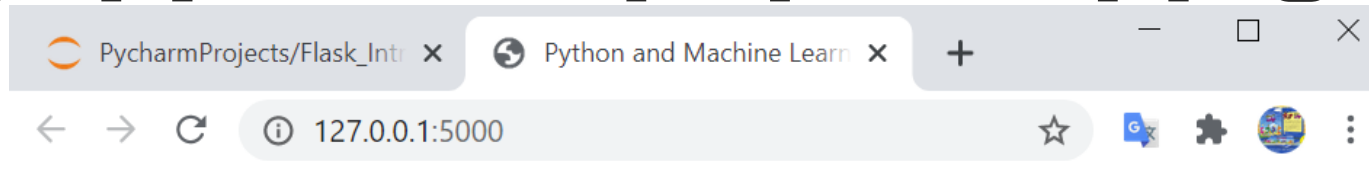
base.html ဖိုင်၏ **{% block head%** **{% endblock%** အကြားသို့ ဝင်ရောက် သွားလိမ့်မည်။

ထို့အတူ <p>This is an example paragraph. This is from block paragraph.</p> လည်း

base.html ဖိုင်၏ **{% block paragraph%** **{% endblock%** အကြားသို့ ဝင်ရောက်

သွားလိမ့်မည်။

base.html မှိုင် လည်း ထည့်ပြီးပြီ index.html မှိုင် ကိုလည်း
ပြင်ပြီးဆိုရင် app.py မှိုင်ကို Run လို့ရပါပြီ။



**Python and Machine Learning for Myanmar
Engineers" Facebook Page. This is from
block head**

Hello "Python and Machine Learning for Myanmar Engineers" Facebook Page

This is an example paragraph. This is from block paragraph.

အခုလို web browser မှာ မြင်ရတယ်ဆိုရင် တတိယအဆင့် အောင်မြင်ပါပြီ။
သင့်ဟာ python နဲ့ web app တစ်ခု စတင်ဘို့ ခြေတစ်လှမ်း စမိပါပြီ။

- End -