**Chương 02: Dividing Text Data and   
Building Text Classifiers**

**Môn: Lập trình hệ thống – Giảng viên: Nguyễn Văn Vịnh**

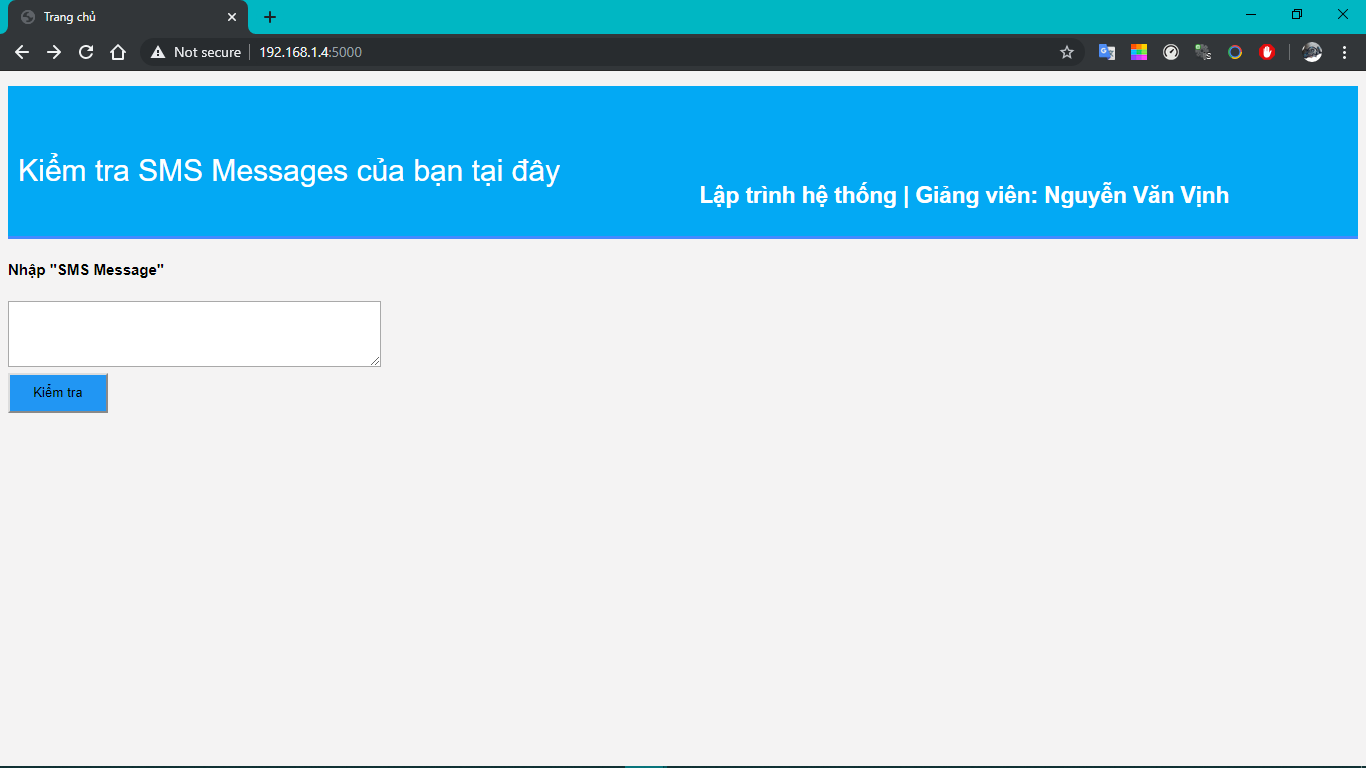
**Nhóm 02:**

1. **Nguyễn Thành Trung**
2. **Lê Quang Hướng**
3. **Nguyễn Mạnh Tiến**

**Báo cáo cuối kì môn Lập trình hệ thống**

**“Hệ thống nhận diện SMS Spam”**

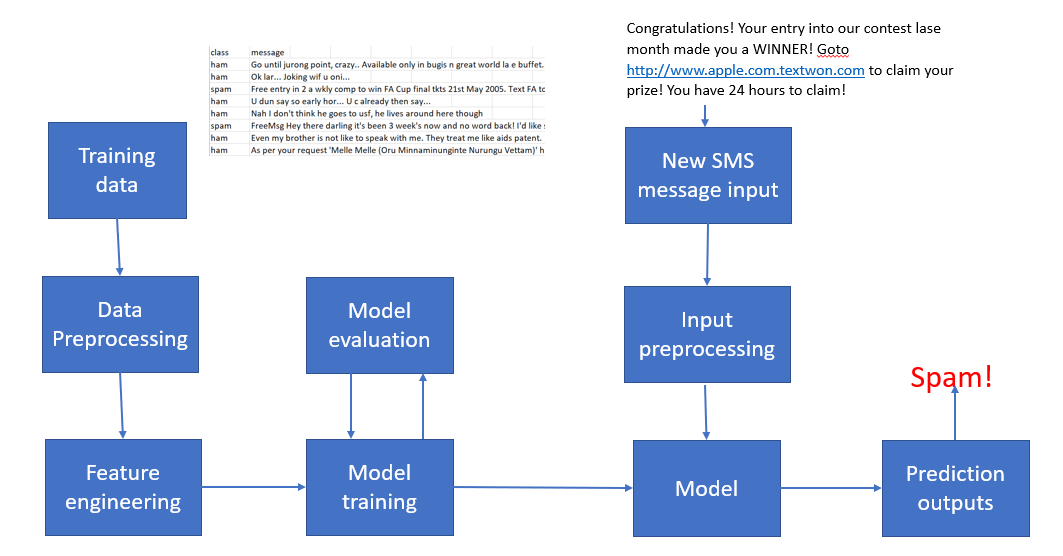
**Triển khai hệ thống nhận diện SMS Spam bằng mô hình máy học (ML - Machine Learning) trên Web App Flask chạy bằng   
“uWSGI làm http server”, “Nginx làm reverse proxy server” trên phần cứng “Raspberry Pi 4”.**



1. **Xây dựng model nhận diện SMS Spam dùng ML trên Flask App.**
2. **Xây dựng Server, Flask App trên Raspberry Pi 4.**

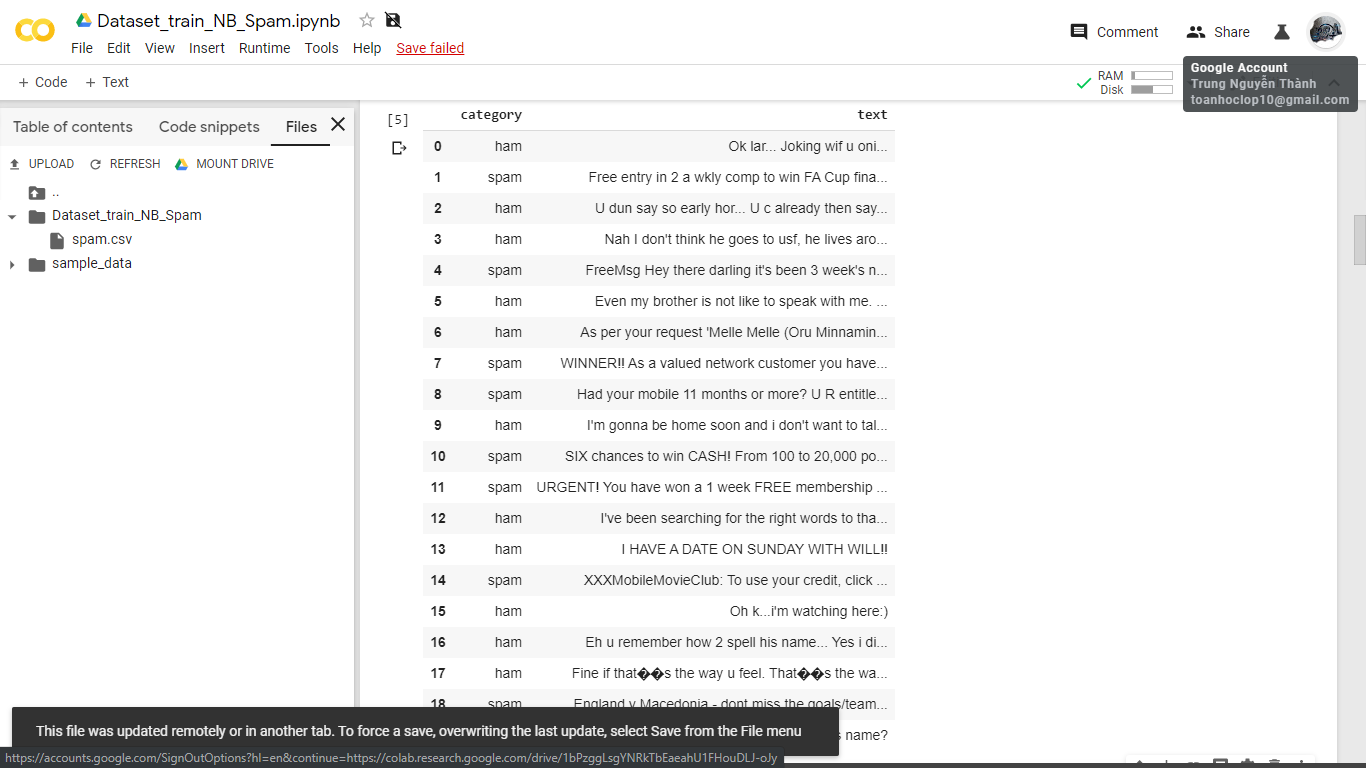
**Video Demo:**

1. **Xây dựng hệ thống nhận diện SMS Spam dùng ML trên Flask App.**

****

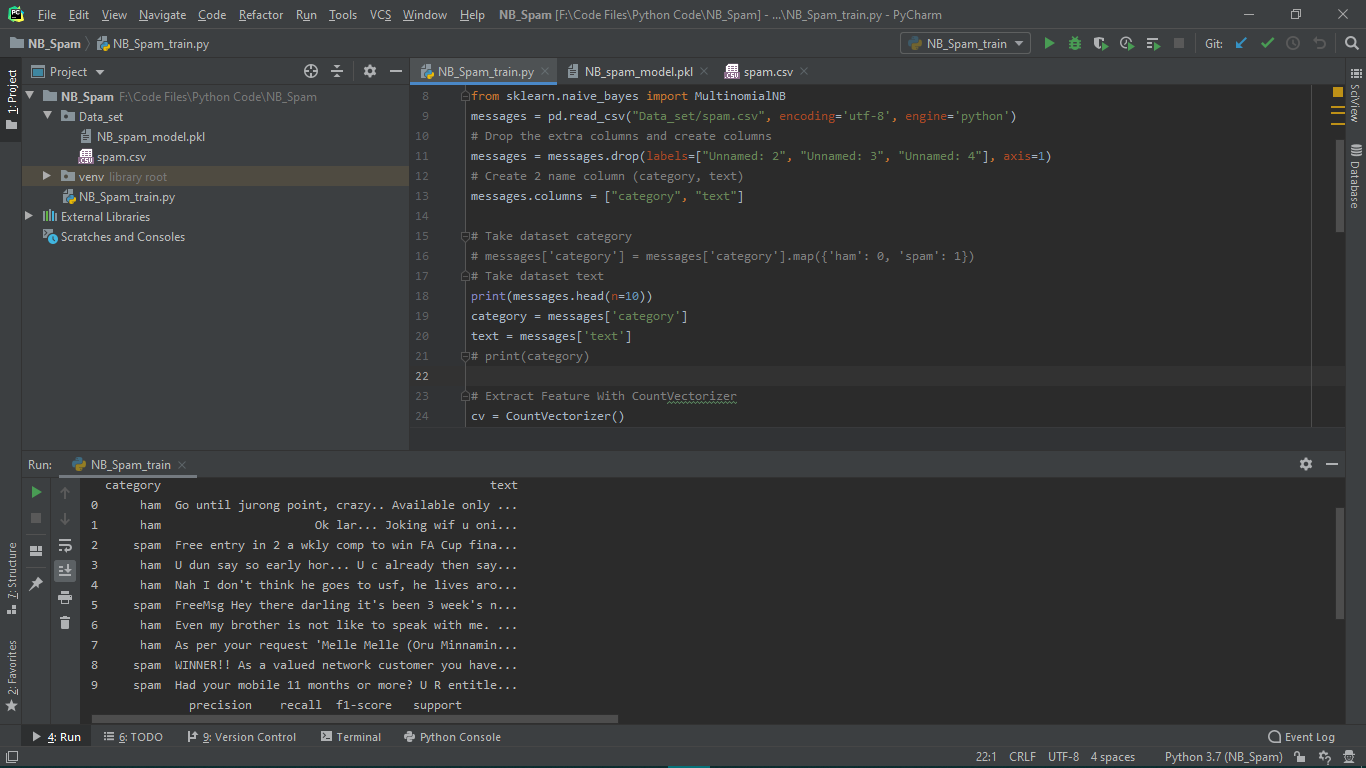
* 1. **Xây dựng model phân loại SMS Spam và không Spam:**
* **Chuẩn bị dữ liệu:**

**Nhóm dùng dataset gồm 5571 tin nhắn SMS được gắn thẻ Spam và Ham (ko phải Spam).**



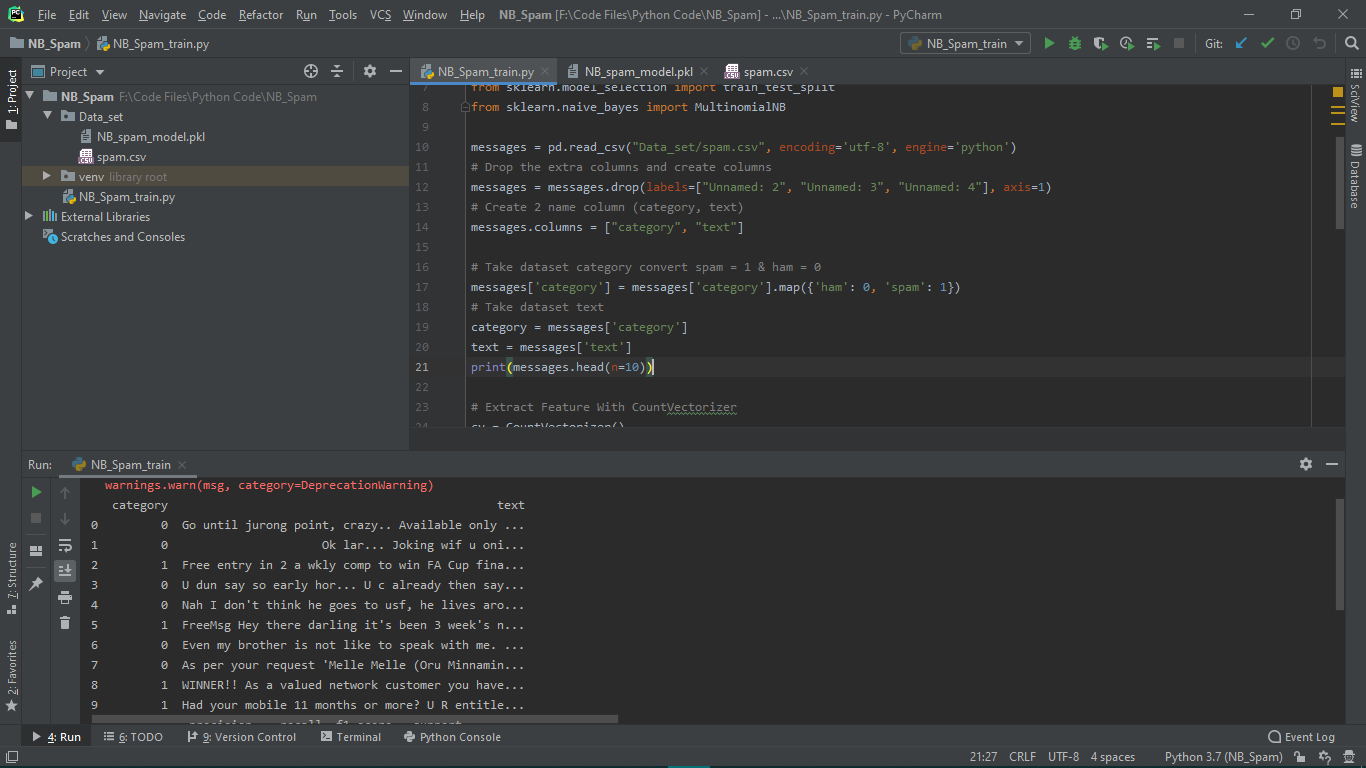
* 1. **Train dữ liệu bằng Naive Bayes Classifier. Xây dựng model và lấy dữ liệu từ model.**
* **Load dataset .csv để sử dụng và đặt tên cột:**

messages = pd.read\_csv("Data\_set/spam.csv", encoding='utf-8', engine='python')  
# Drop the extra columns and create columns  
messages = messages.drop(labels=["Unnamed: 2", "Unnamed: 3", "Unnamed: 4"], axis=1)  
# Create 2 name column (category, text)  
messages.columns = ["category", "text"]



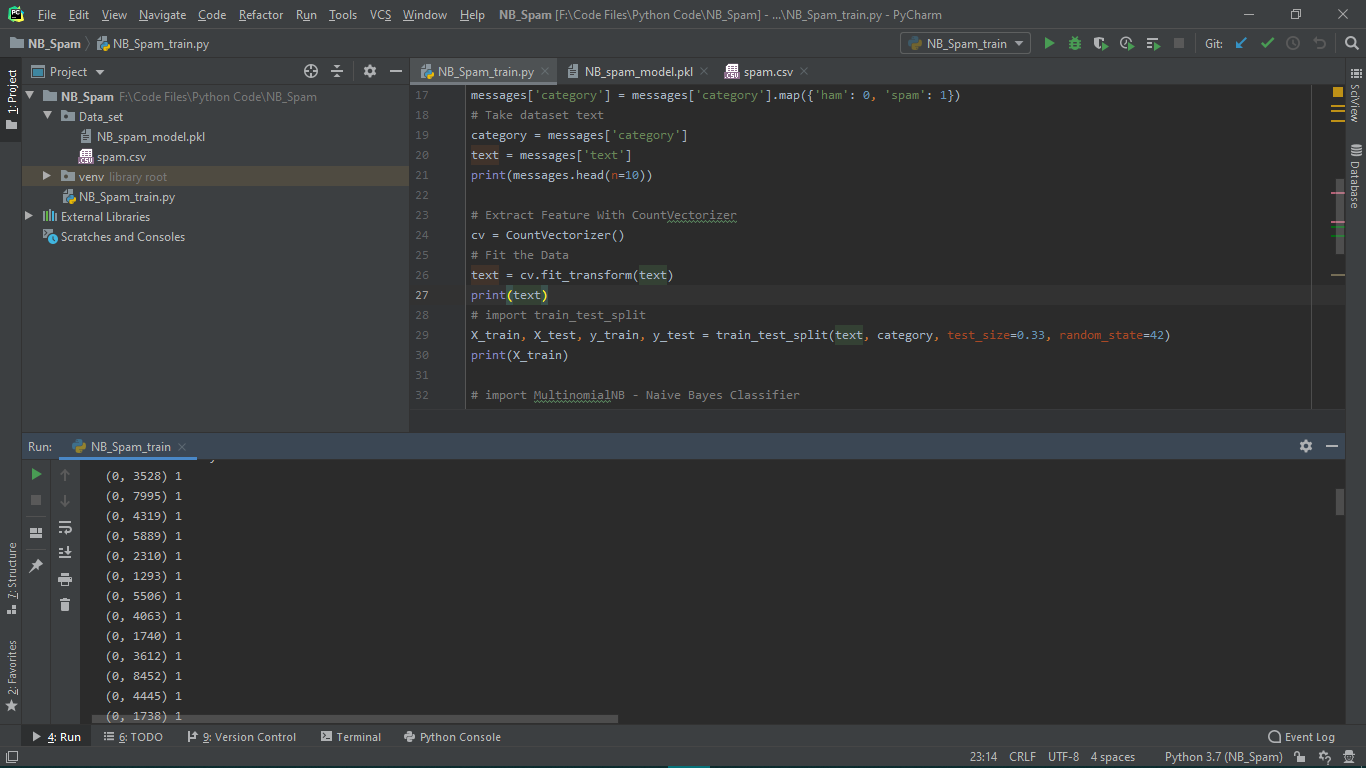
* **Giải thuật Machine Learning (Naive Bayes Classifier) chỉ làm việc được với số nên nhóm chuyển “Spam”, “Ham” sang dạng số tương ứng “Spam : 1”, “Ham : 0”.**

# Take dataset category convert spam = 1 & ham = 0  
messages['category'] = messages['category'].map({'ham': 0, 'spam': 1})  
# Take dataset text  
category = messages['category']  
text = messages['text']  
print(messages.head(n=10))



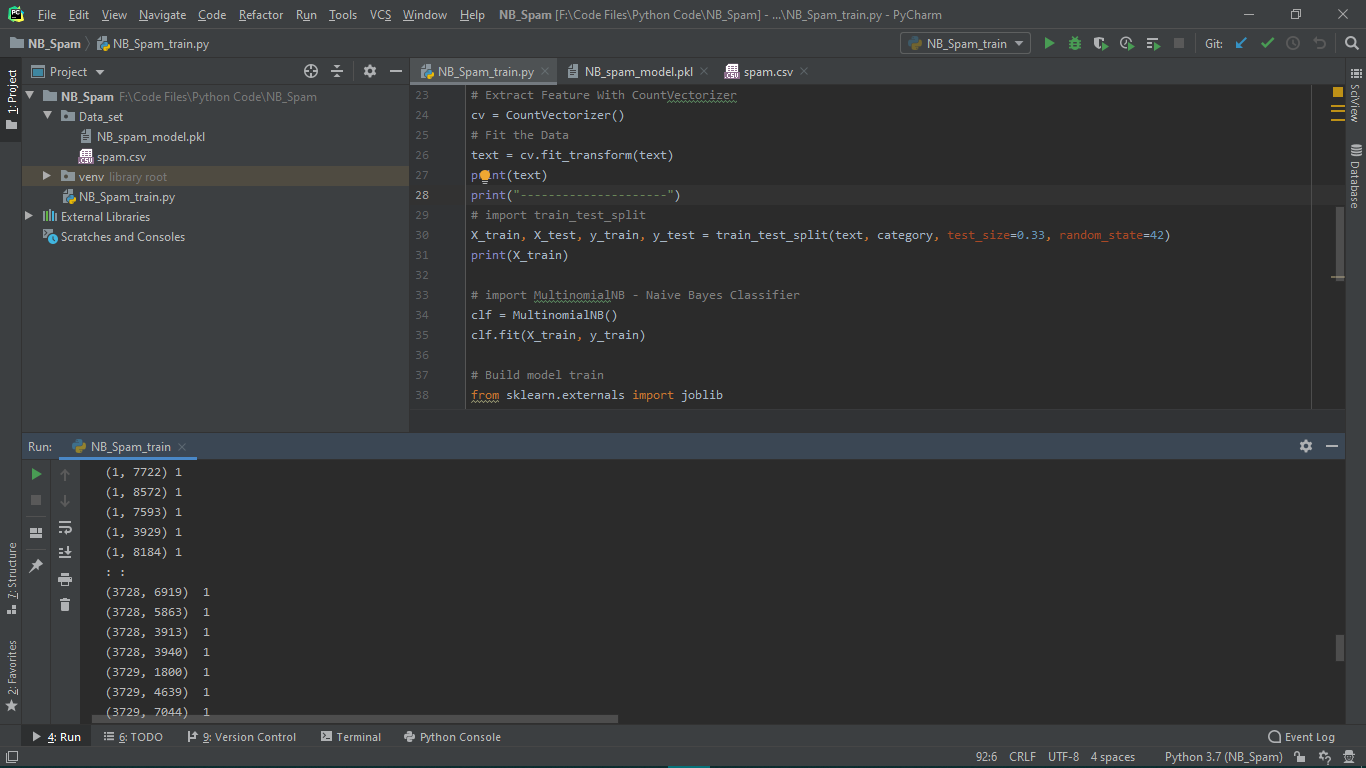
* **Mô hình hóa văn bản thành vector:**

# Extract Feature With CountVectorizer  
cv = CountVectorizer()  
# Fit the Data  
text = cv.fit\_transform(text)



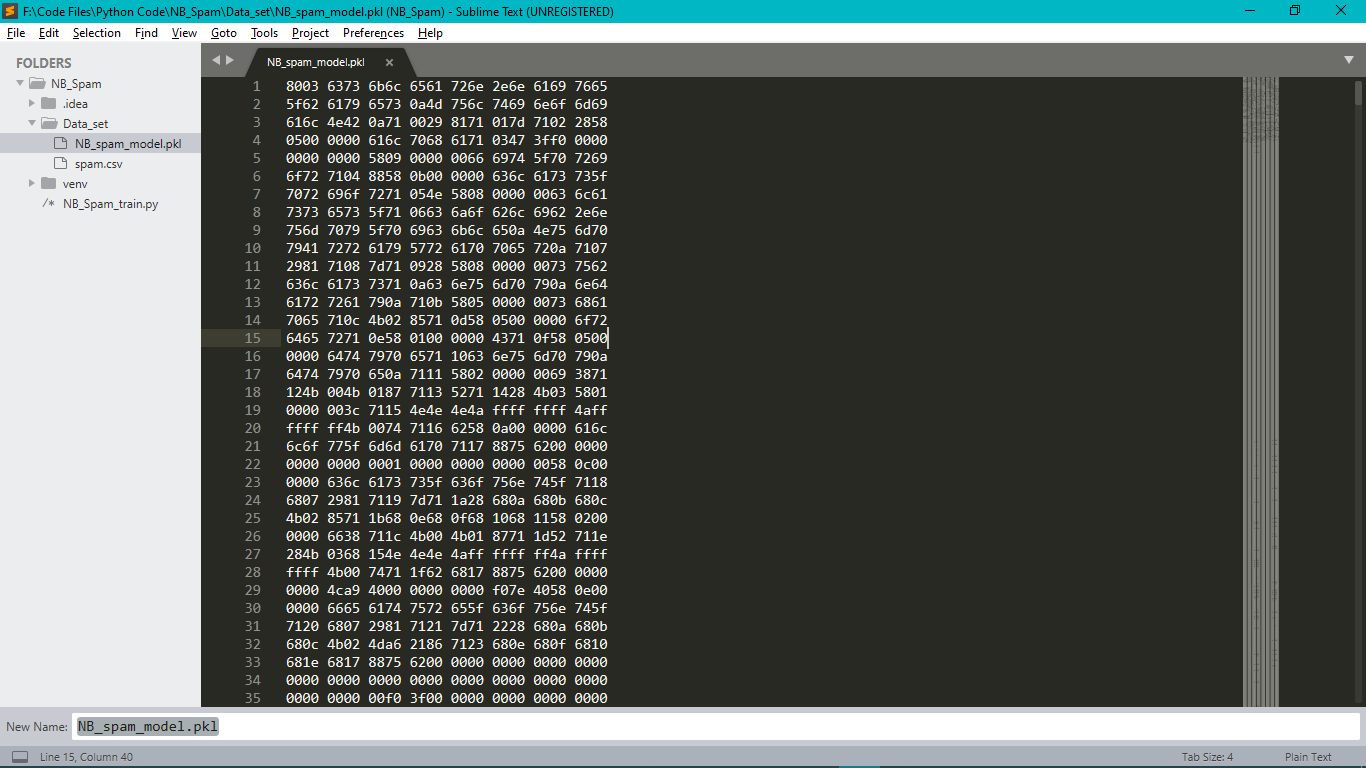
* **Xây dựng ma trận vector(train & test):**

# import train\_test\_split  
X\_train, X\_test, y\_train, y\_test = train\_test\_split(text, category, test\_size=0.33, random\_state=42)



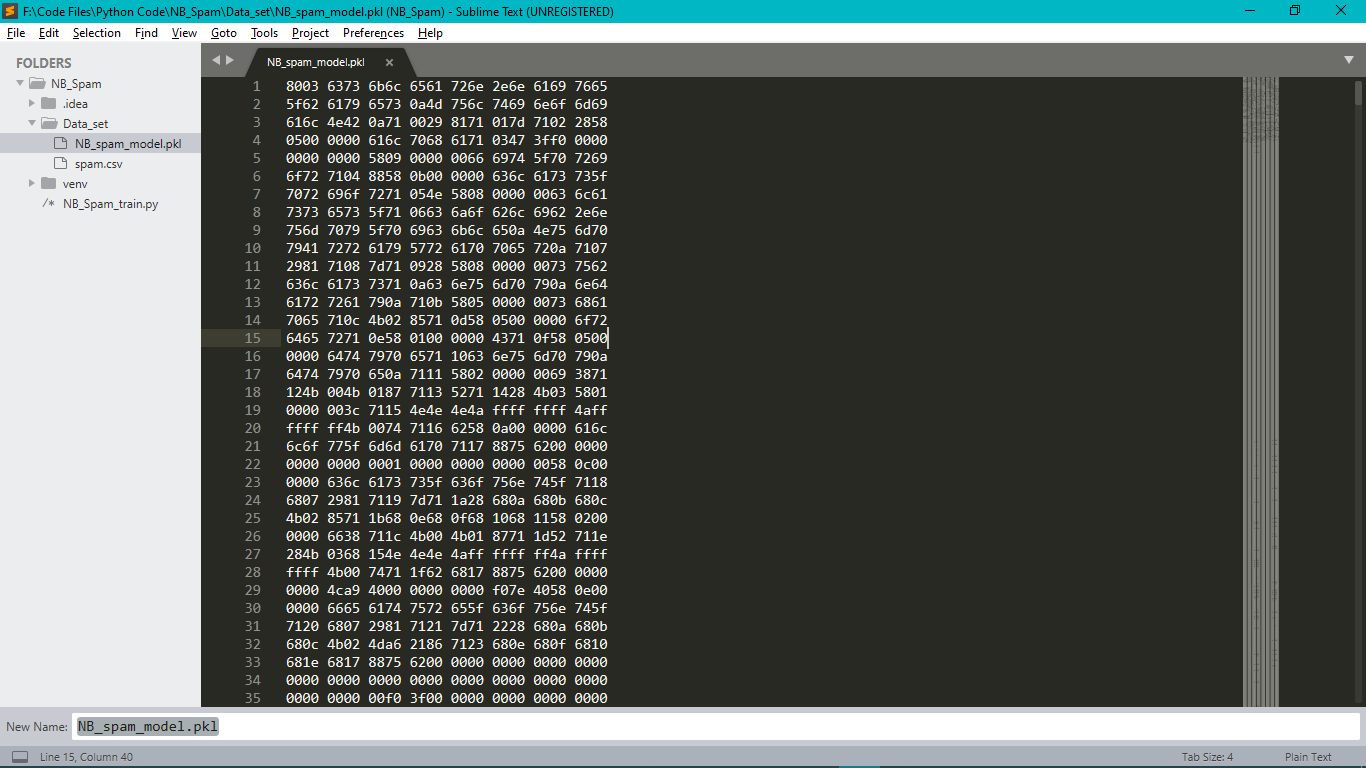
* **Train dữ liệu bằng thuật toán Naive Bayes Classifier:**

# import MultinomialNB - Naive Bayes Classifier  
clf = MultinomialNB()  
clf.fit(X\_train, y\_train)



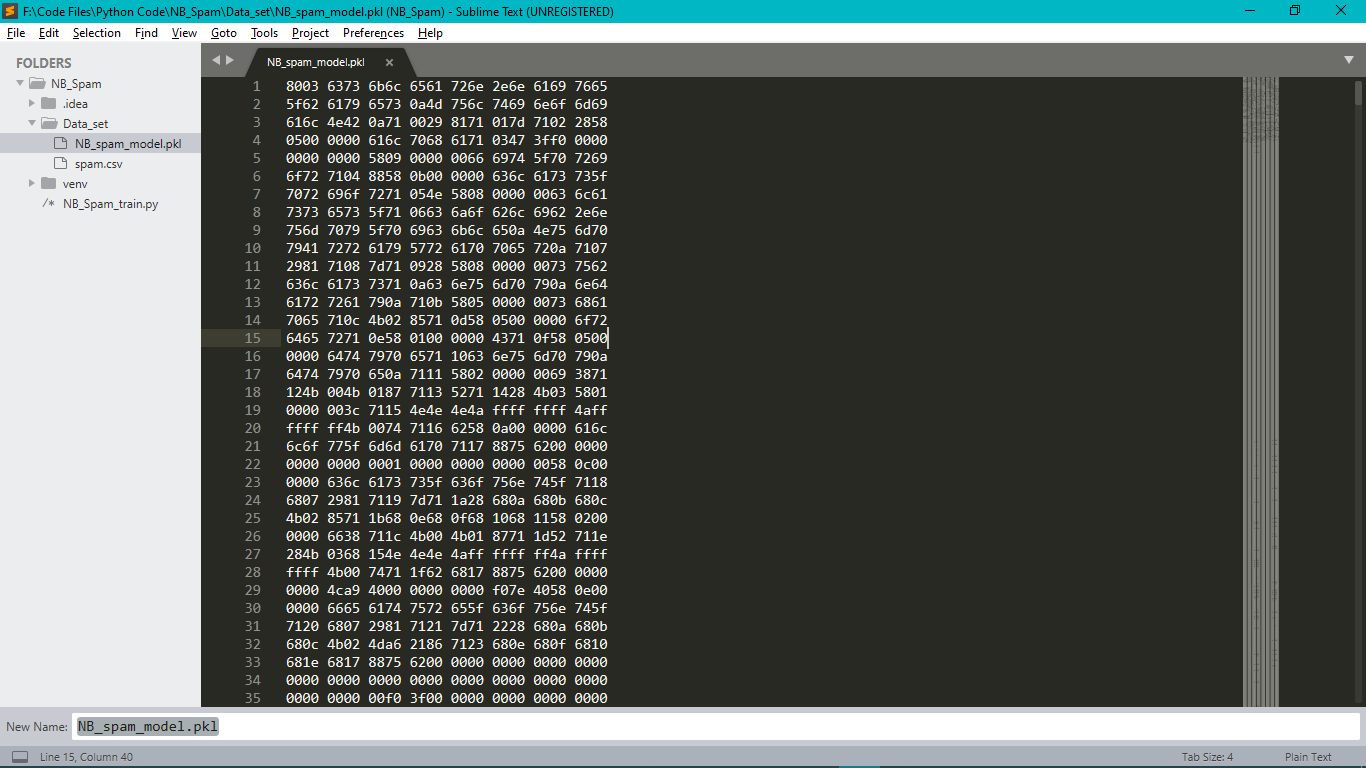
* **Xây dựng model train (file .pkl).**

# Build model train  
from sklearn.externals import joblib  
joblib.dump(clf, 'Data\_set/NB\_spam\_model.pkl')



* **Tải dữ liệu model train (file .pkl):**

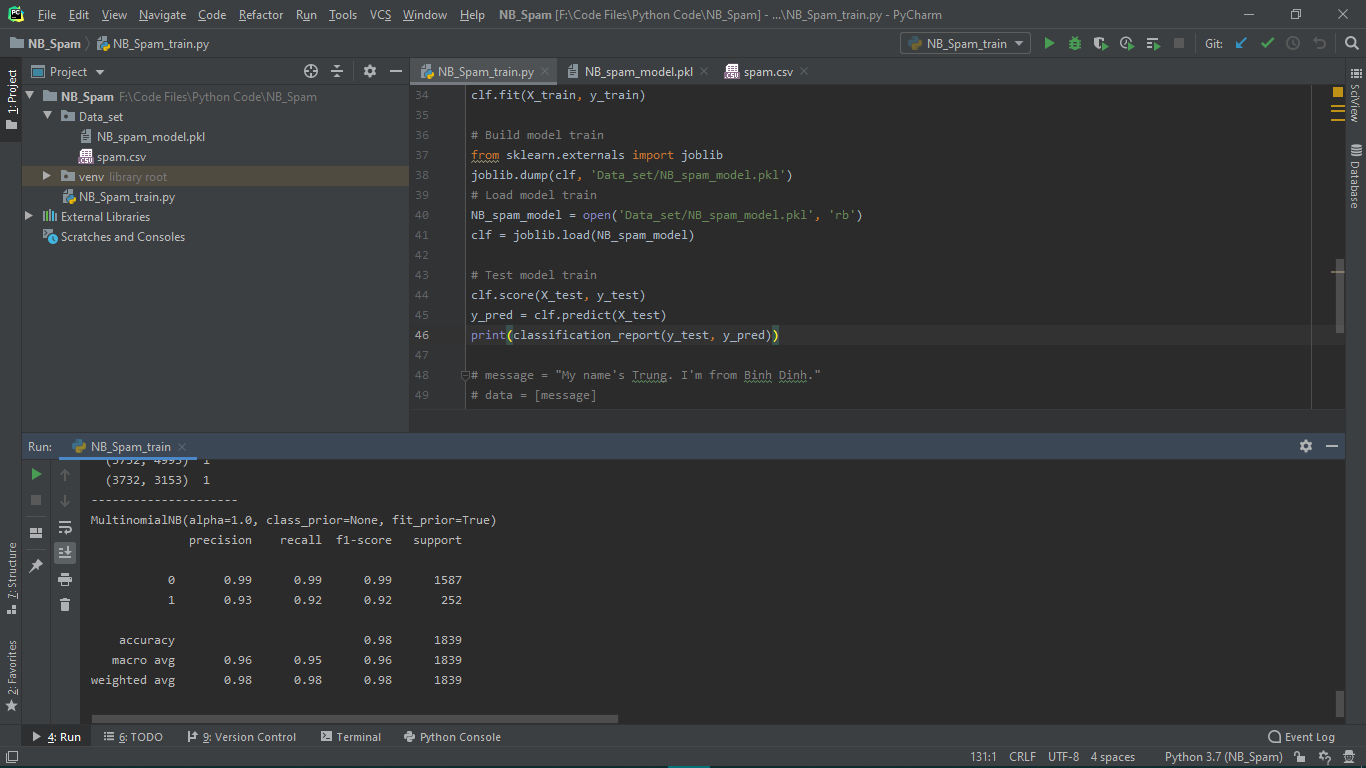
# Load model train  
NB\_spam\_model = open('Data\_set/NB\_spam\_model.pkl', 'rb')  
clf = joblib.load(NB\_spam\_model)



* **Test model train:**

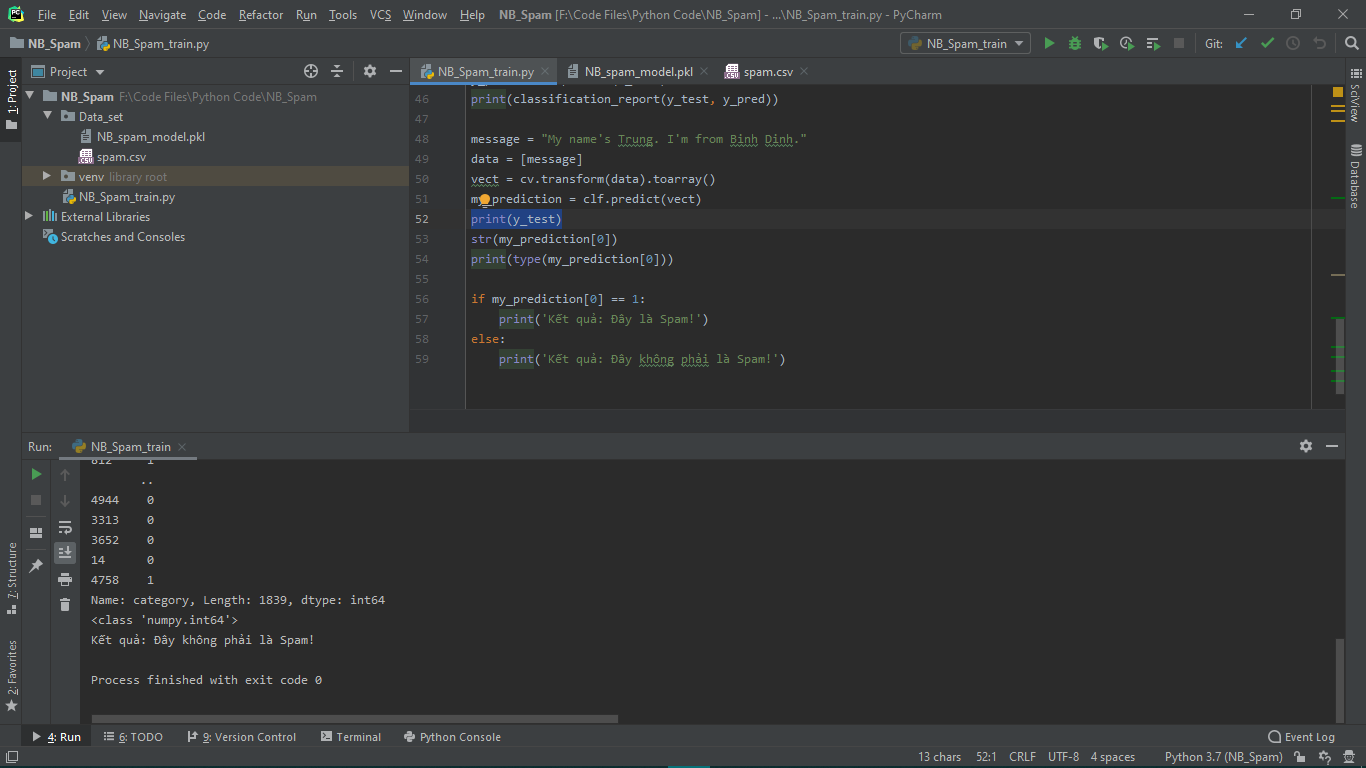
# Test model train

clf.score(X\_test, y\_test)  
y\_pred = clf.predict(X\_test)  
print(classification\_report(y\_test, y\_pred))

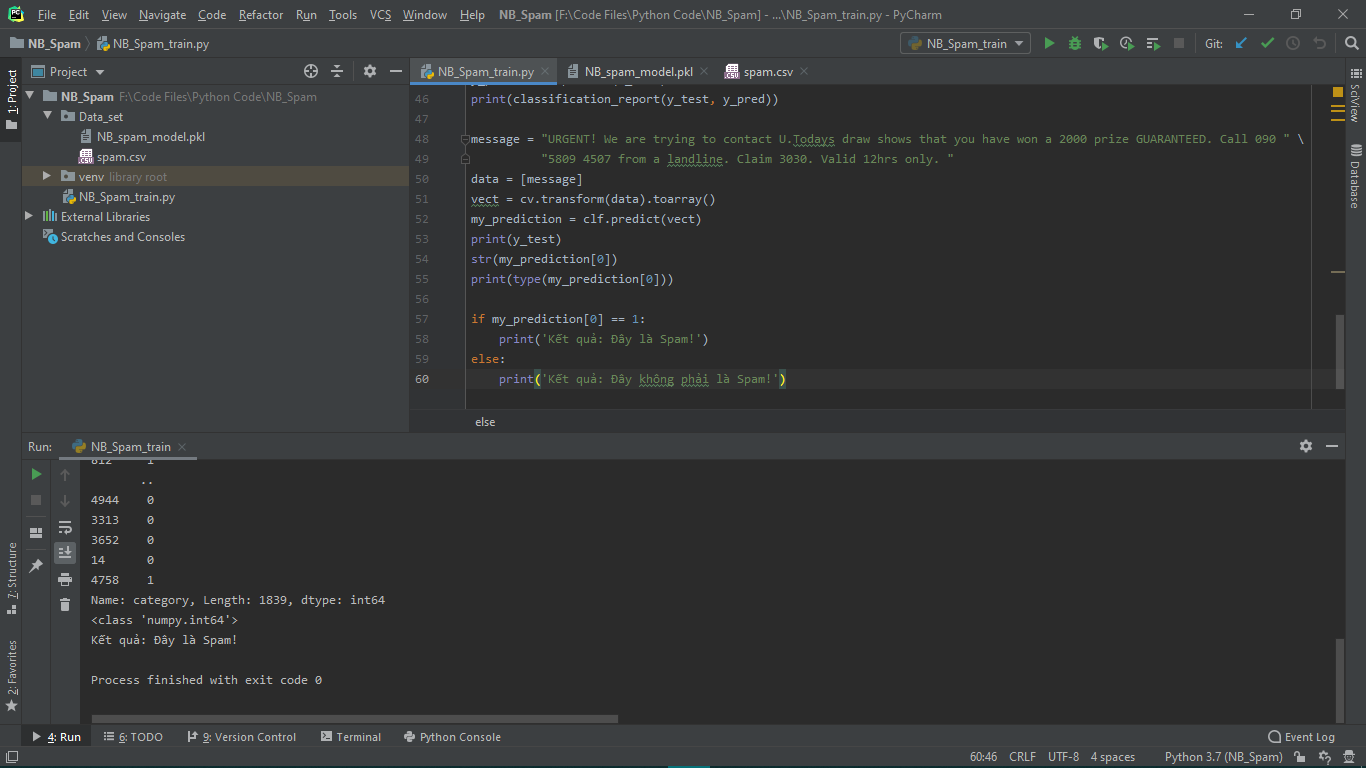


* **Độ chính xác: 0.98 (98%)**
* **Dữ liệu test gồm 1839 SMS [1587 (ham) và 252 (spam)].**
* **Tổng văn bản dữ liệu: 5571(spam, ham).**
* **Kiểm thử hệ thống nhận diện SMS Spam:**

**Not Spam:** My name's Trung. I'm from Binh Dinh.



**Spam:** URGENT! We are trying to contact U.Todays draw shows that you have won a 2000 prize GUARANTEED. Call 090 5809 4507 from a landline. Claim 3030. Valid 12hrs only.



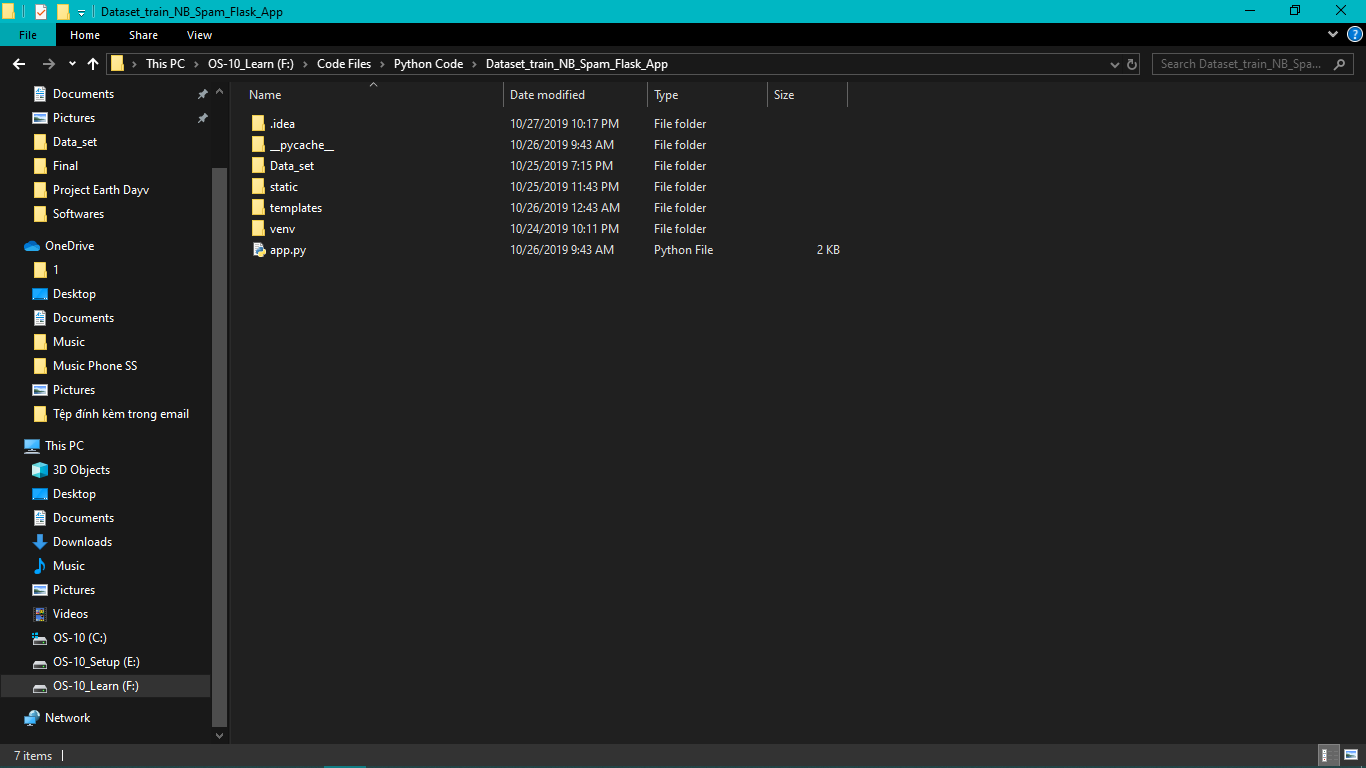
* 1. **Triển khai hệ thống trên ứng dụng web Flask:**

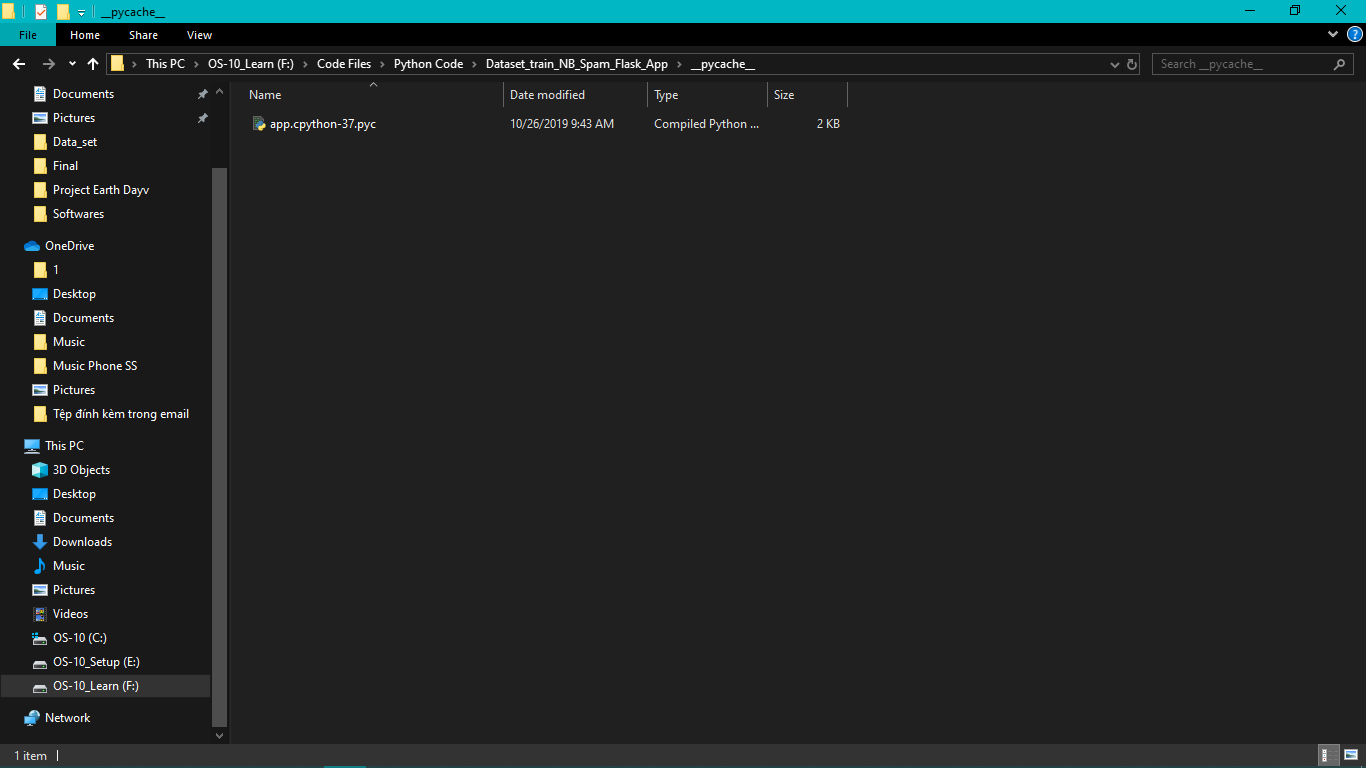
Xây dựng Flask app đơn giản thực hiện nhận diện SMS Spam.

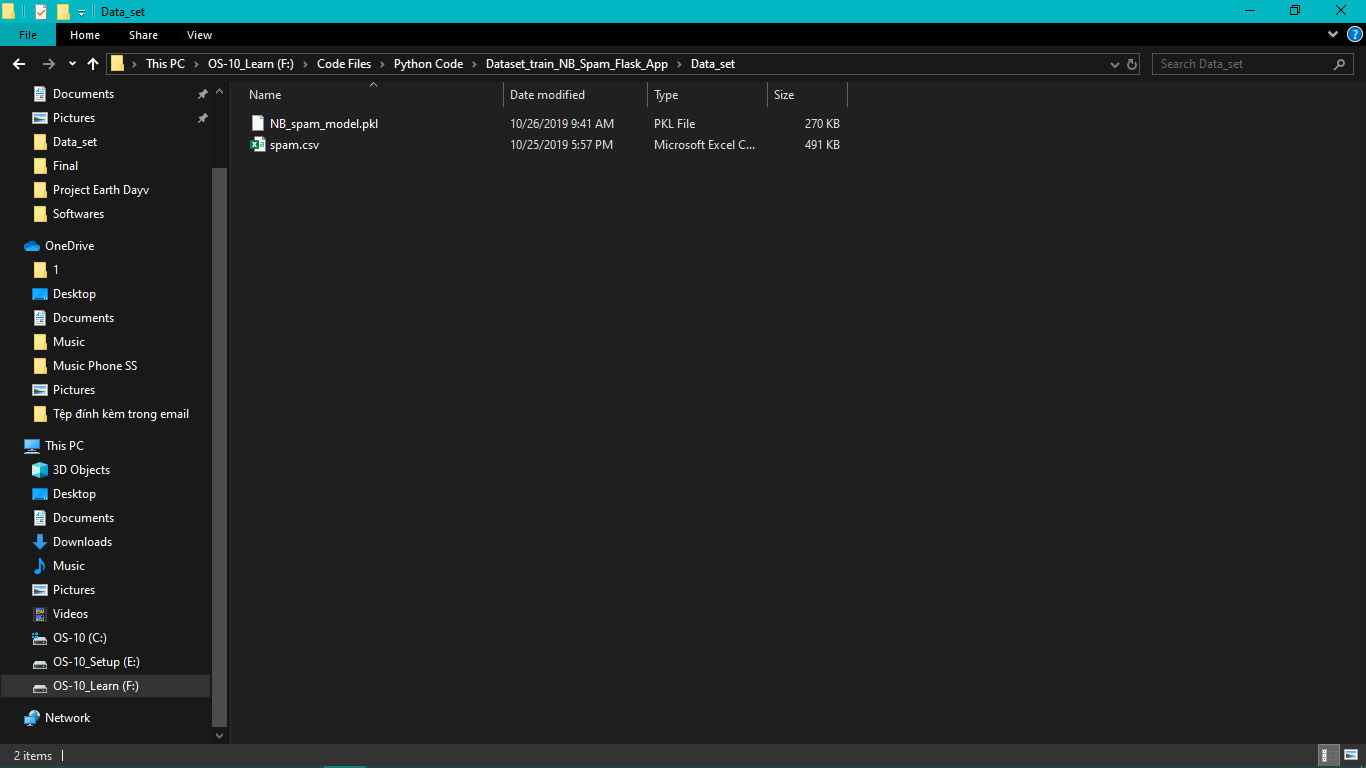
Đầu tiên: Tạo folder “Dataset\_train\_NB\_Spam\_Flask\_App”.

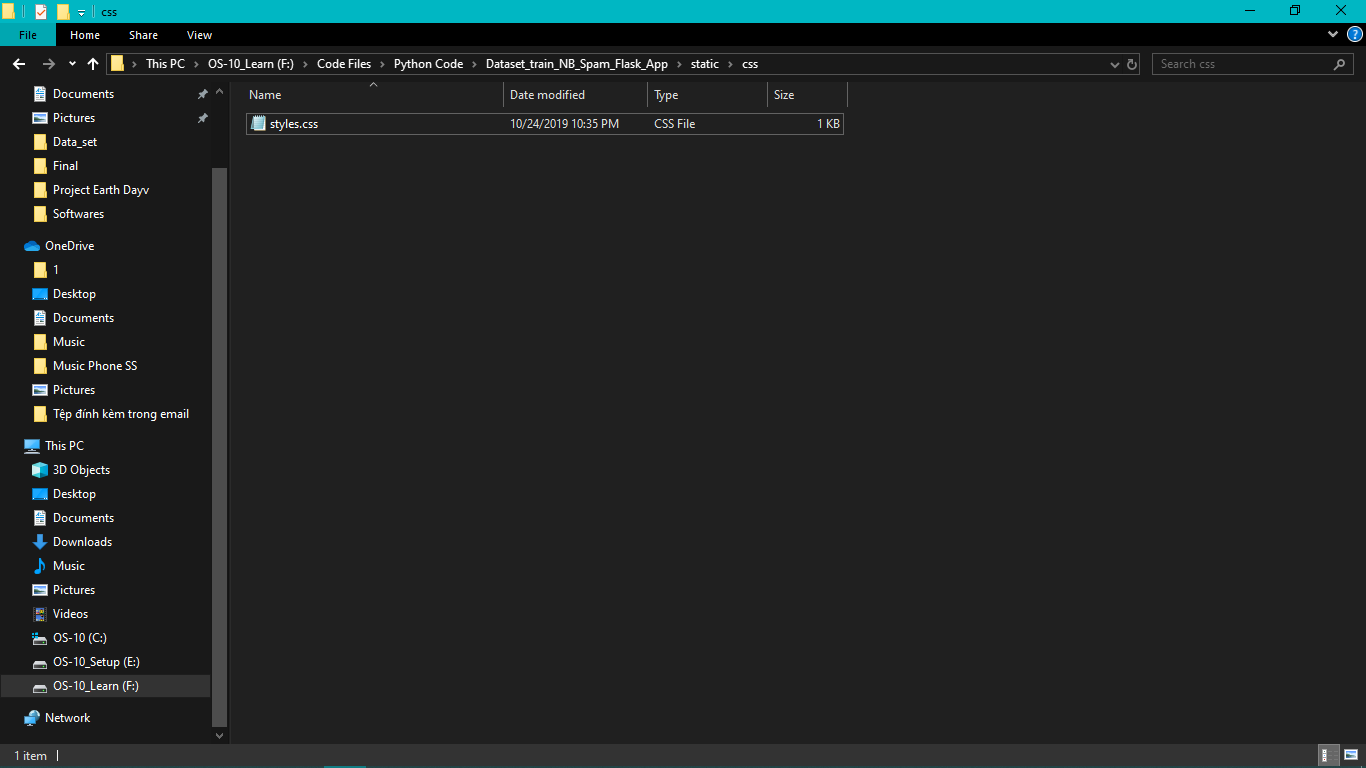
Cấu trúc file trong folder:

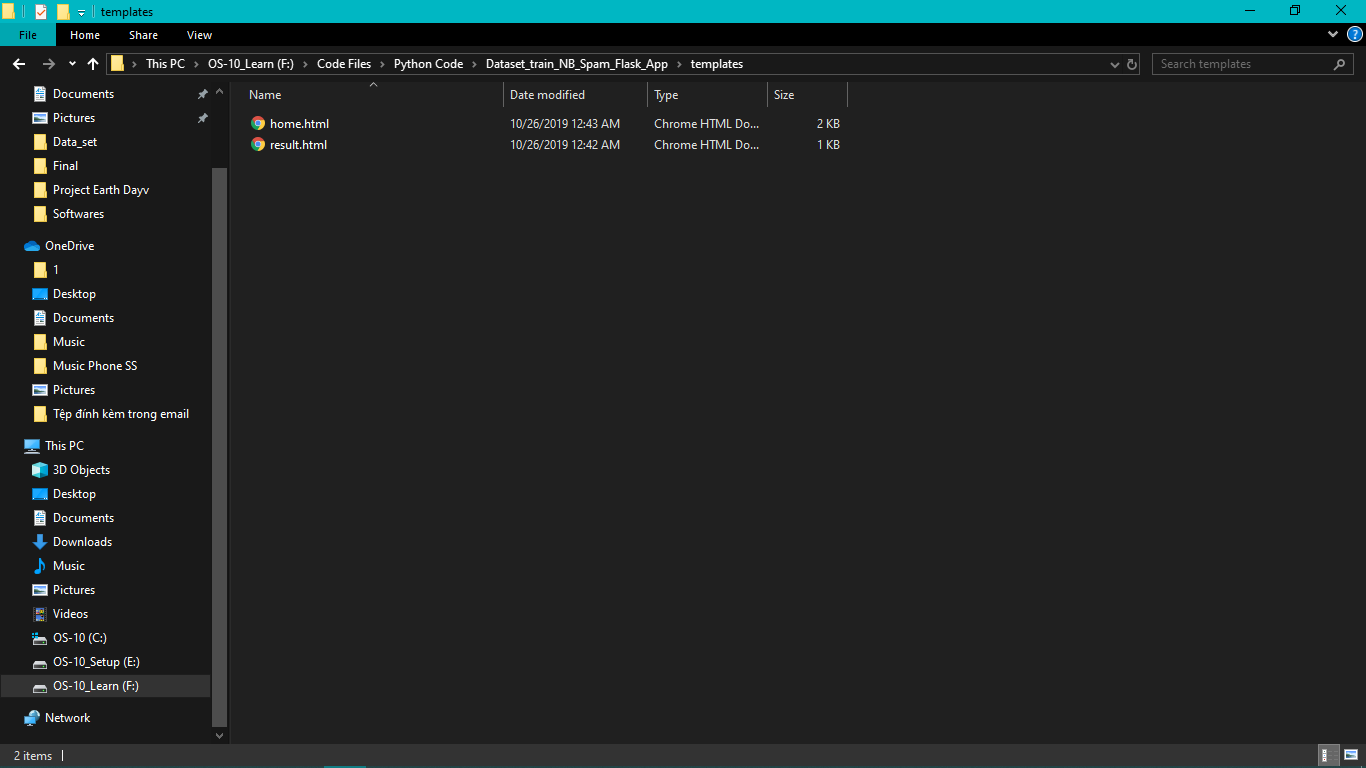
\_\_pycache\_\_/  
 app.cpython-37.pyc  
Dataset/  
 MB\_spam\_model.pkl  
 spam.csv  
templates/  
 home.html  
 result.html  
static/  
 css/  
 style.css  
app.py



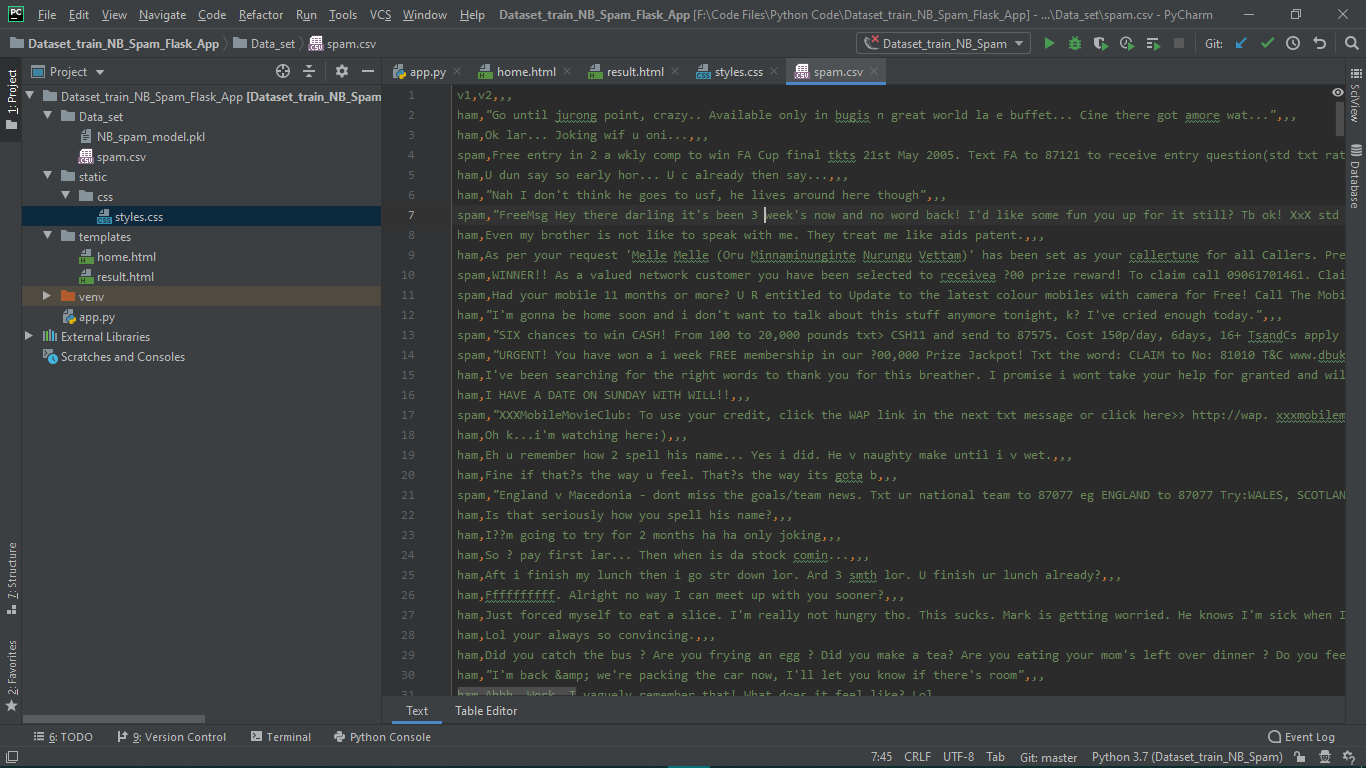




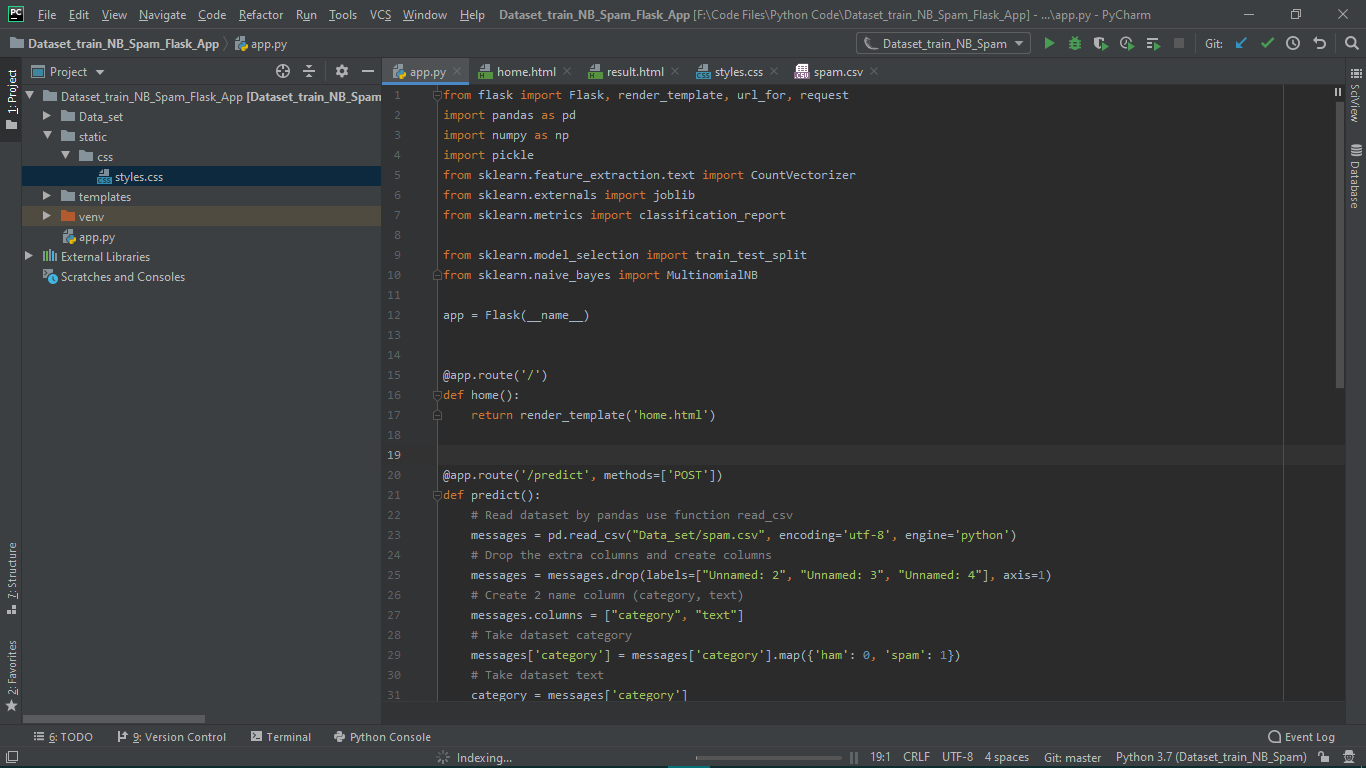




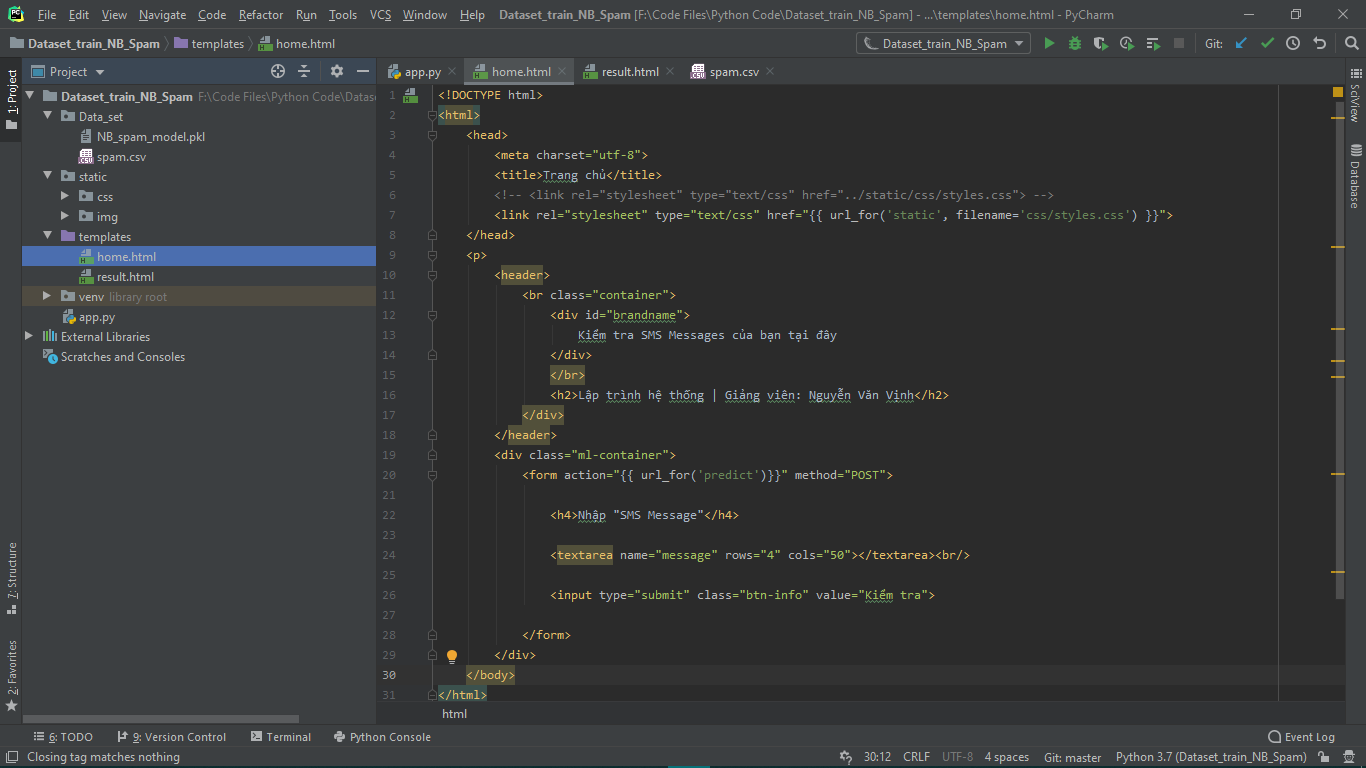
**spam.csv**



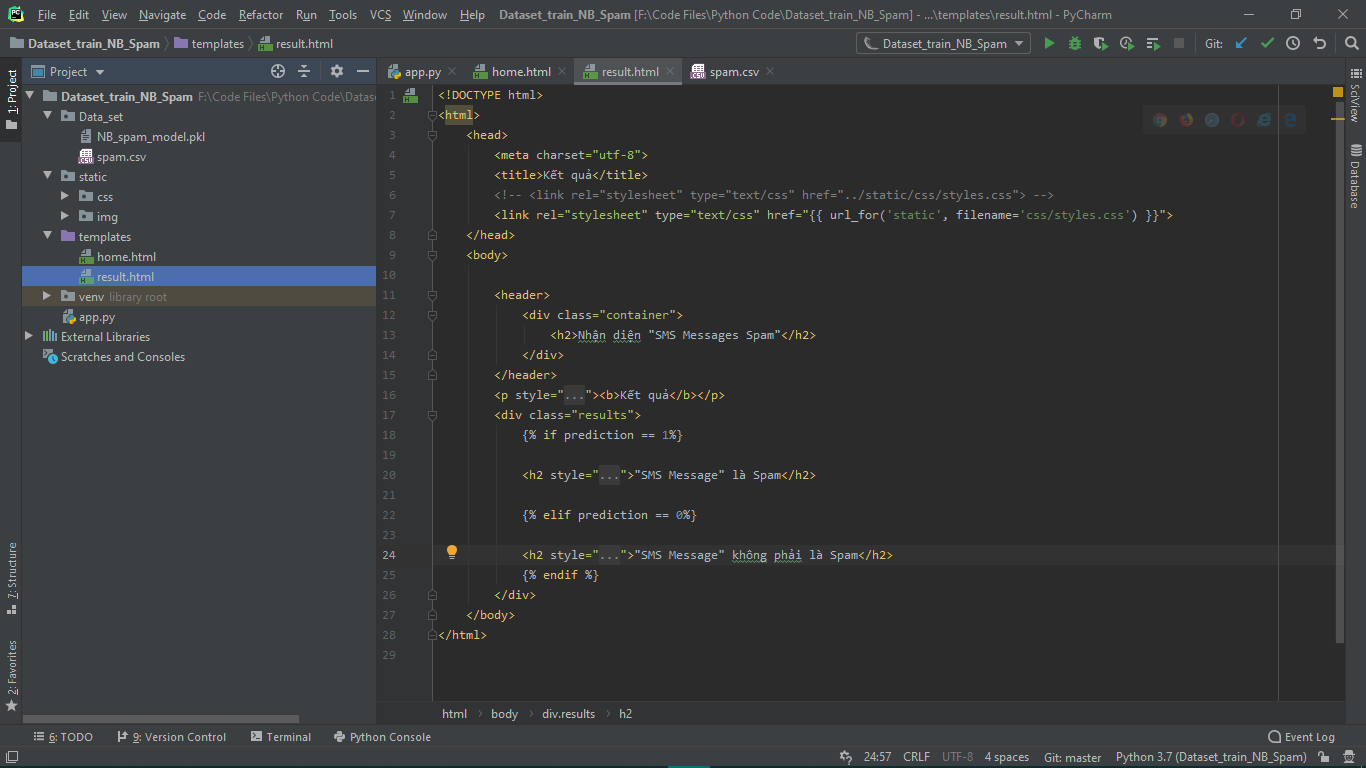
**app.py**



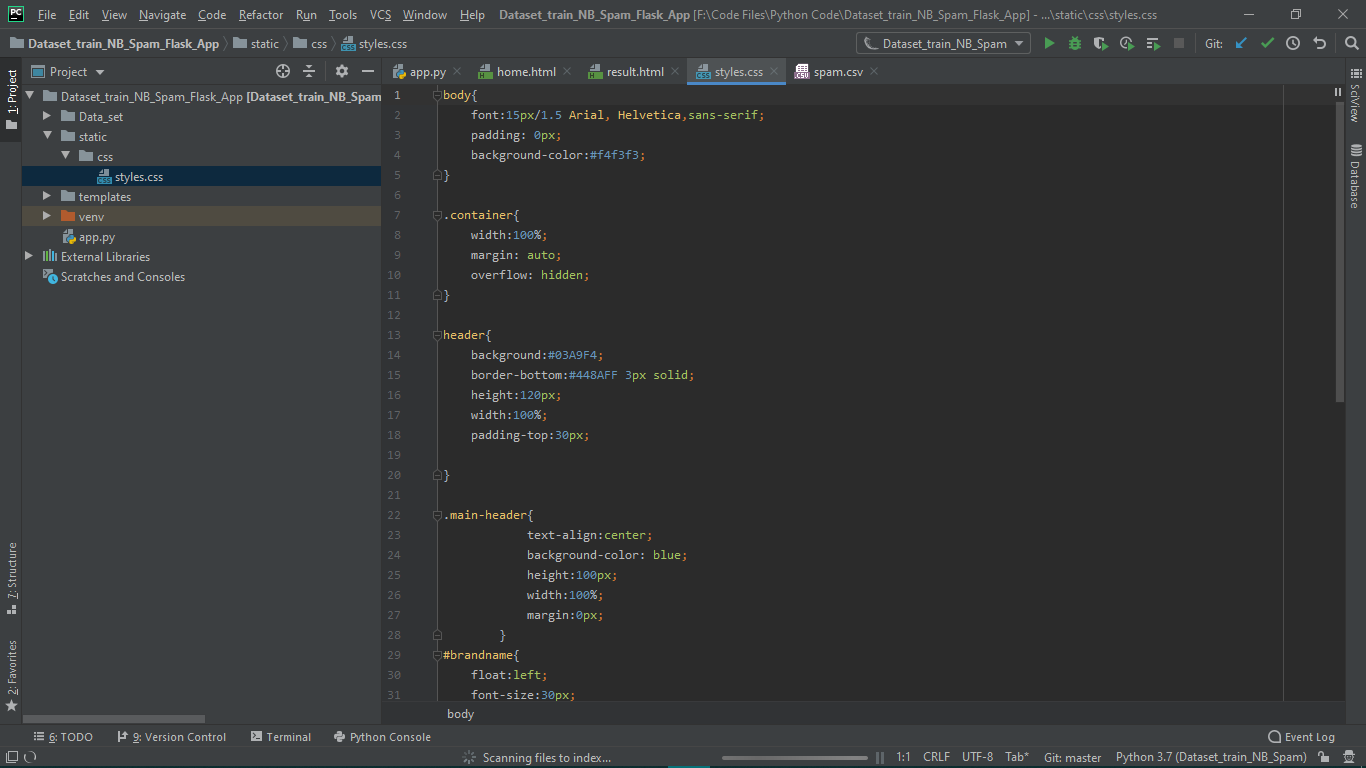
**home.html**



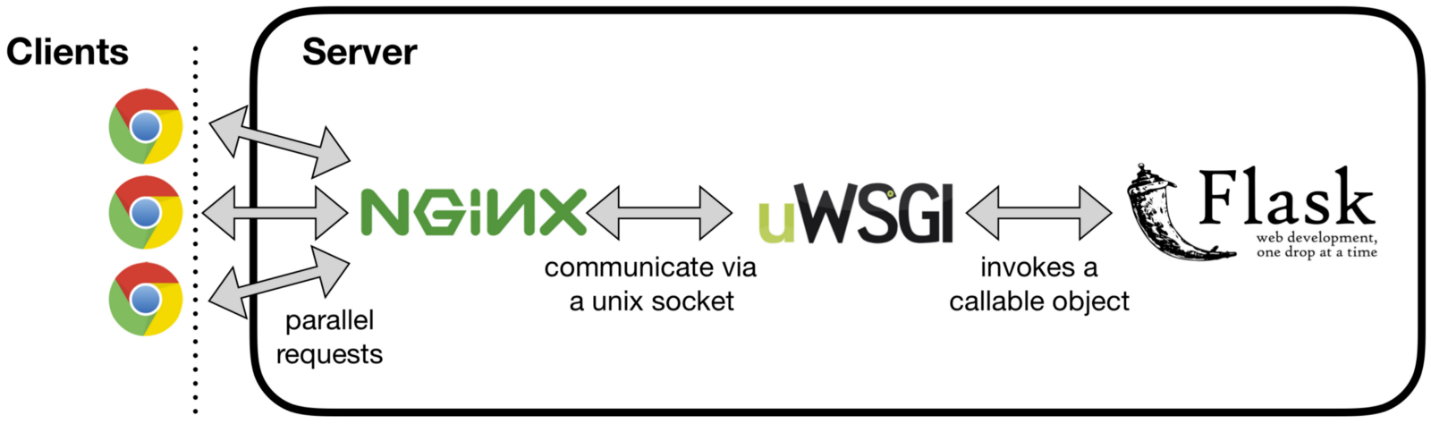
**result.html**



**styles.css**



1. **Xây dựng Server, Flask App trên Raspberry Pi 4.**

****

* **Flask:** ứng dụng web app giao diện.
* **uWSGI:** nhận và xử lý các file Python.
* **Nginx:** Làm front-end để hứng các request.
* **Clients:** Trình duyệt truy cập vào domain(hoặc IP).
  1. **Tạo môi trường ảo Python:**

$ sudo apt install python3-venv

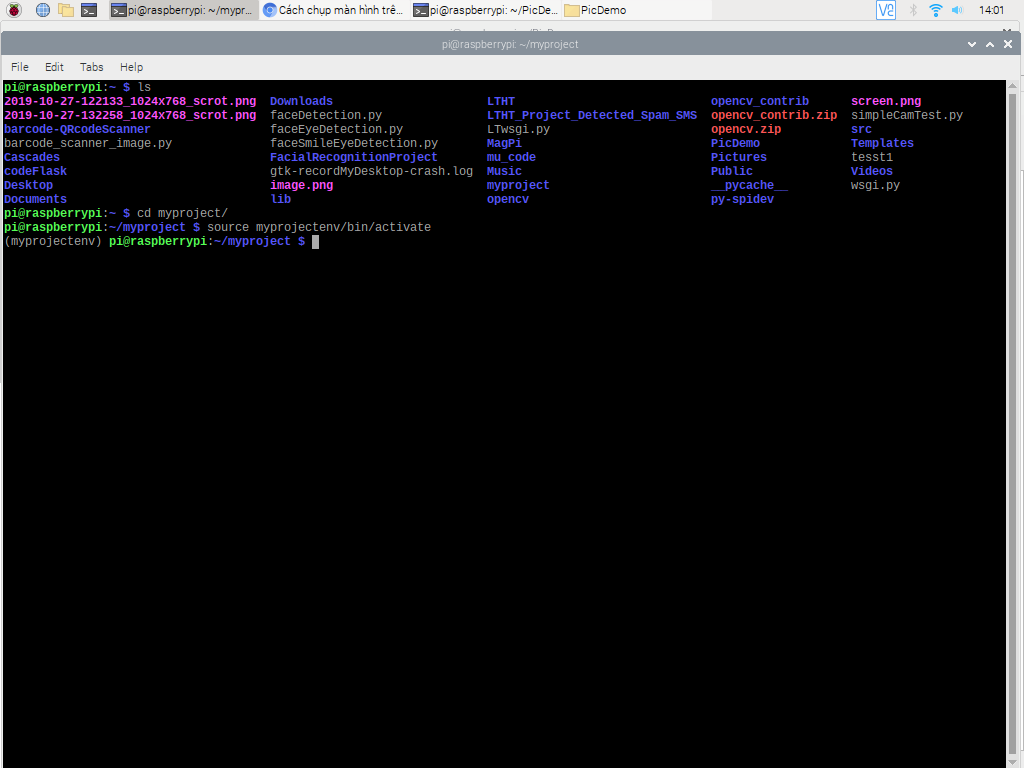
$ mkdir ~/myproject

$ cd ~/myproject

$ python3.6 -m venv myprojectenv

$ source myprojectenv/bin/activate

**Demo:**

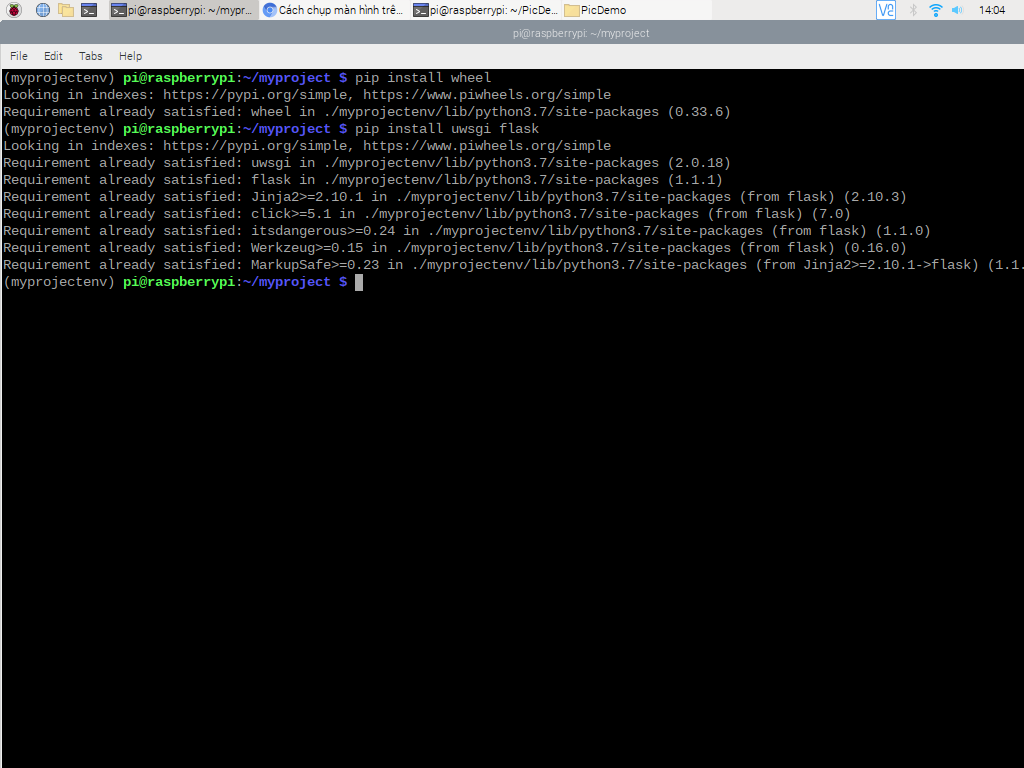
****

* 1. **Cài đặt Flask App:**

$ pip install wheel

(myprojectenv) $ pip install uwsgi flask

**Demo:**



* 1. **Cấu hình uWSGI Server:**

**Khởi tạo uWSGI:**

$ pip install uWSGI

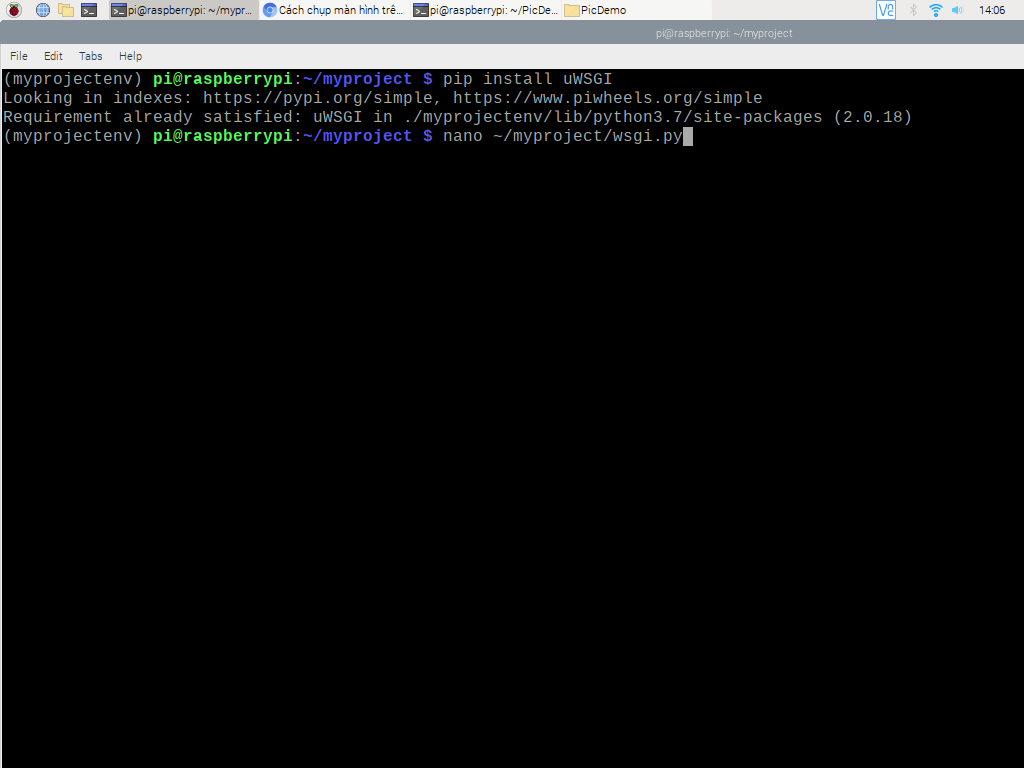
$ nano ~/myproject/wsgi.py

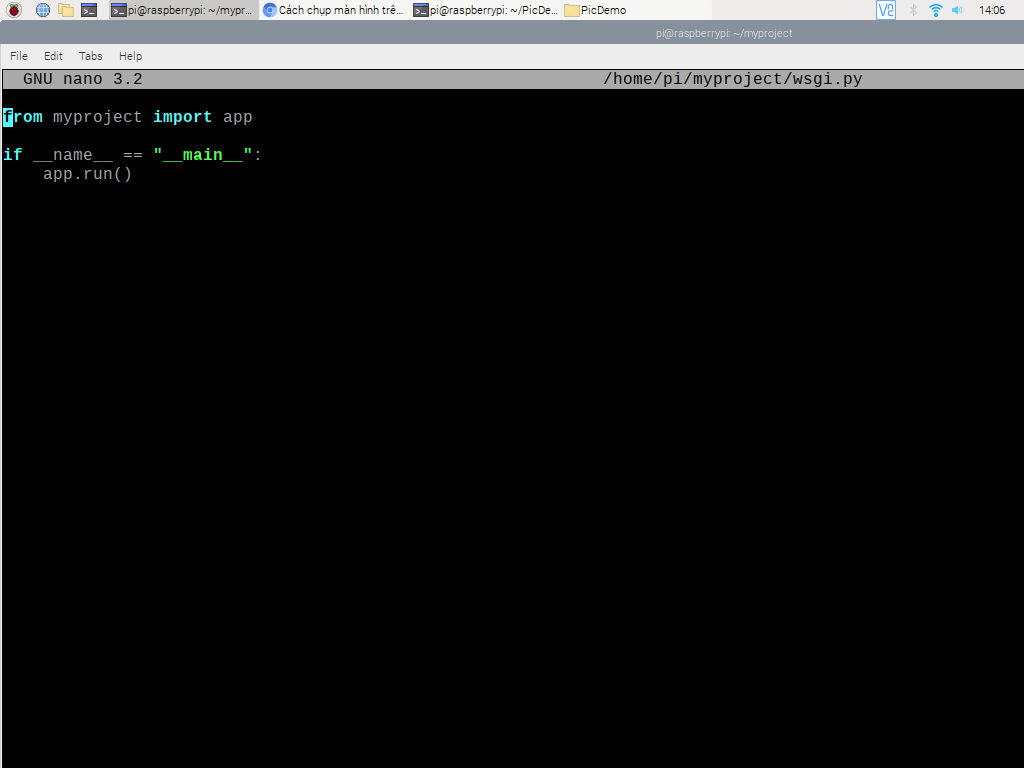
**from** myproject **import** app

**if** \_\_name\_\_ == "\_\_main\_\_":

app.run()

**Demo:**

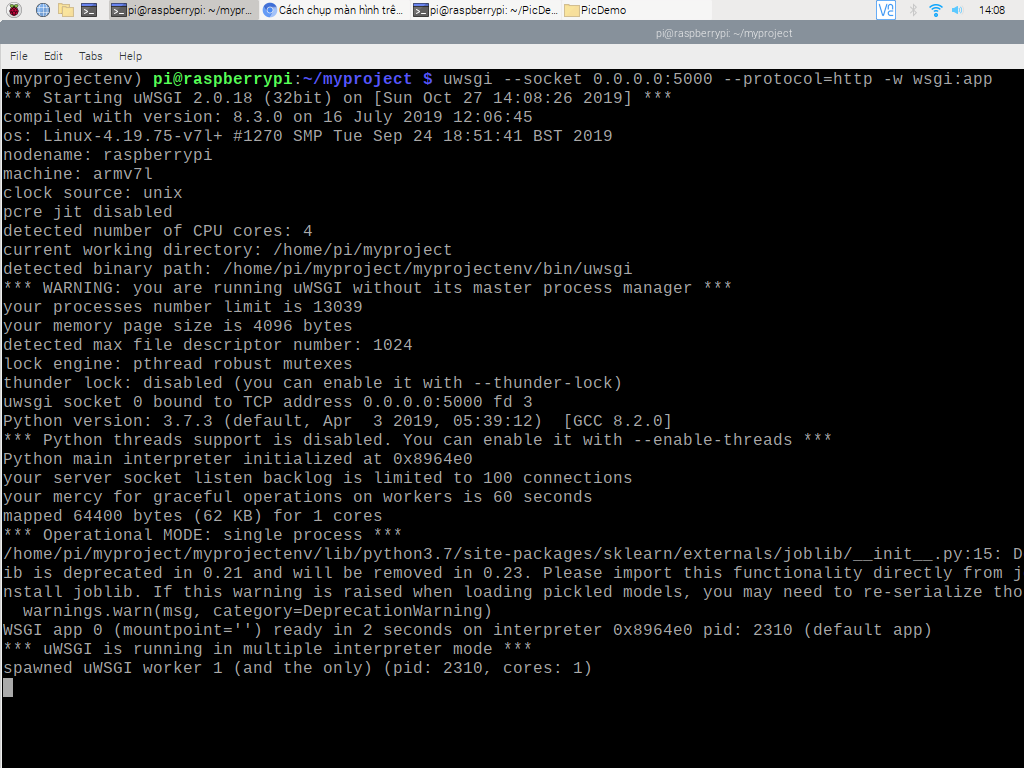




**Kiểm thử uWSGI Serving:**

$ uwsgi --socket 0.0.0.0:5000 --protocol=http -w wsgi:app

**Demo:**



**Tạo file cấu hình uWSGI:**

[uwsgi]

module = wsgi:app

master = true

processes = 5

socket = myproject.sock

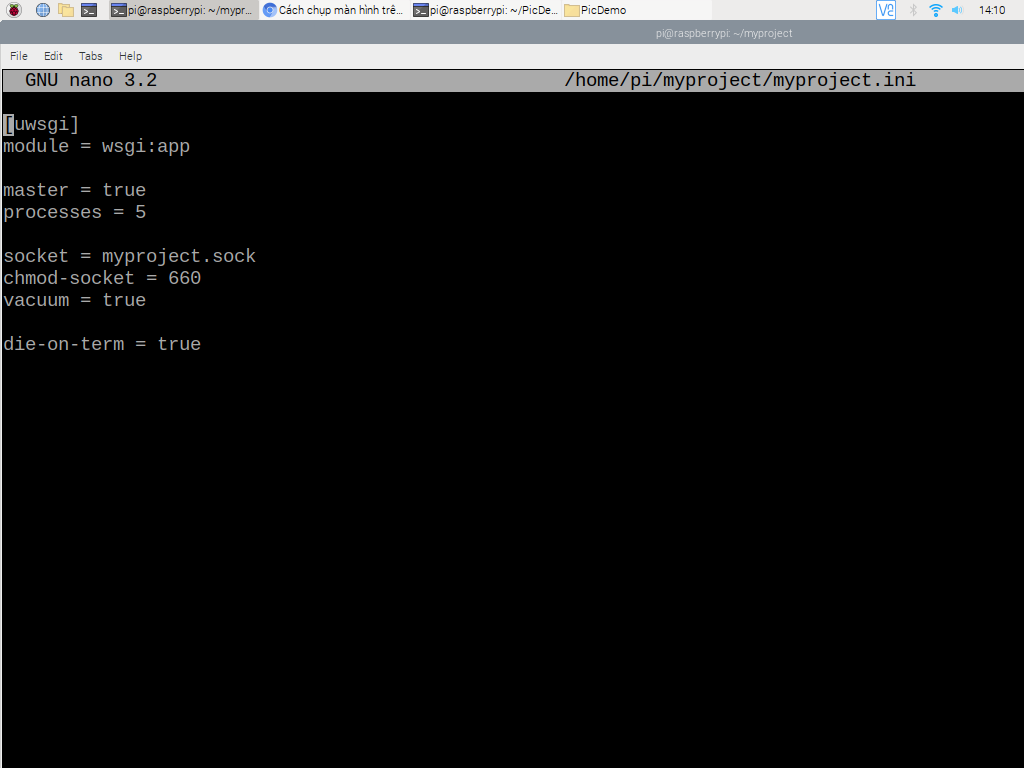
chmod-socket = 660

vacuum = true

die-on-term = true

**Demo:**





**Tạo file Unit hệ thống:**

$ sudo nano /etc/systemd/system/myproject.service

[Unit]

Description=uWSGI instance to serve myproject

After=network.target

[Service]

User=pi

Group=www-data

WorkingDirectory=/home/pi/myproject

Environment="PATH=/home/pi/myproject/myprojectenv/bin"

ExecStart=/home/pi/myproject/myprojectenv/bin/uwsgi --ini myproject.ini

[Install]

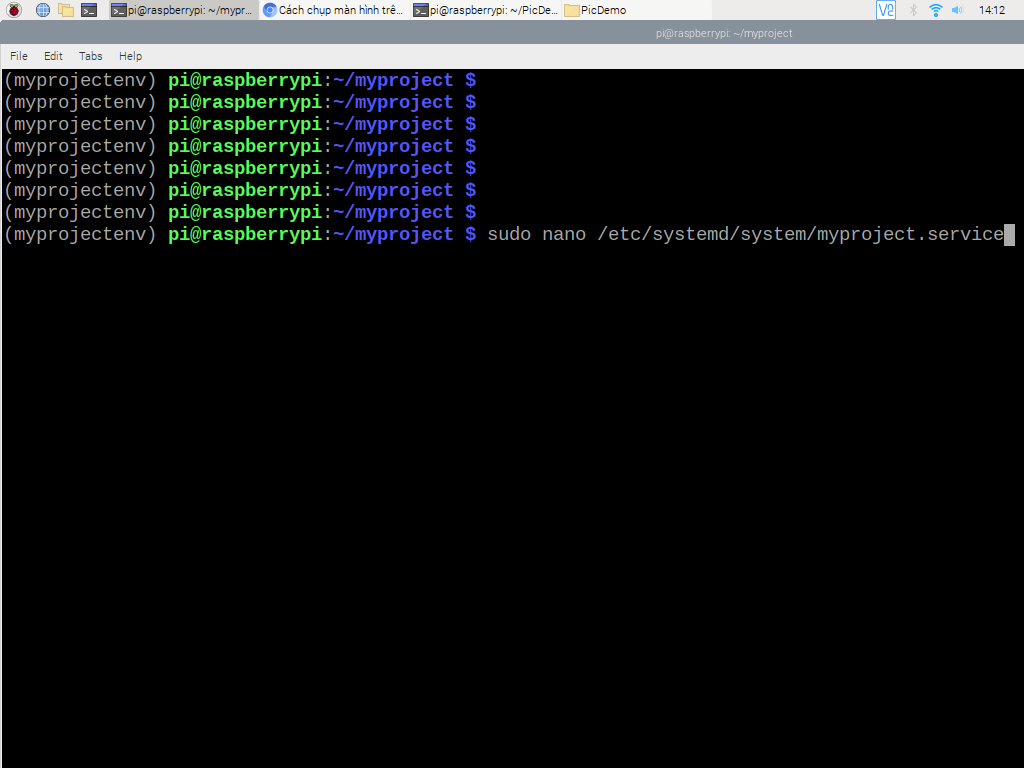
WantedBy=multi-user.target

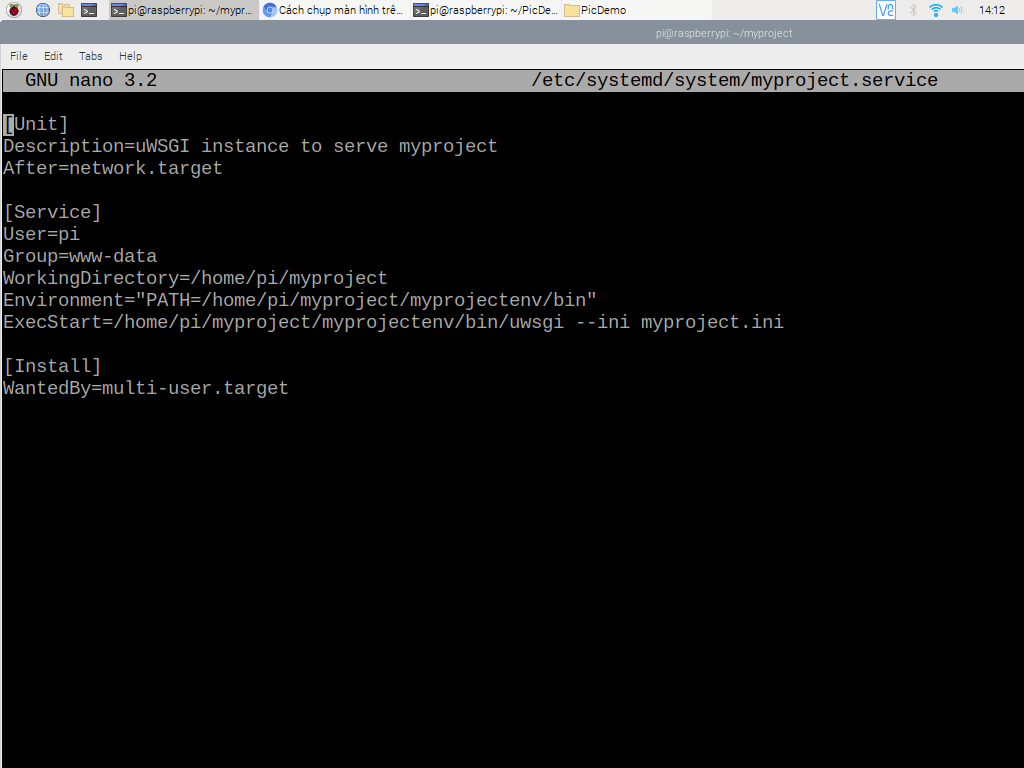
$ sudo systemctl start myproject

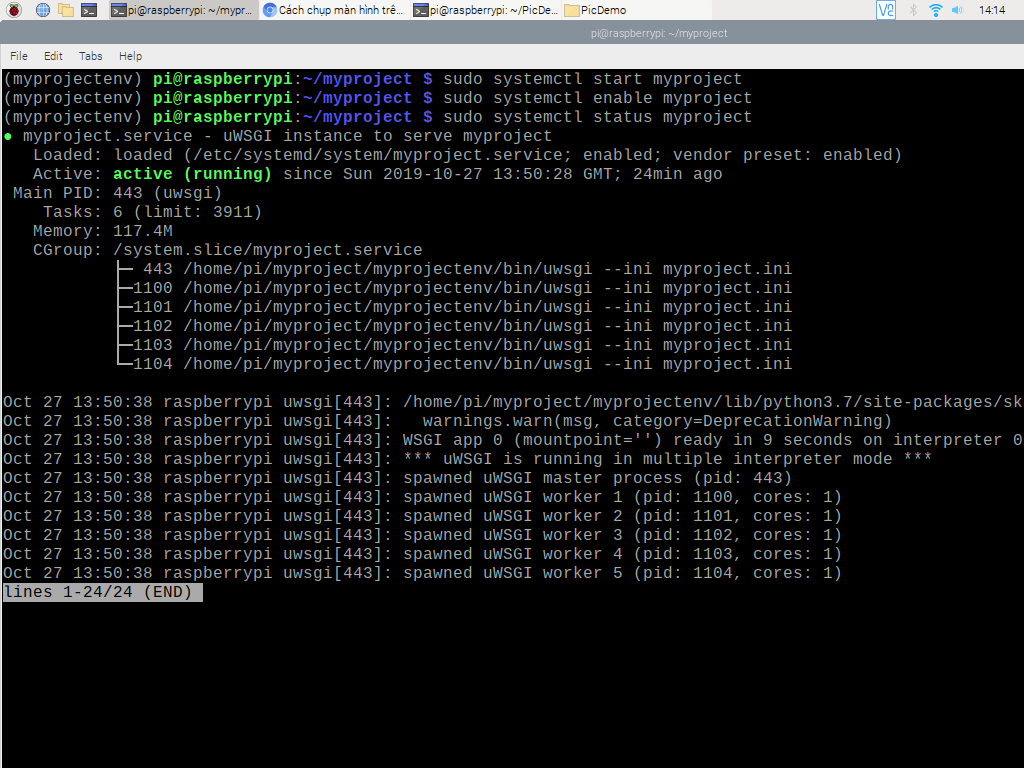
$ sudo systemctl enable myproject

$ sudo systemctl status myproject

**Demo:**







* 1. **Cấu hình Nginx làm Proxy Requests:**

$ sudo nano /etc/nginx/sites-available/myproject

server {

listen 80;

server\_name your\_domain www.your\_domain;

location / {

include uwsgi\_params;

uwsgi\_pass unix:/home/pi/myproject/myproject.sock;

}

}

$ sudo ln -s /etc/nginx/sites-available/myproject /etc/nginx/sites-enabled

$ sudo nginx -t

$ sudo systemctl restart nginx

$ sudo ufw delete allow 5000

$ sudo ufw allow 'Nginx Full'

Giờ ta có thể điều hướng đến tên miền máy chủ trên trình duyệt:

<http://your_domain>

Tham khảo:

<https://www.digitalocean.com/community/tutorials/how-to-serve-flask-applications-with-uswgi-and-nginx-on-ubuntu-18-04?fbclid=IwAR0mcRjb5ORPB4srbFZxIAIe0TDt5rtUgZN0HRaBdf1LHKXOqhIaS5LnBGQ>

<https://www.kaggle.com/uciml/sms-spam-collection-dataset>

<https://towardsdatascience.com/develop-a-nlp-model-in-python-deploy-it-with-flask-step-by-step-744f3bdd7776>

Code hệ thống đề tài “**Hệ thống nhận diện SMS Spam**”:

<https://github.com/orasentidolhl99/LTHT-Spam>