Guideline setup project Suspicious Human Activity Detection

Project bao gồm hai môi trường anaconda là notebook và web_app. Guideline sẽ giới thiệu về việc cài đặt môi trường, chạy notebook cũng như khởi chạy ứng dụng.

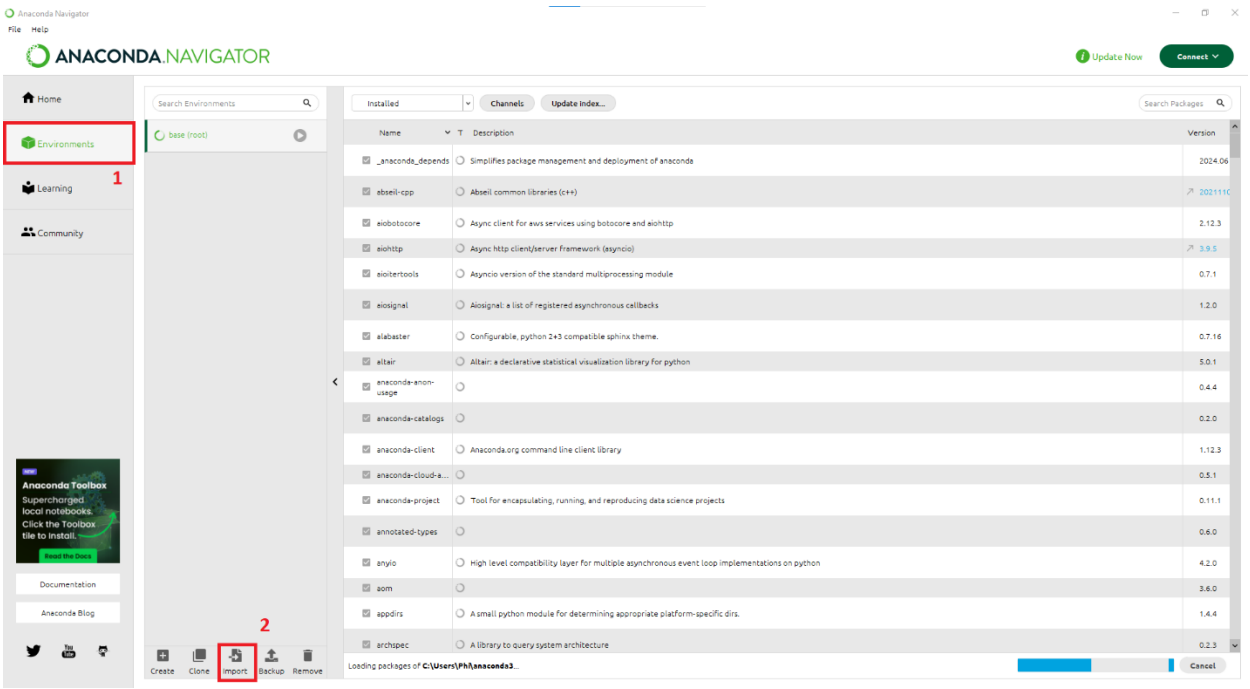
Prerequisite environment:

- *Anaconda*
- *PyCharm Professional Edition*

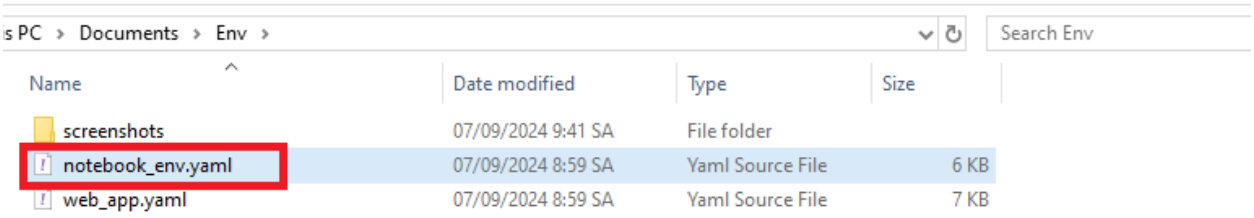
1. Cài đặt môi trường

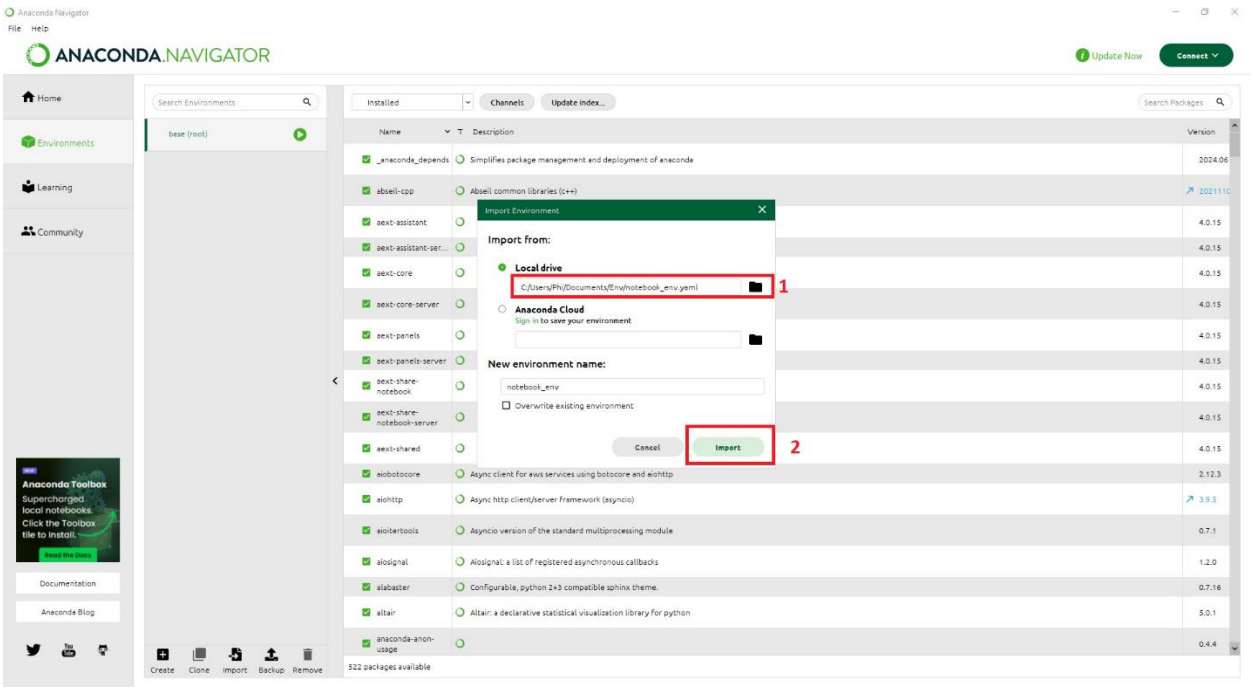
1.1. Import môi trường cho notebook

Vào Environments của Anaconda Navigator.

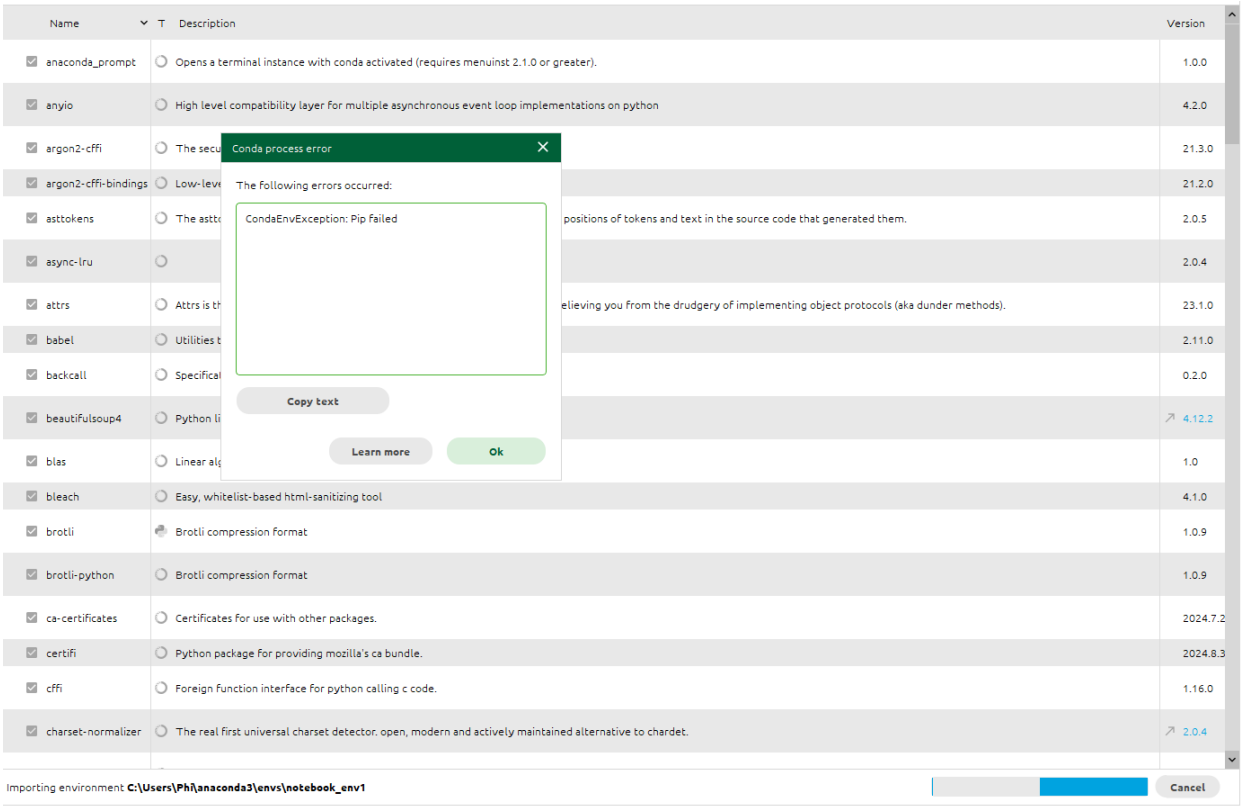


Chọn import file environment: **notebook_env.yaml**.



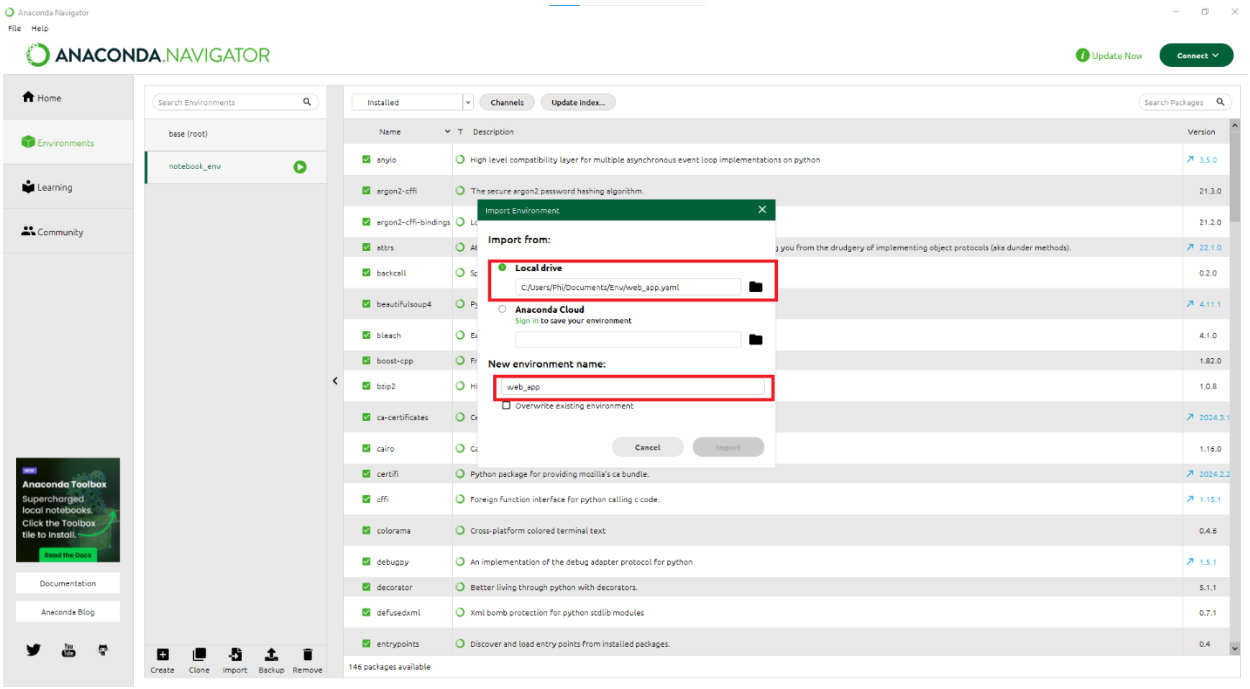


Nếu có xuất hiện lỗi như màn hình bên dưới, vui lòng tiếp tục, các thư viện bị lỗi đã khai báo trong các bước tiếp theo, đã đảm bảo dependencies đã đầy đủ.



1.2. Import môi trường cho web app

Thực hiện tương tự với môi trường **web_app**.



1.3. Phần mềm hỗ trợ

Cài đặt tool cmd convert video hỗ trợ là ffmpeg. Ta thực hiện cài đặt chocolatey là phần mềm quản lý cài đặt phần mềm và phần mềm ffmpeg.

Cài đặt chocolatey

Truy cập: https://chocolatey.org/install

Ta thực hiện bật Powershell bằng quyền administrator và chạy lệnh:

```
Set-ExecutionPolicy Bypass -Scope Process -Force;
[System.Net.ServicePointManager]::SecurityProtocol =
[System.Net.ServicePointManager]::SecurityProtocol -bor 3072; iex ((New-Object
System.Net.WebClient).DownloadString('https://community.chocolatey.org/install.ps1'))
```

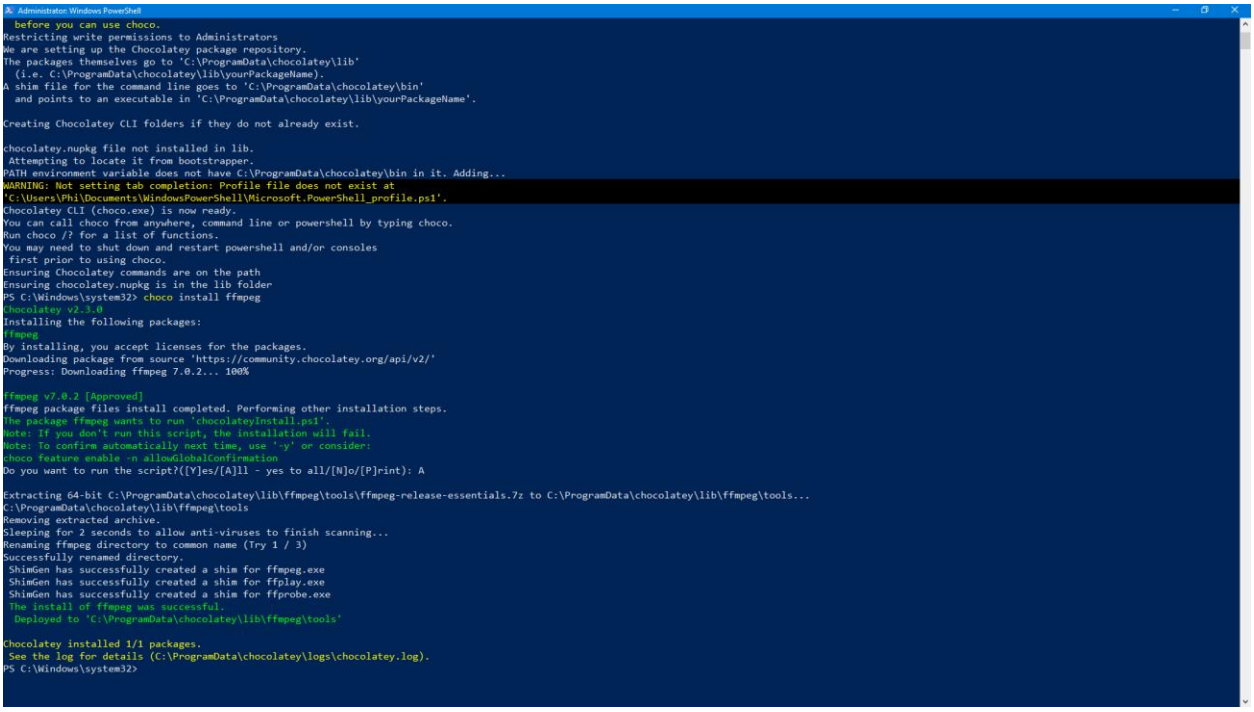
Cài đặt ffmpeg

Truy cập: https://www.ffmpeg.org/download.html

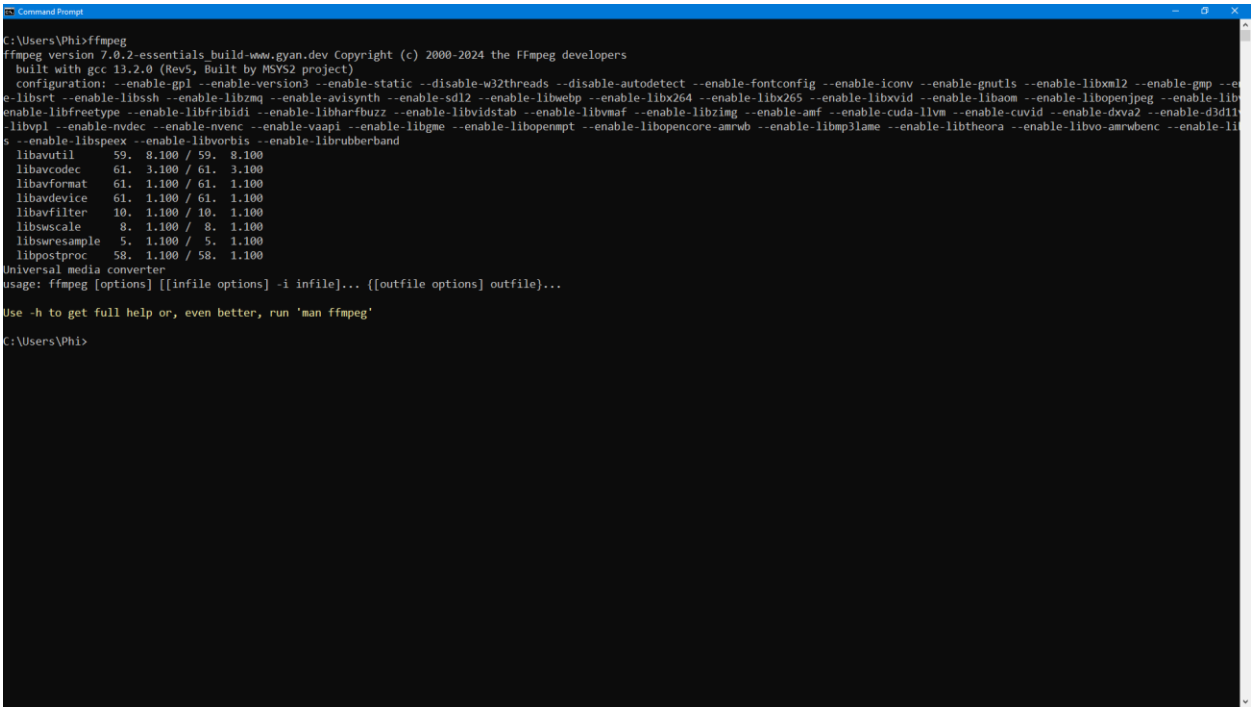
Ta thực hiện bật Powershell bằng quyền administrator và chạy lệnh:

```
choco install ffmpeg
```

Nhấn A hoặc Y để chấp nhận chạy script.

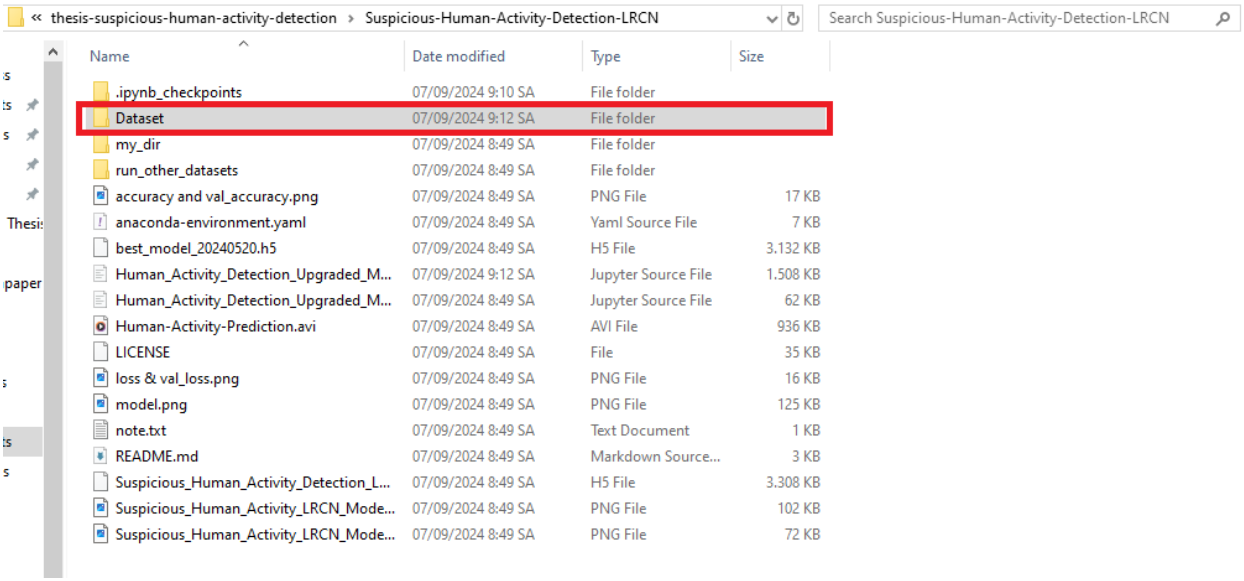


Sau khi cài đặt thành công thực hiện chạy thử bằng cách bật cmd và gõ lệnh ffmpeg, ta sẽ nhận được kết quả:

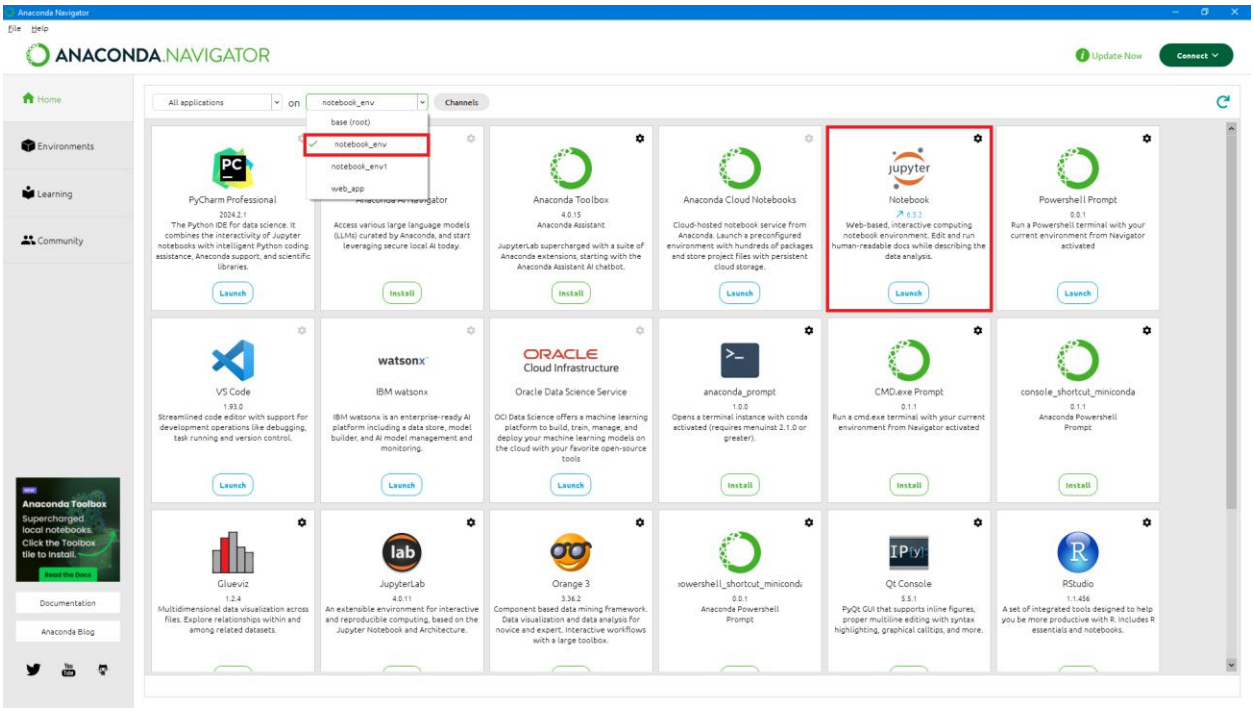


2. Notebook

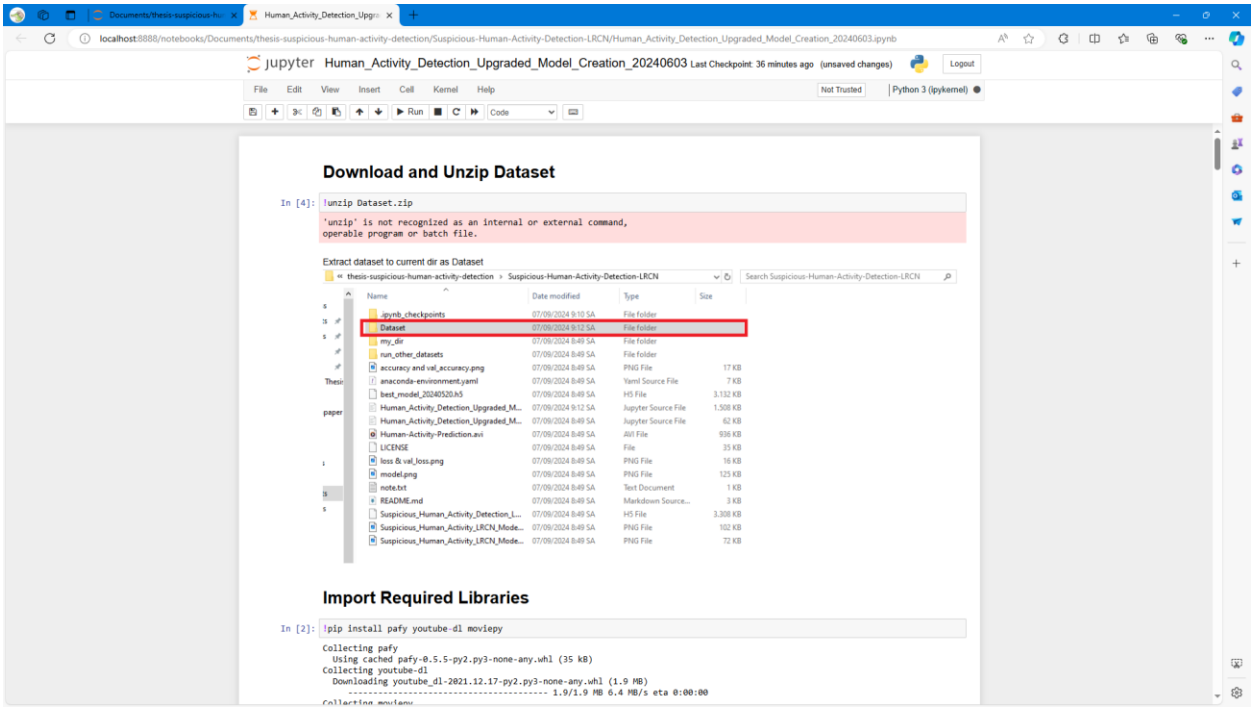
Thực hiện chép dataset vào folder notebook là Suspicious-Human-Activity-Detection-LRCN, ta có kết quả như hình.



Mở notebook từ Anaconda Navigator.

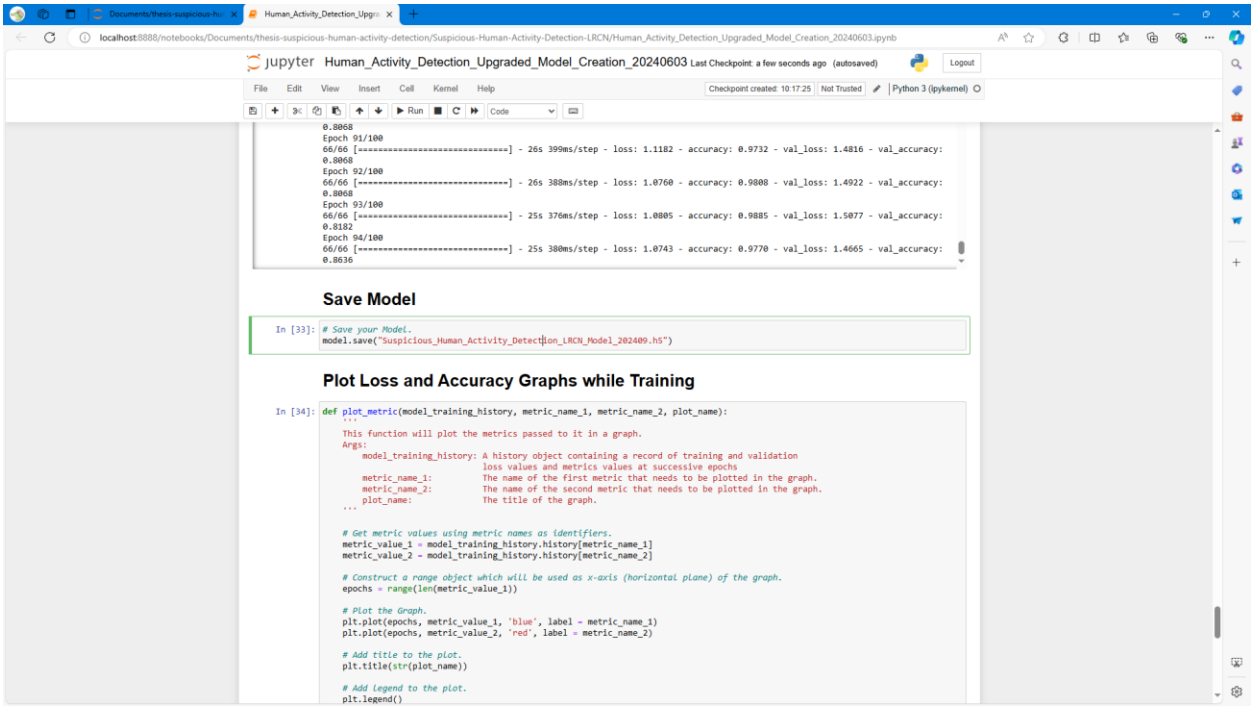


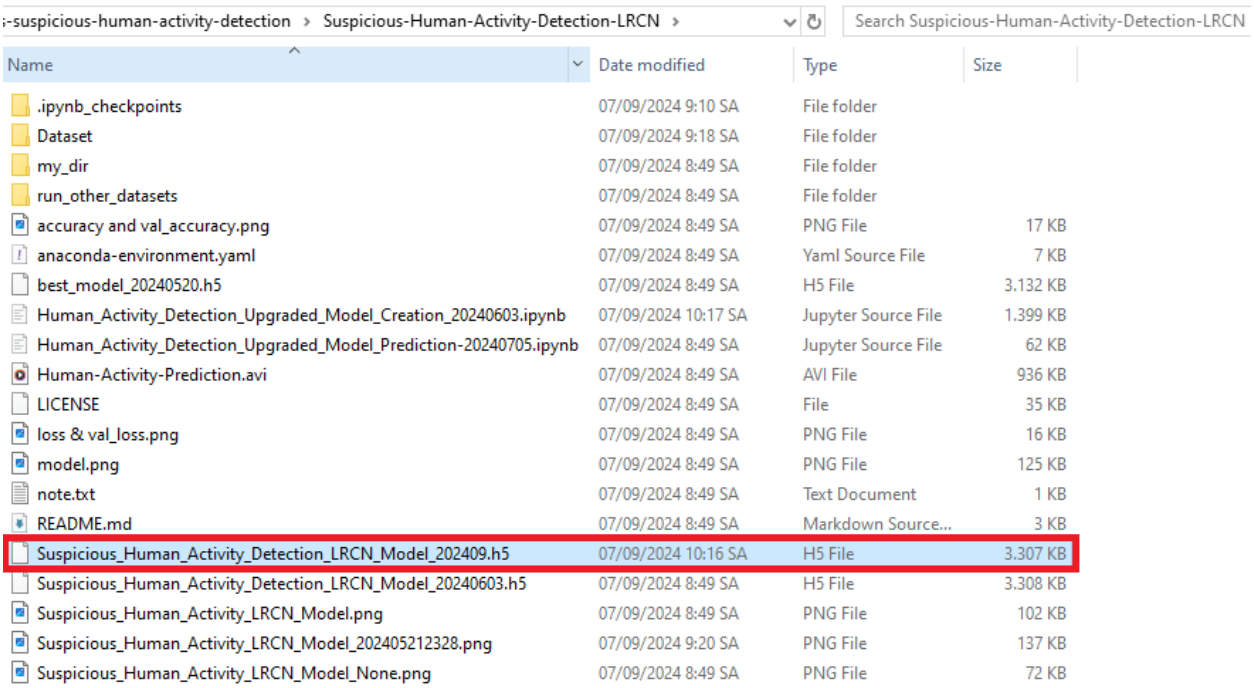
Ta thực hiện mở file notebook: *Suspicious-Human-Activity-Detection-LRCN/Human_Activity_Detection_Upgraded_Model_Creation_20240603.ipynb*.



Thực hiện chạy các cell trong notebook (bao gồm cả cài đặt các thư viện cần thiết).

Thực hiện lưu lại model đã huấn luyện.



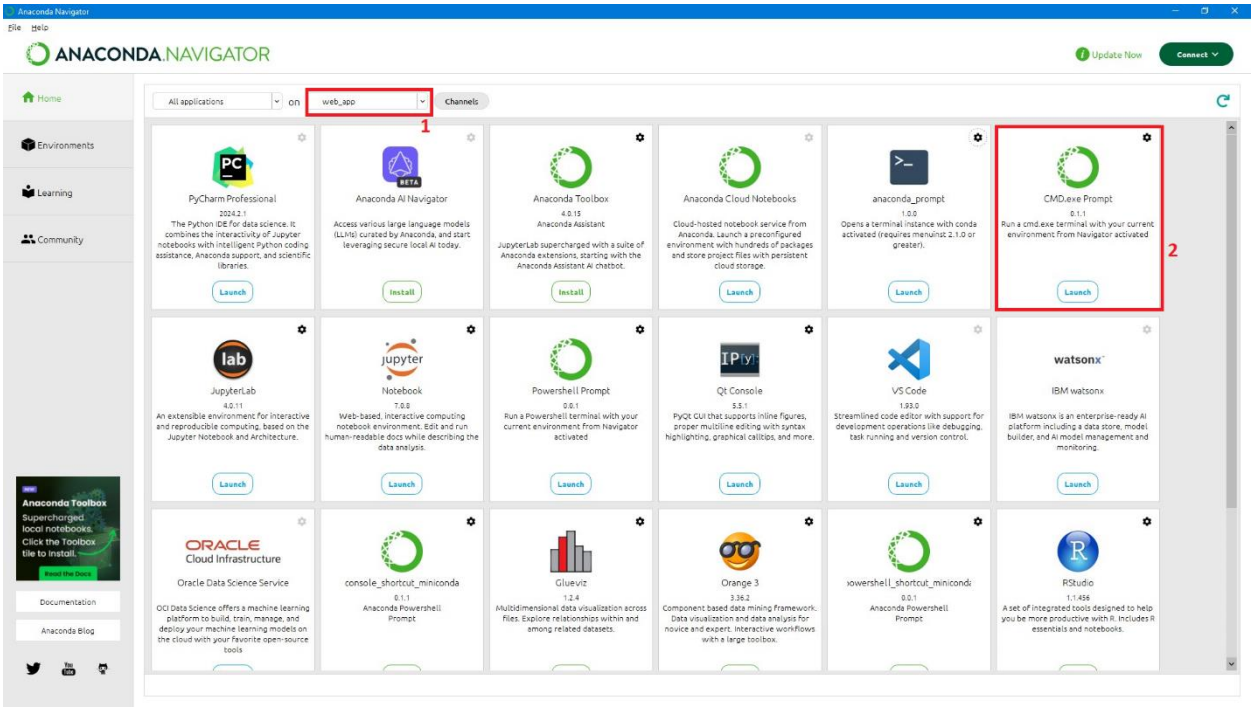


3. Web app

3.1. Cài đặt dependencies

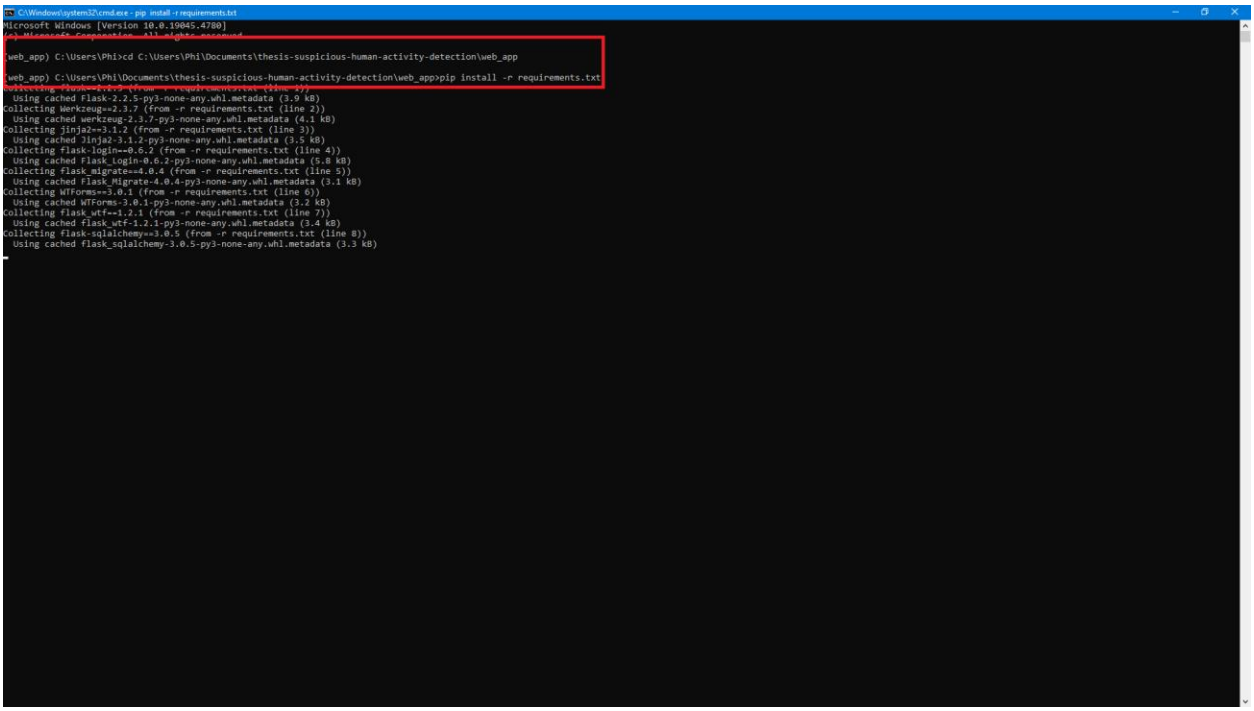
Đầu tiên, ta thực hiện cài đặt dependencies cho web app.

Chuyển sang môi trường web_app trong conda:

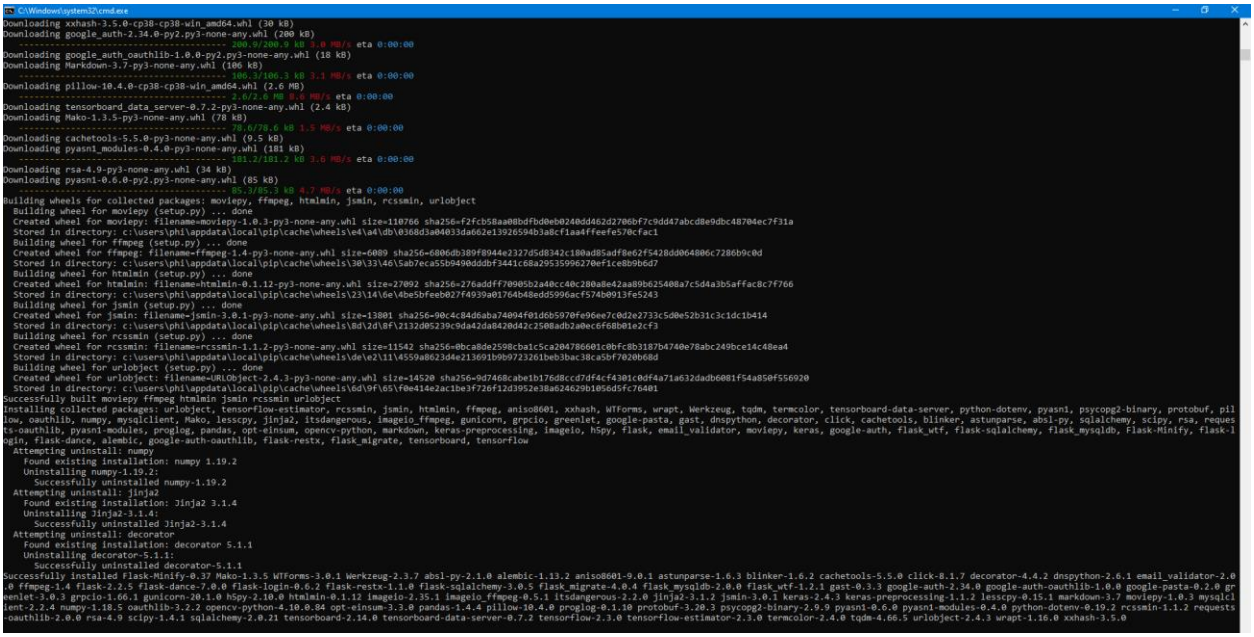


Sau đó thực hiện bật CMD Prompt, kiểm tra môi trường đã chuyển sang web_app của anaconda.

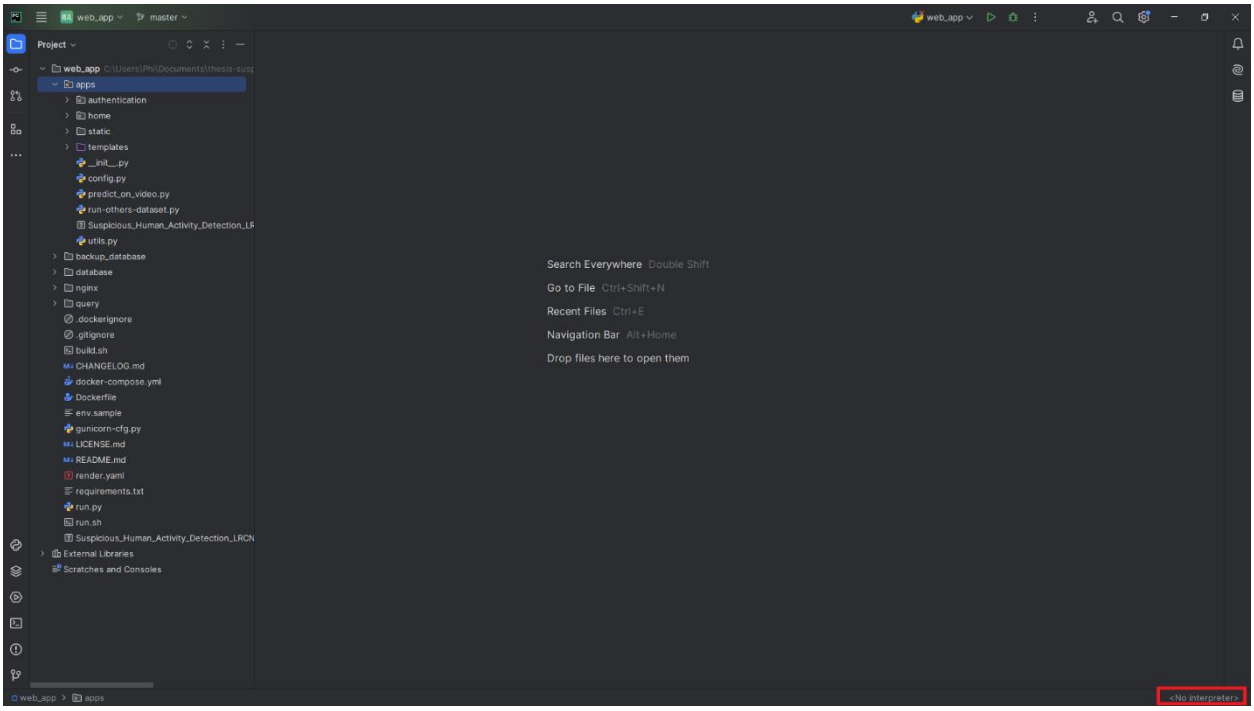
Thực hiện đổi thư mục tới folder web_app trên máy và thực hiện cài đặt dependencies đã khai báo trong file requirement.txt bằng lệnh: **`pip install -r requirements.txt`**



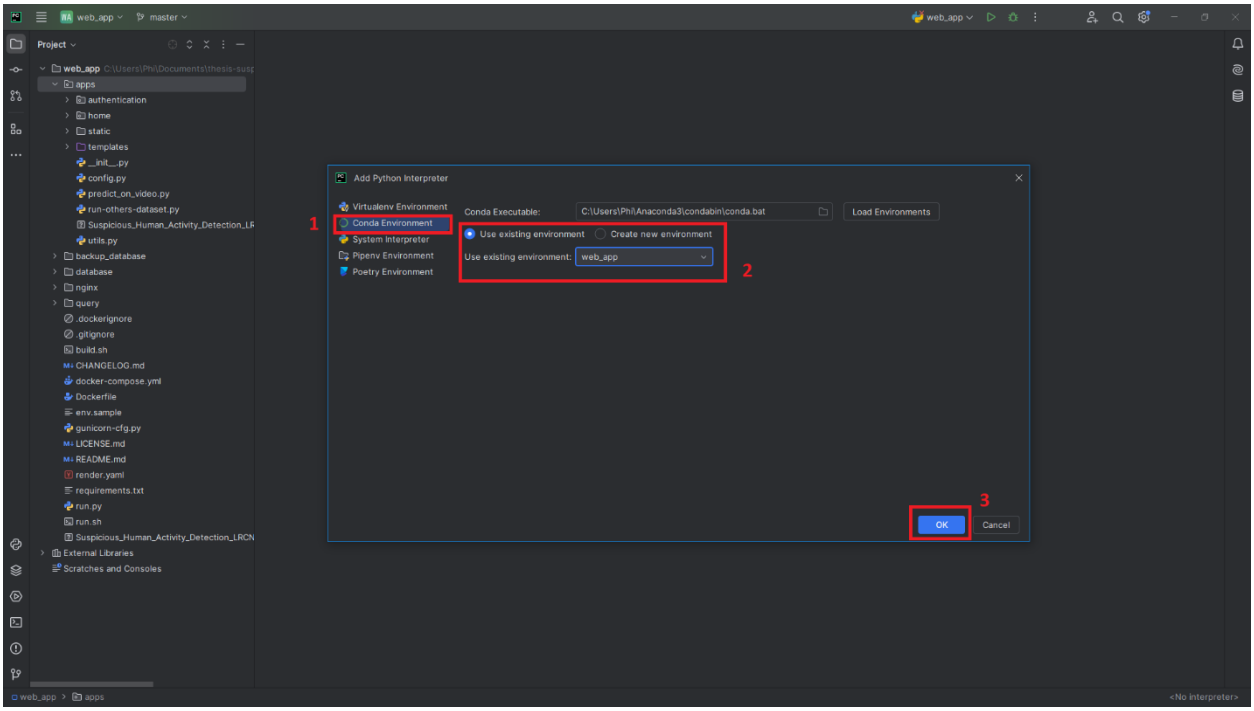
Thực hiện cài đặt các dependencies thành công:



Mở project web_app với Pycharm, nhấn vào mục configure interpreter.

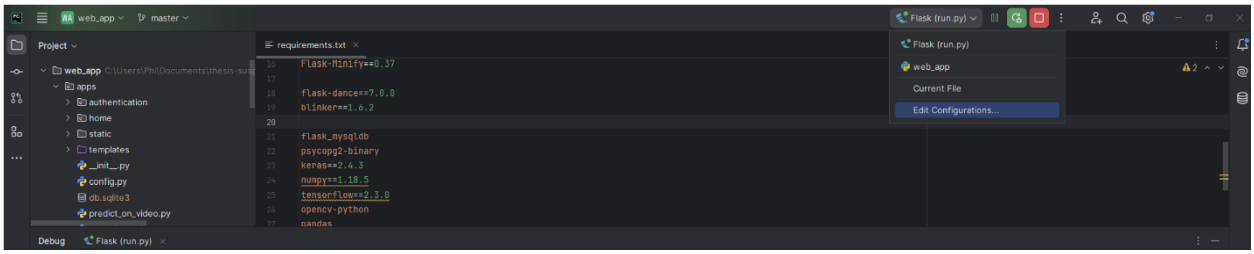


Thực hiện chọn môi trường conda là web_app.

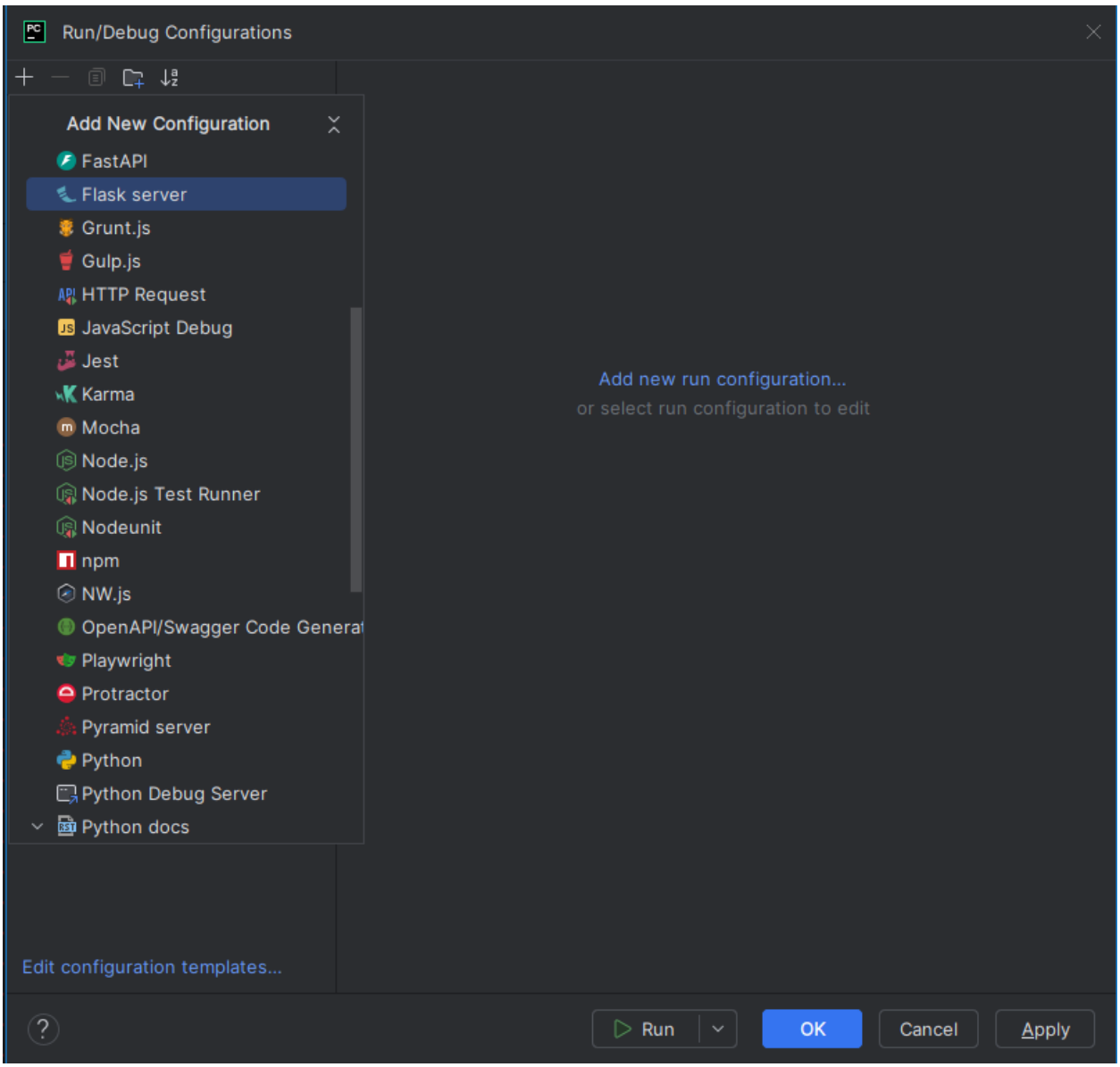


3.2. Chạy debug mode với Pycharm

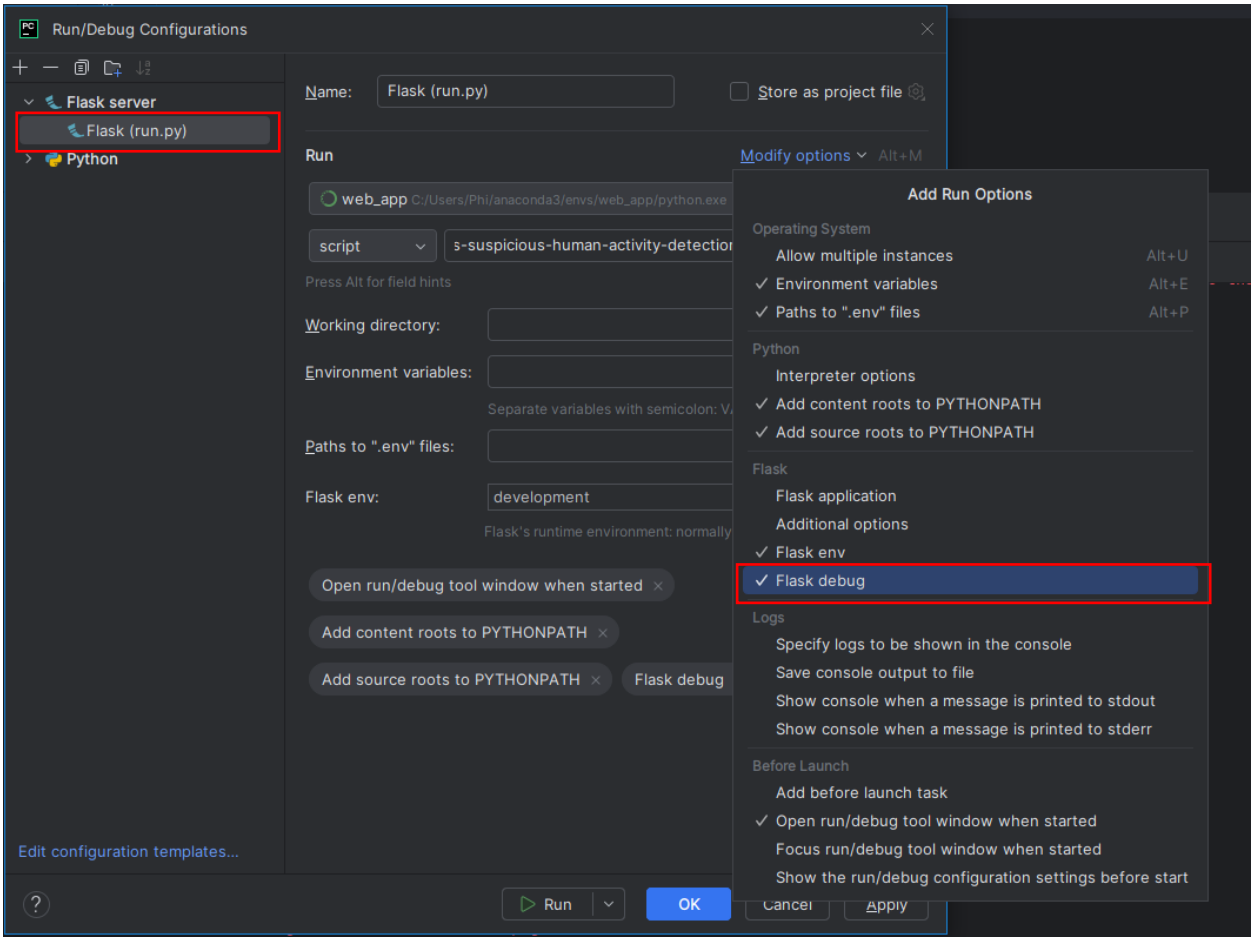
Thực hiện chon run configuration trong Pycharm:

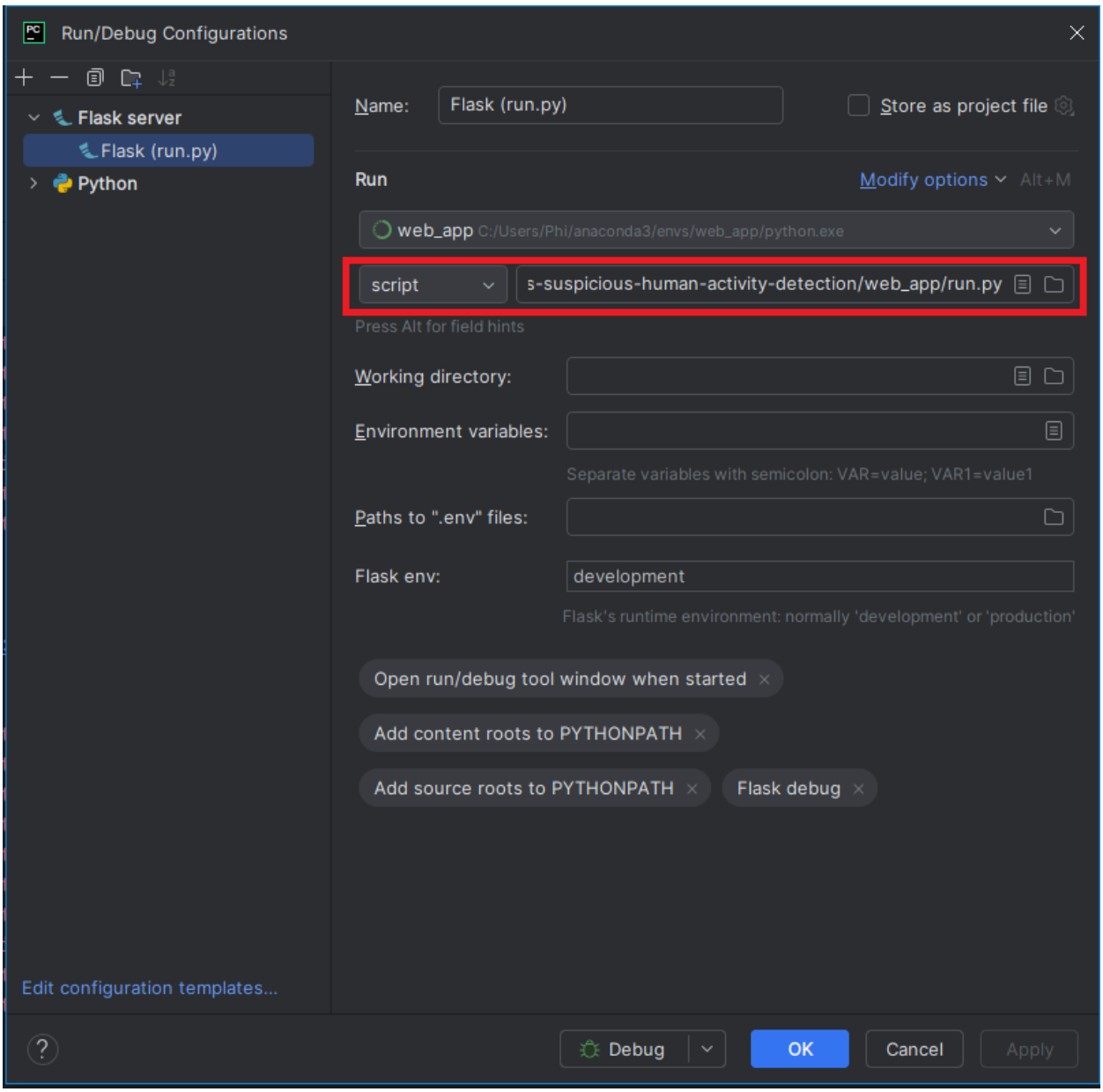


Nhấn vào dấu + và chọn Flask server:



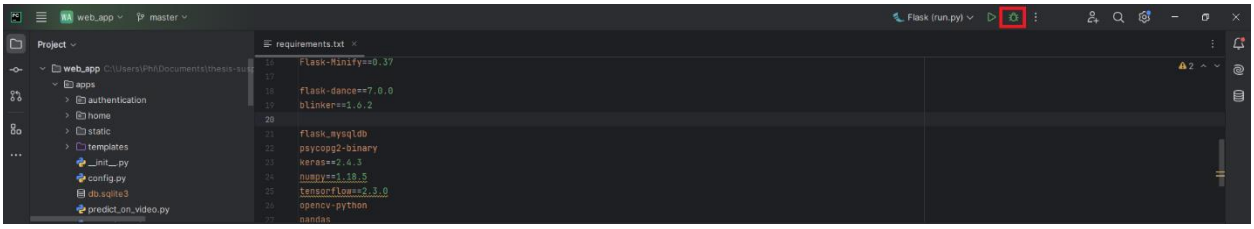
Chọn vào *Modify options* chọn Flask debug để bật cờ debug.



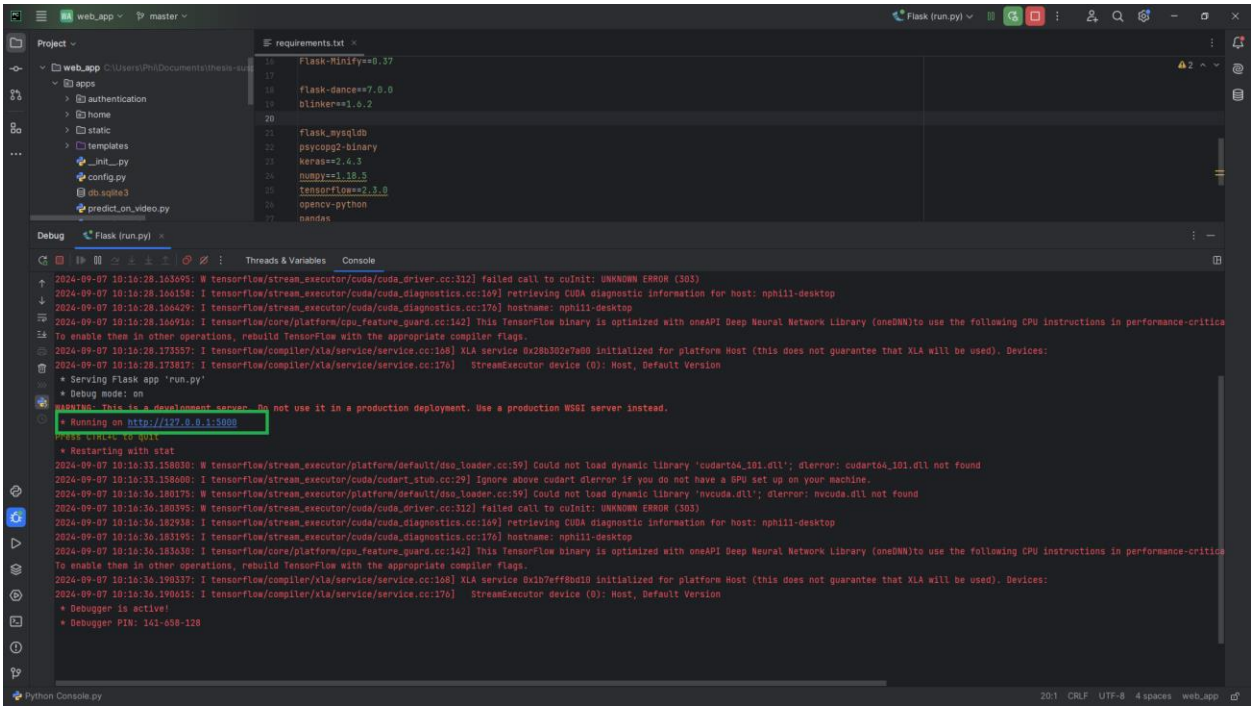


Chọn startup là script với file: ***web_app/run.py***.

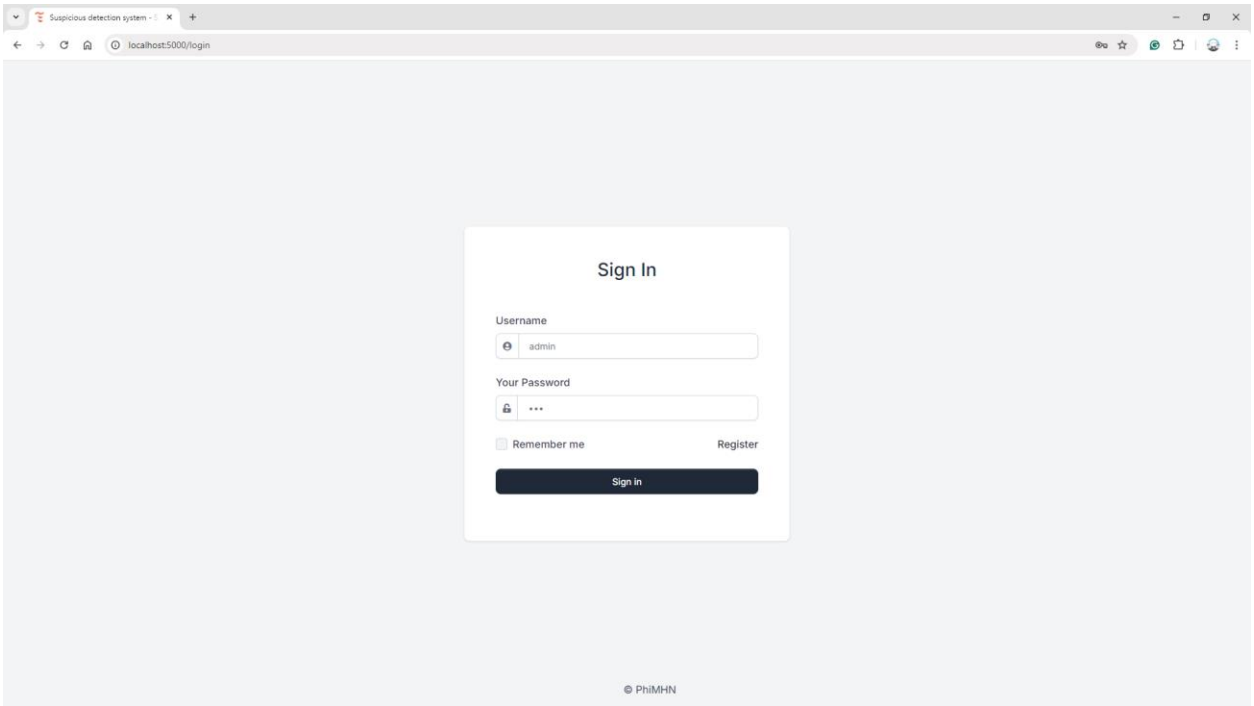
Nhấn vào nút debug.



Ứng dụng được chạy thành công.



Màn hình ứng dụng.



Default user:

Username: admin; Password: 123

3.3. Chạy ứng dụng bằng command line

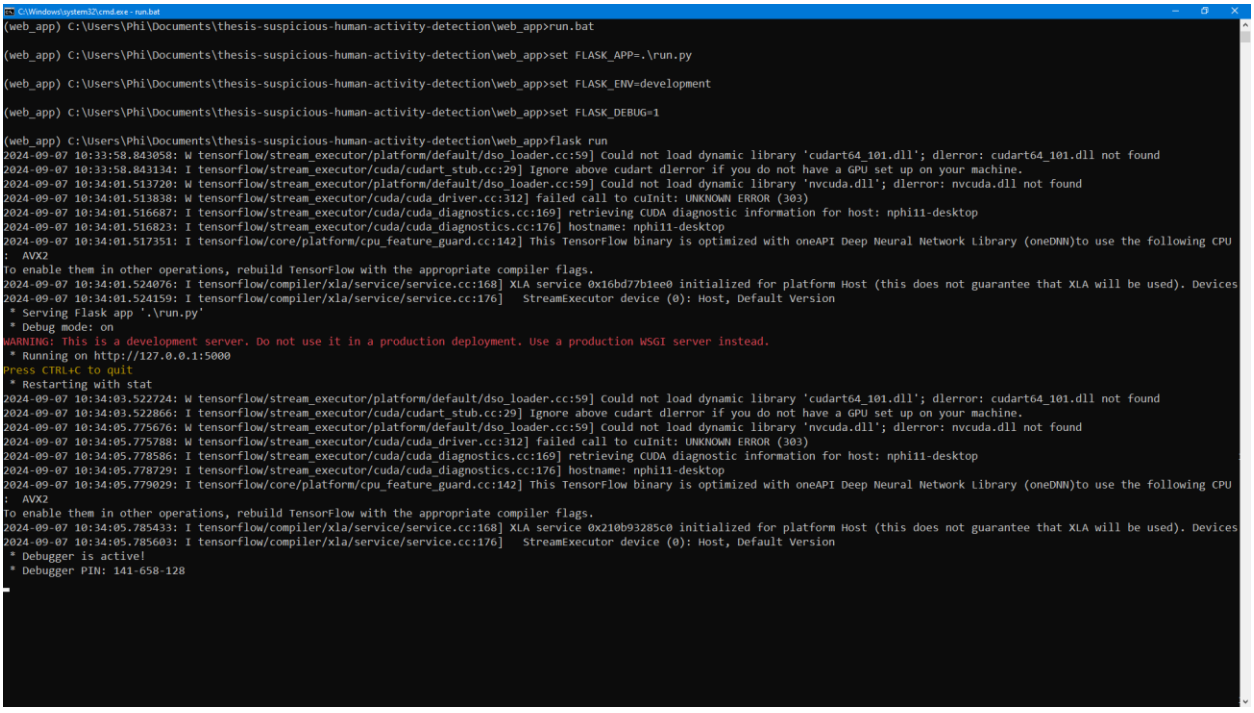
Bật CMD prompt như bước 3.1.

Mở folder web_app chưa ứng dụng, chạy command: ***run.bat***.

Nội dung file run.bat:

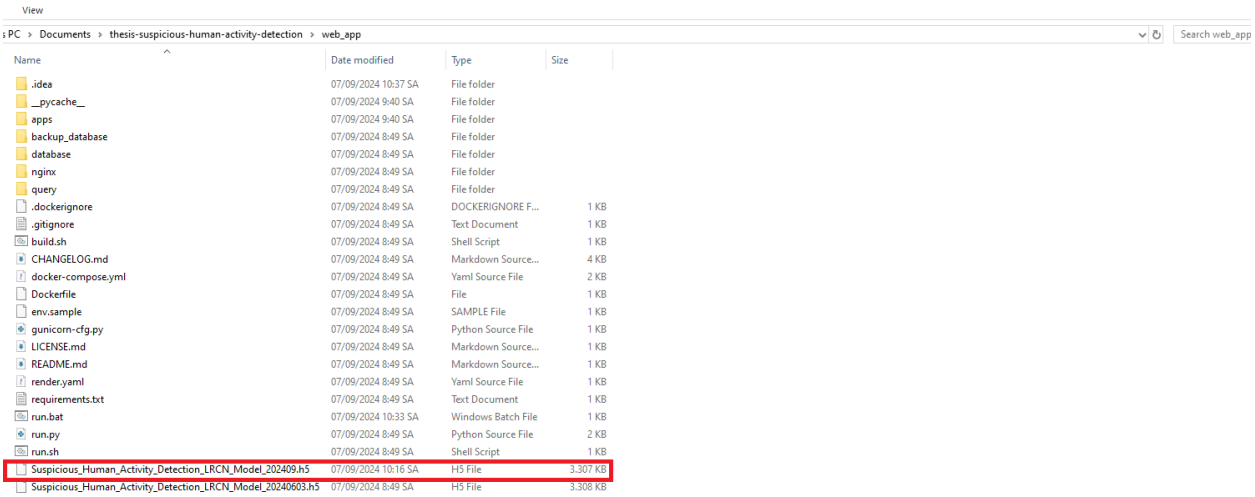
```
set FLASK_APP=.\run.py
set FLASK_ENV=development
set FLASK_DEBUG=1
flask run
```

Ứng dụng sẽ được chạy từ command line:

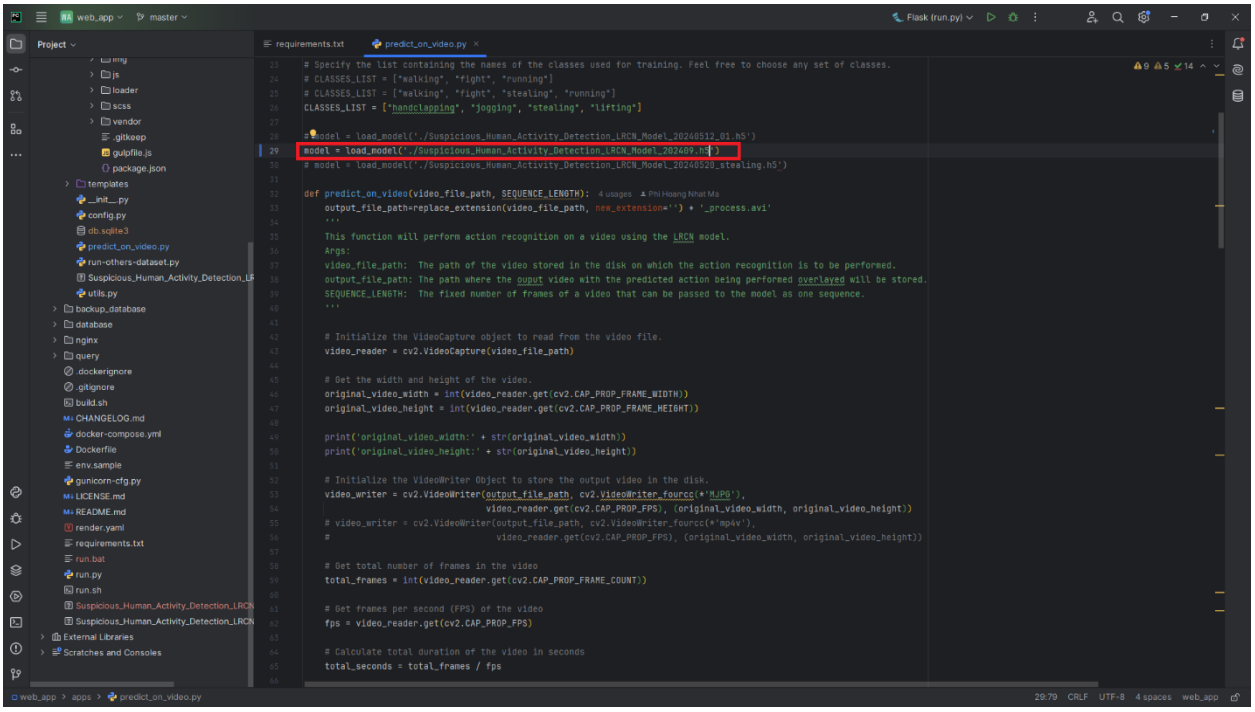


3.4. Appendix: Đổi model huấn luyện cho ứng dụng

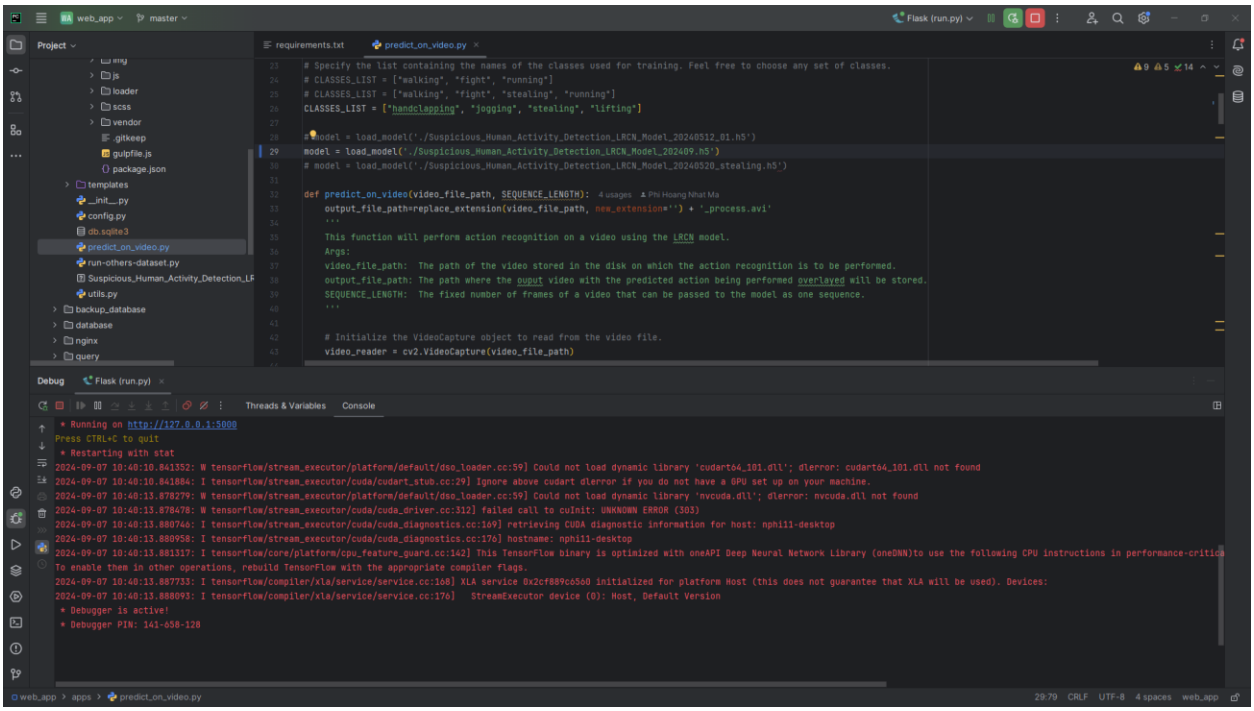
Chép mô hình đã training vào folder root của web_app như hình bên dưới.



Đổi tên mô hình mới trong file: ***web_app/apps/predict_on_video.py***



Thực hiện chạy lại ứng dụng sau khi thay đổi mô hình.


web_app
master
Flask (run.py)
Project
requirements.txt
predict_on_video.py
# Specify the list containing the names of the classes used for training. Feel free to choose any set of classes.
# CLASSES_LIST = ["walking", "fight", "running"]
# CLASSES_LIST = ["walking", "fight", "stealing", "running"]
CLASSES_LIST = ["handclapping", "jogging", "stealing", "lifting"]
# model = load_model('./Suspicious_Human_Activity_Detection_LRCN_Model_20240512_01.h5')
model = load_model('./Suspicious_Human_Activity_Detection_LRCN_Model_202409.h5')
# model = load_model('./Suspicious_Human_Activity_Detection_LRCN_Model_20240520_stealing.h5')
def predict_on_video(video_file_path, SEQUENCE_LENGTH): 4 usages Phi Hoang Nhat Ma
output_file_path=replace_extension(video_file_path, new_extension='') + '_process.avi'
'''
This function will perform action recognition on a video using the LRCN model.
Args:
video_file_path: The path of the video stored in the disk on which the action recognition is to be performed.
output_file_path: The path where the ouput video with the predicted action being performed overlayed will be stored.
SEQUENCE_LENGTH: The fixed number of frames of a video that can be passed to the model as one sequence.
'''
# Initialize the VideoCapture object to read from the video file.
video_reader = cv2.VideoCapture(video_file_path)
templates
__init__.py
config.py
db.sqlite3
predict_on_video.py
run-others-dataset.py
Suspicious_Human_Activity_Detection_LR
utils.py
backup_database
database
nginx
query
Debug
Flask (run.py)
Threads & Variables
Console
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
* Restarting with stat
2024-09-07 10:40:10.841352: W tensorflow/stream_executor/platform/default/dso_loader.cc:59] Could not load dynamic library 'cudart64_101.dll'; dlerror: cudart64_101.dll not found
2024-09-07 10:40:10.841884: I tensorflow/stream_executor/cuda/cudart_stub.cc:29] Ignore above cudart dlerror if you do not have a GPU set up on your machine.
2024-09-07 10:40:13.878279: W tensorflow/stream_executor/platform/default/dso_loader.cc:59] Could not load dynamic library 'nvcuda.dll'; dlerror: nvcuda.dll not found
2024-09-07 10:40:13.878478: W tensorflow/stream_executor/cuda/cuda_driver.cc:312] failed call to cuInit: UNKNOWN ERROR (303)
2024-09-07 10:40:13.880746: I tensorflow/stream_executor/cuda/cuda_diagnostics.cc:169] retrieving CUDA diagnostic information for host: nphi11-desktop
2024-09-07 10:40:13.880958: I tensorflow/stream_executor/cuda/cuda_diagnostics.cc:176] hostname: nphi11-desktop
2024-09-07 10:40:13.881317: I tensorflow/core/platform/cpu_feature_guard.cc:142] This TensorFlow binary is optimized with oneAPI Deep Neural Network Library (oneDNN)to use the following CPU instructions in performance-critica
To enable them in other operations, rebuild TensorFlow with the appropriate compiler flags.
2024-09-07 10:40:13.887733: I tensorflow/compiler/xla/service/service.cc:168] XLA service 0x2cf889c6560 initialized for platform Host (this does not guarantee that XLA will be used). Devices:
2024-09-07 10:40:13.888093: I tensorflow/compiler/xla/service/service.cc:176] StreamExecutor device (0): Host, Default Version
* Debugger is active!
* Debugger PIN: 141-658-128
web_app > apps > predict_on_video.py
29:79 CRLF UTF-8 4 spaces web_app