# **Python Startups :**

* It is a Popular programming language and can be used on a server to create web applications.
* Python is a **high-level**, **general-purpose**, **interpreted** programming language.

1. **High Level :**

* It is a high-level programming language.
* It's simple and so easy to learn.

1. **General Purpose :**

* It is a general-purpose language.
* It means that you can use Python in various domains including:
  + Web applications
  + Big data applications
  + Testing
  + Automation
  + Data science, machine learning, and AI
  + Desktop software
  + Mobile apps

1. **Interpreted :** 
   1. It is an interpreted language.
   2. interpreter is a computer program that converts each high-level program statement into machine code.
   3. To develop a Python program, you write Python code into a file called source code.
   4. To execute the source code, you need to convert to the machine language for the computer to understand.
   5. When the Python program executes, the Python interpreter turns the source code, line by line, once at a time, into the machine code.

## **Python Uses :**

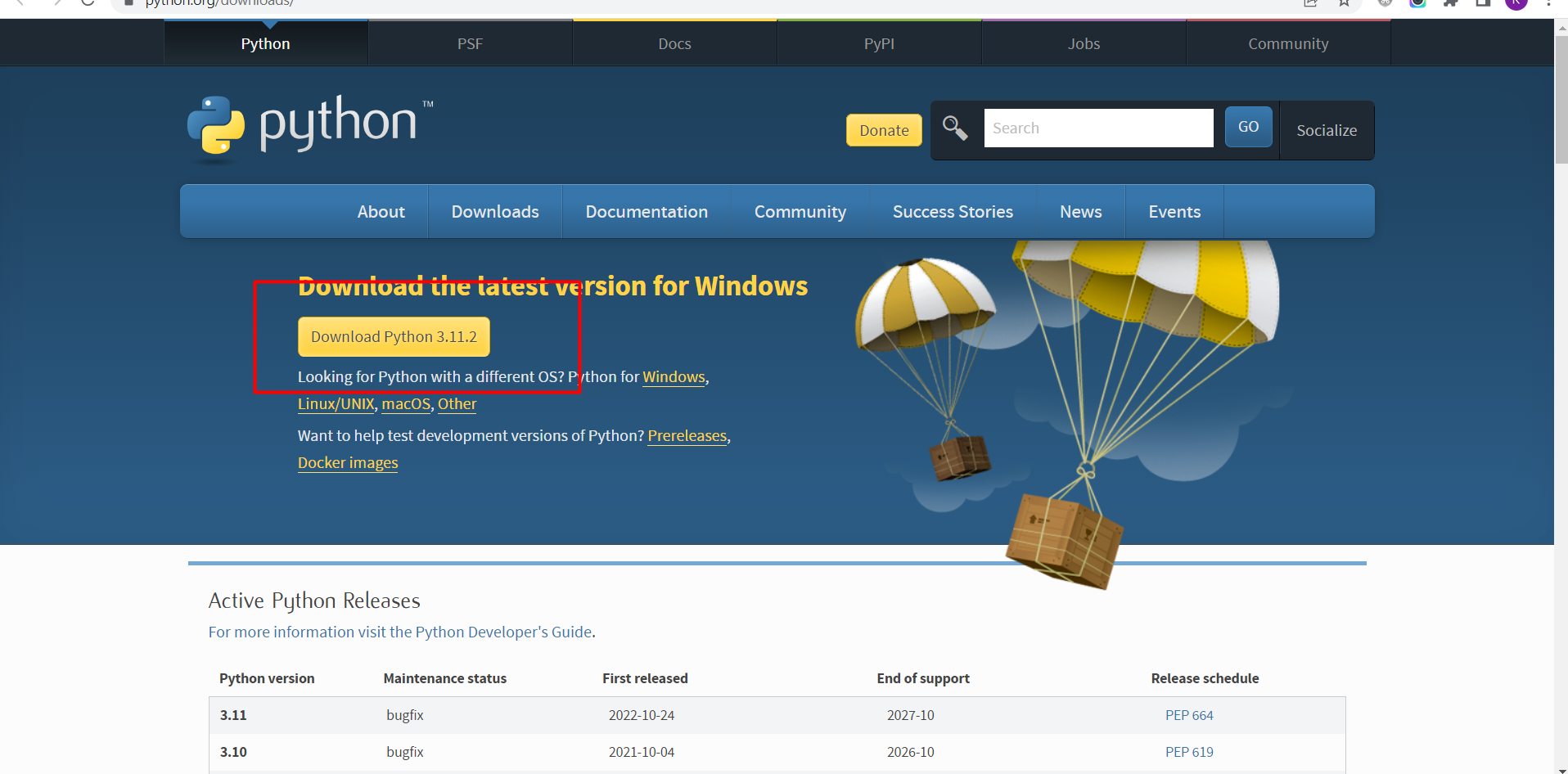
* It allows you to solve complex problems in less time and fewer lines of code.
* It is quite easy to learn in comparison with other programming languages.
* syntax is clear and beautiful.
* It is cross-platform. Python programs can run on Windows, Linux, and macOS.
* It has a huge community. Whenever you get stuck, you can get help from an active community.
* It is much in demand and ensures a high salary.
* It can be used to handle big data and perform complex mathematics.

## **Python Version :**

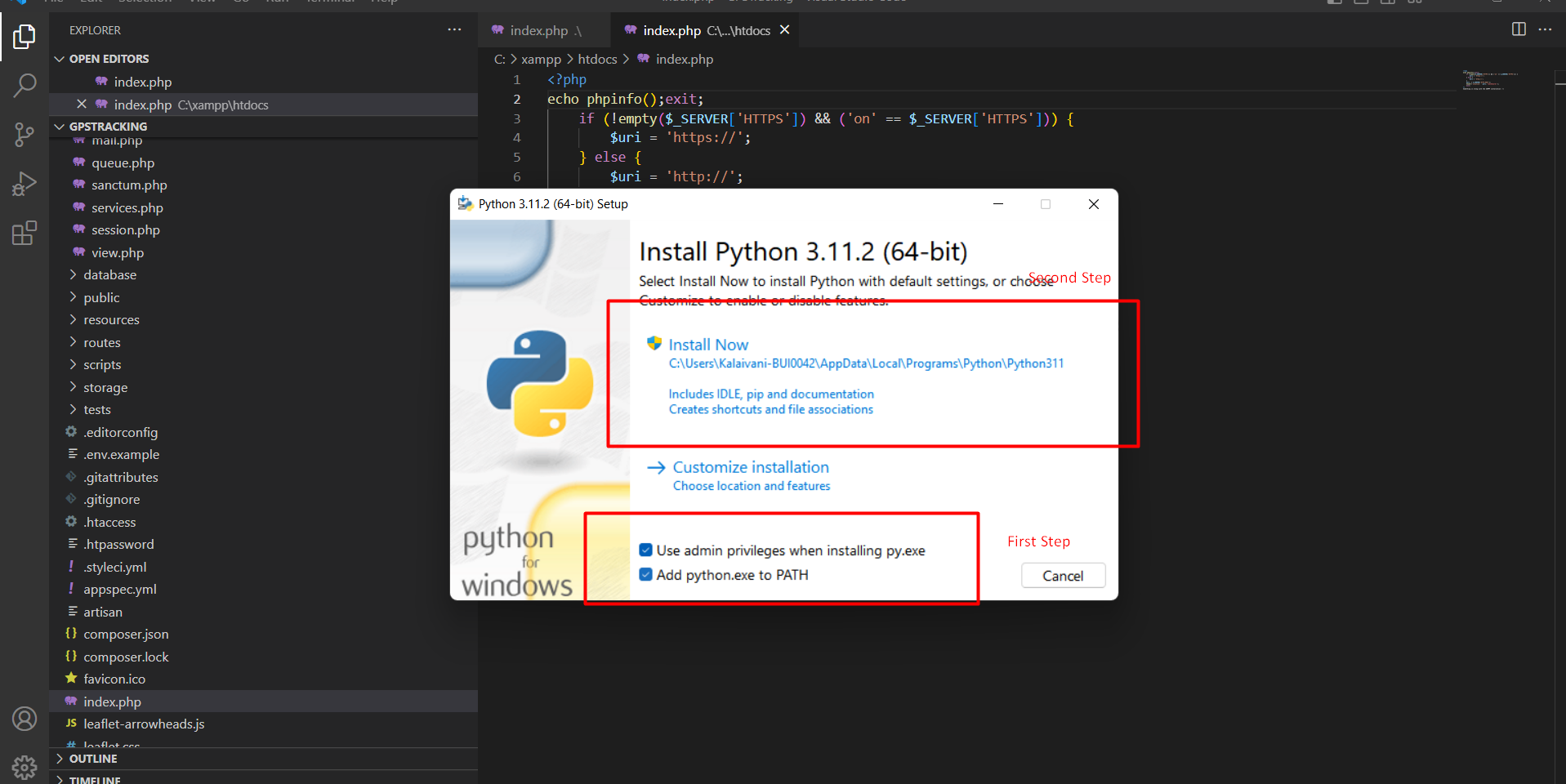
* It has two major versions: 2x and 3x.
* Python 2.x was released in 2000.
* Python 3.x was released in 2008.Basically, Python 3 isn’t compatible with Python 2. And you should use the latest versions of Python 3 for your new projects.

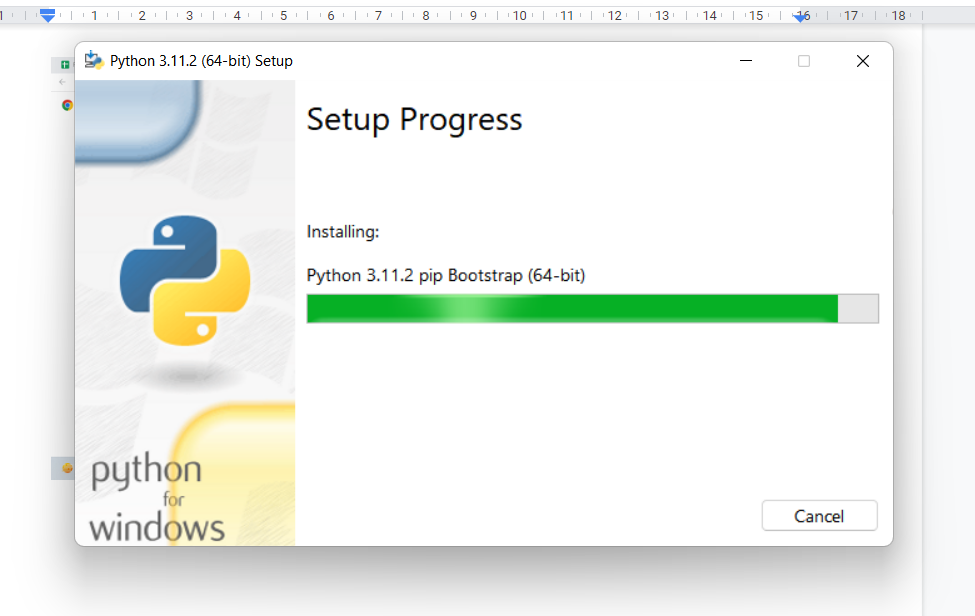
## **Installation :**

* First Download Latest Version from following link :
  + <https://www.python.org/downloads/>

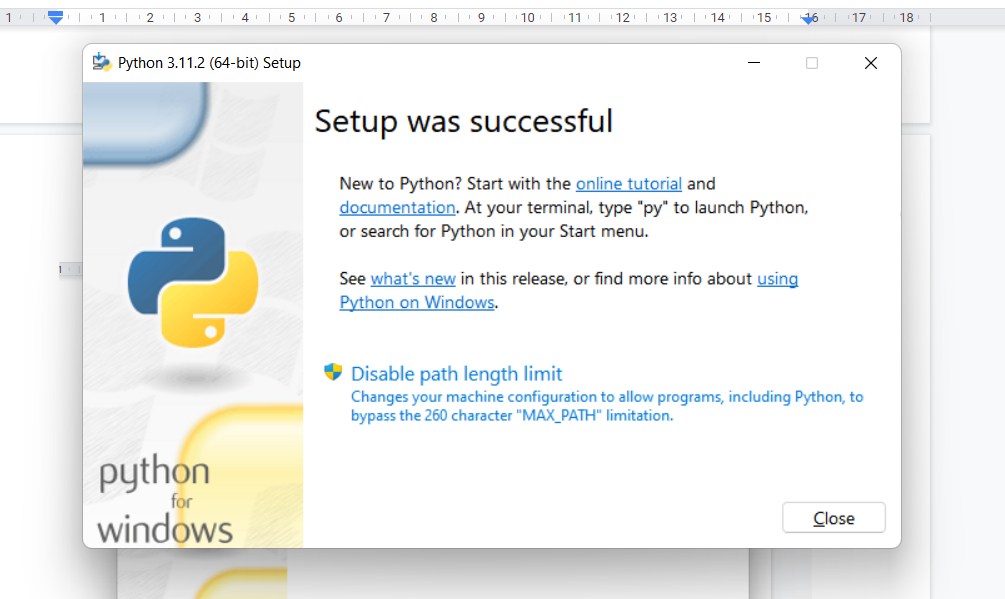


* double-click the installer file to launch the setup wizard
* Check Add Python,exe to Path and Next Click Install Now

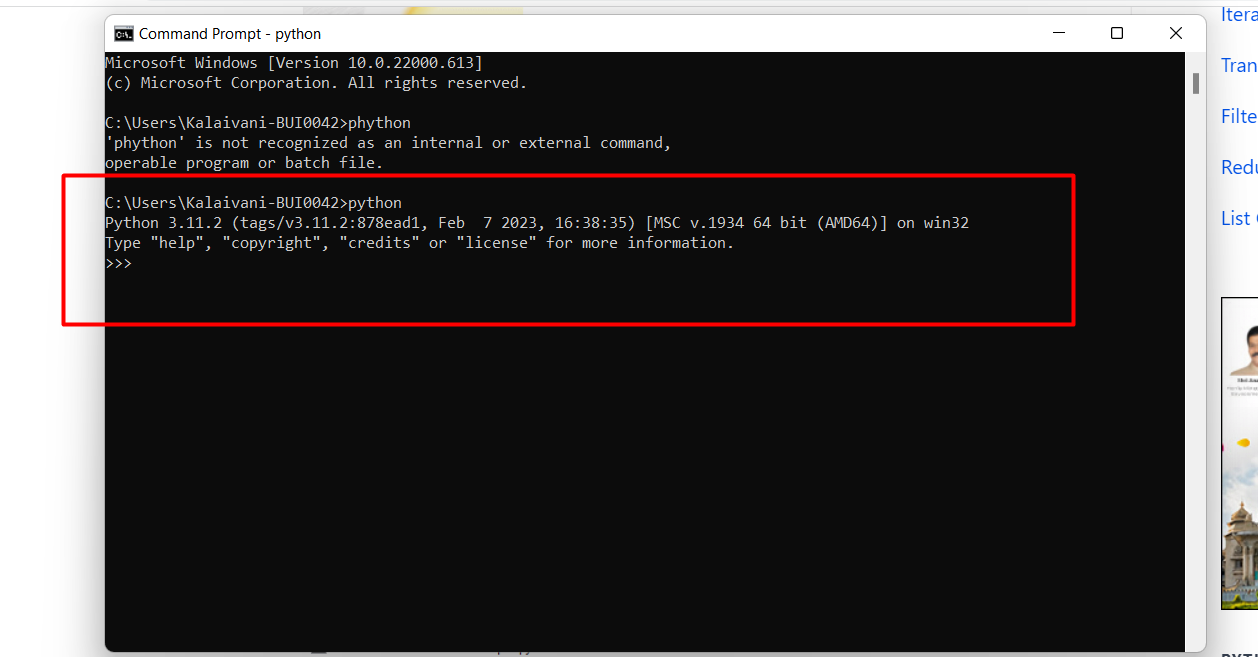




* Once setup is done you will see the following screen .

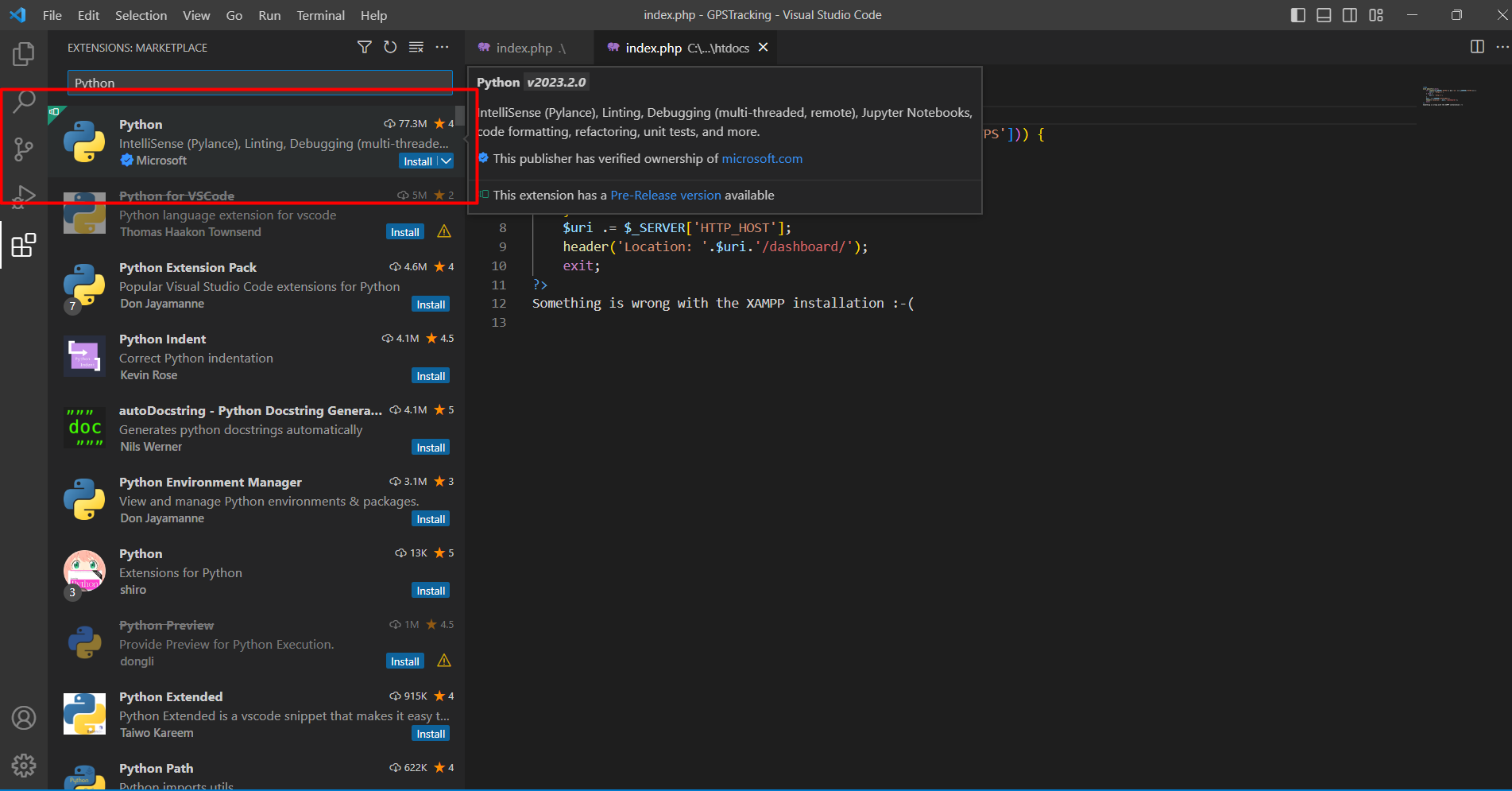


* To Verify Installation to run python in cmd.



## **Setup Python in VS Code :**

* Install Python extension in VS Code.

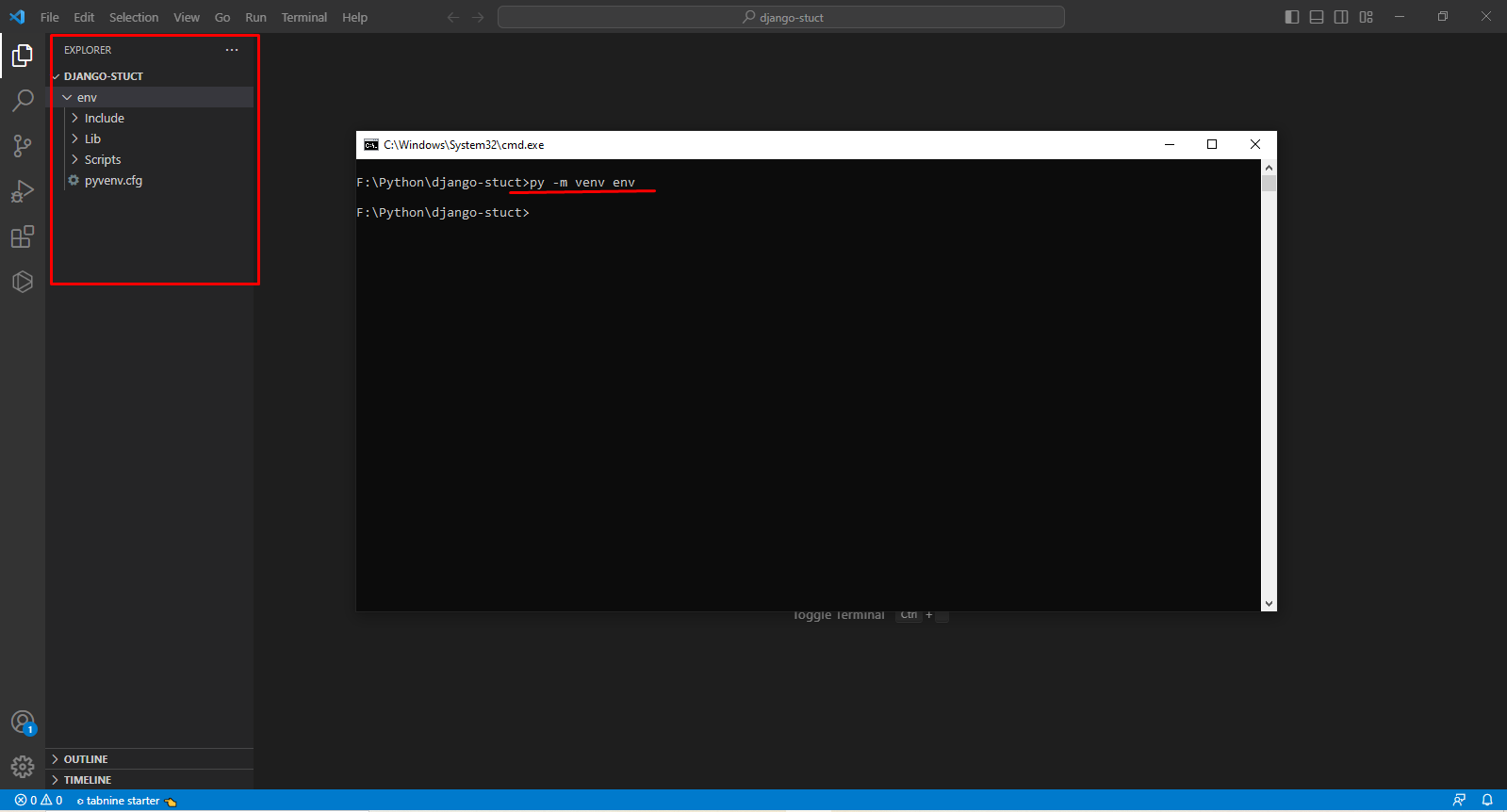
****

## Django Project Structure :

Creating a virtual environment

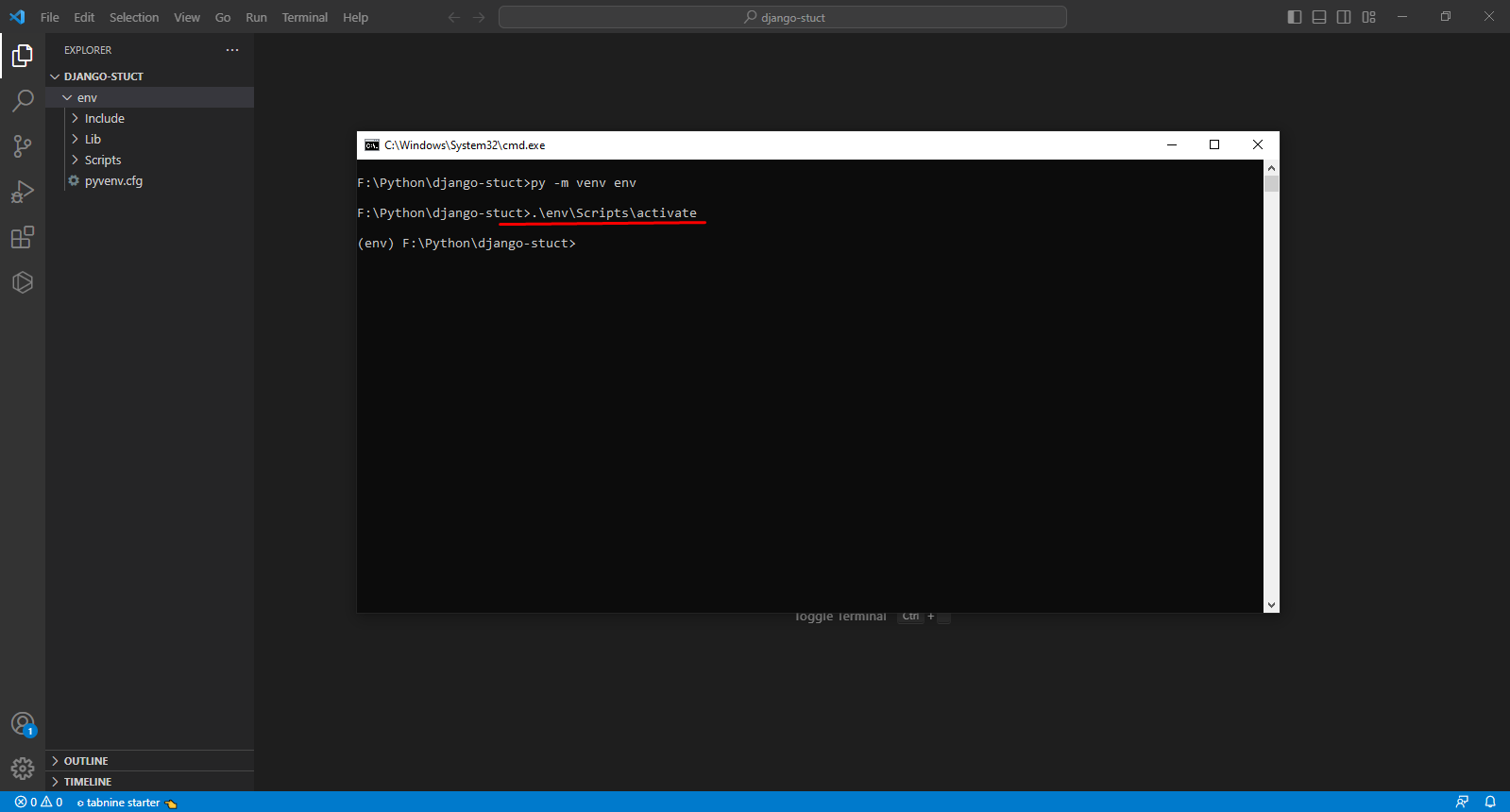
**venv** (for Python 3) and **virtualenv** (for Python 2) allow you to manage separate package installations for different projects.

py -m venv env



Before you can start installing or using packages in your virtual environment **you’ll need to activate it**. Activating a virtual environment will put the virtual environment-specific python and pip executables into your shell’s PATH.

F:\Python\django-stuct>.\env\Scripts\activate



* If not env activate try this 👍

### **Solution:**

1. **Open PowerShell as Administrator**

* Press Windows + S, type PowerShell, right-click, and choose **Run as Administrator**.

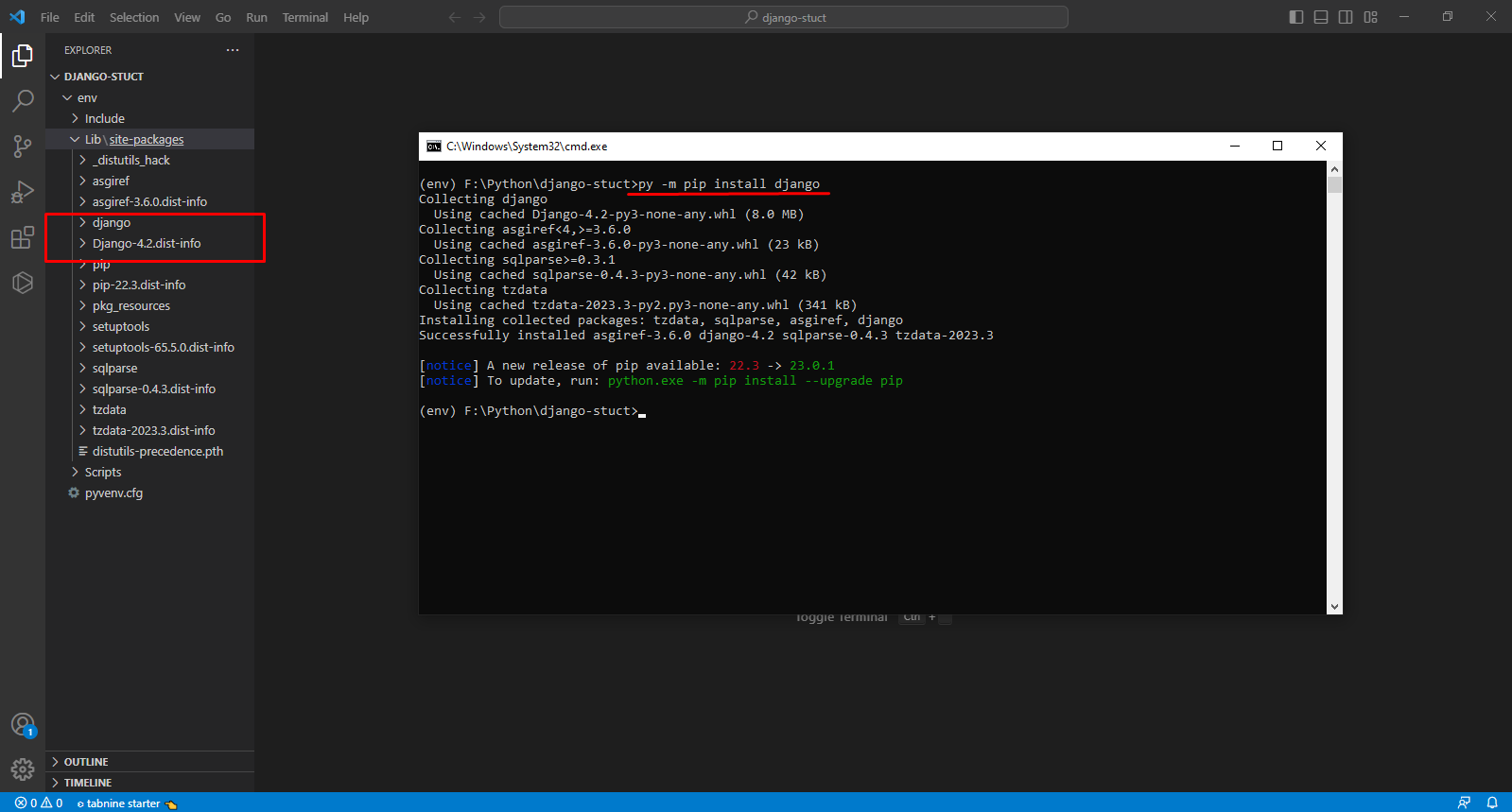
1. **Run the Following Command: Set-ExecutionPolicy RemoteSigned**
2. **If prompted, press Y and hit Enter.**
3. **Try Activating the Environment Again:**

**Powershell/cmd :**

**Copy code: .\env\Scripts\activate**

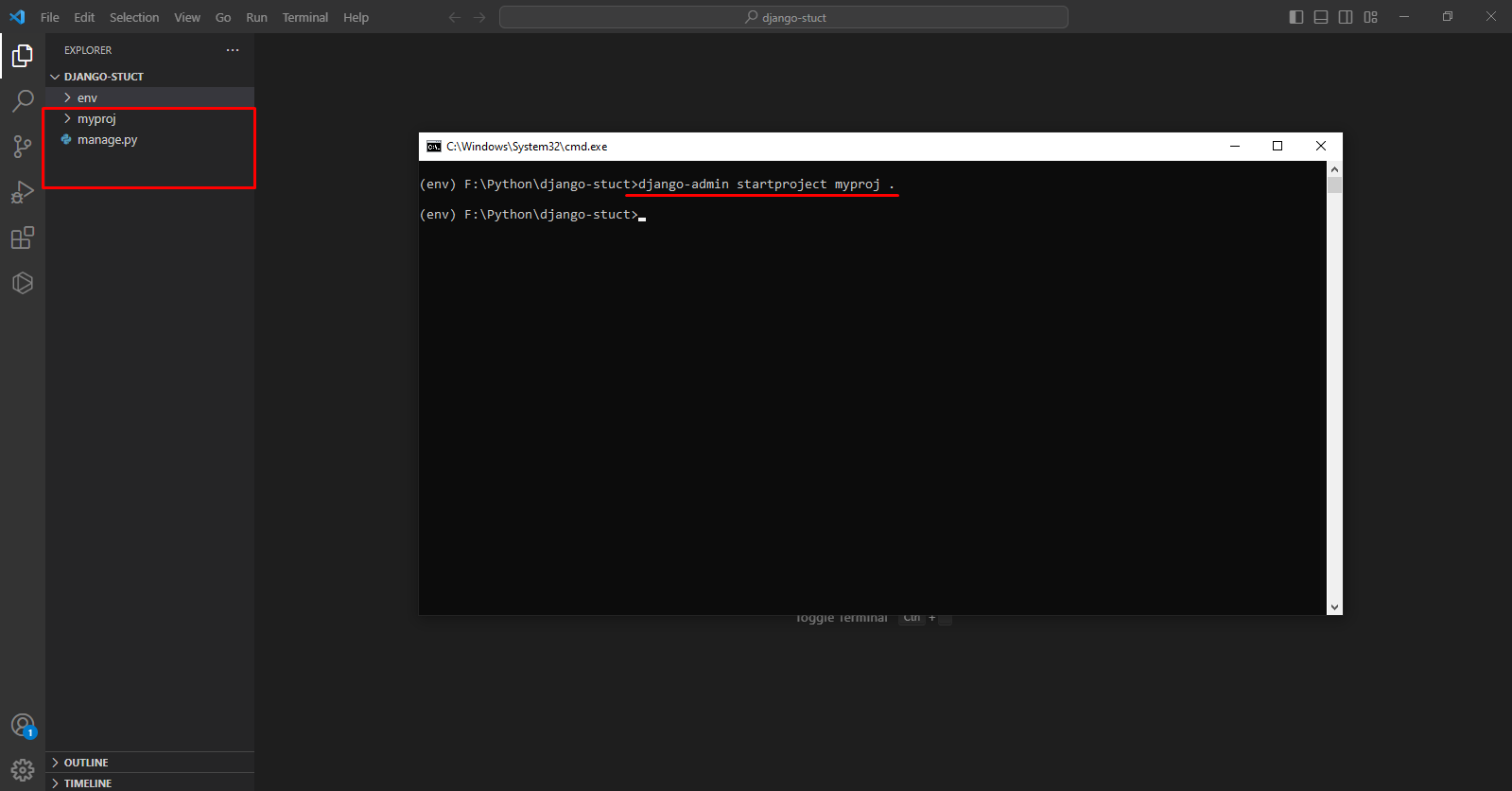
-install Django in your Virtual Environment, we will be using Pip3 for install Django.

py -m pip install django

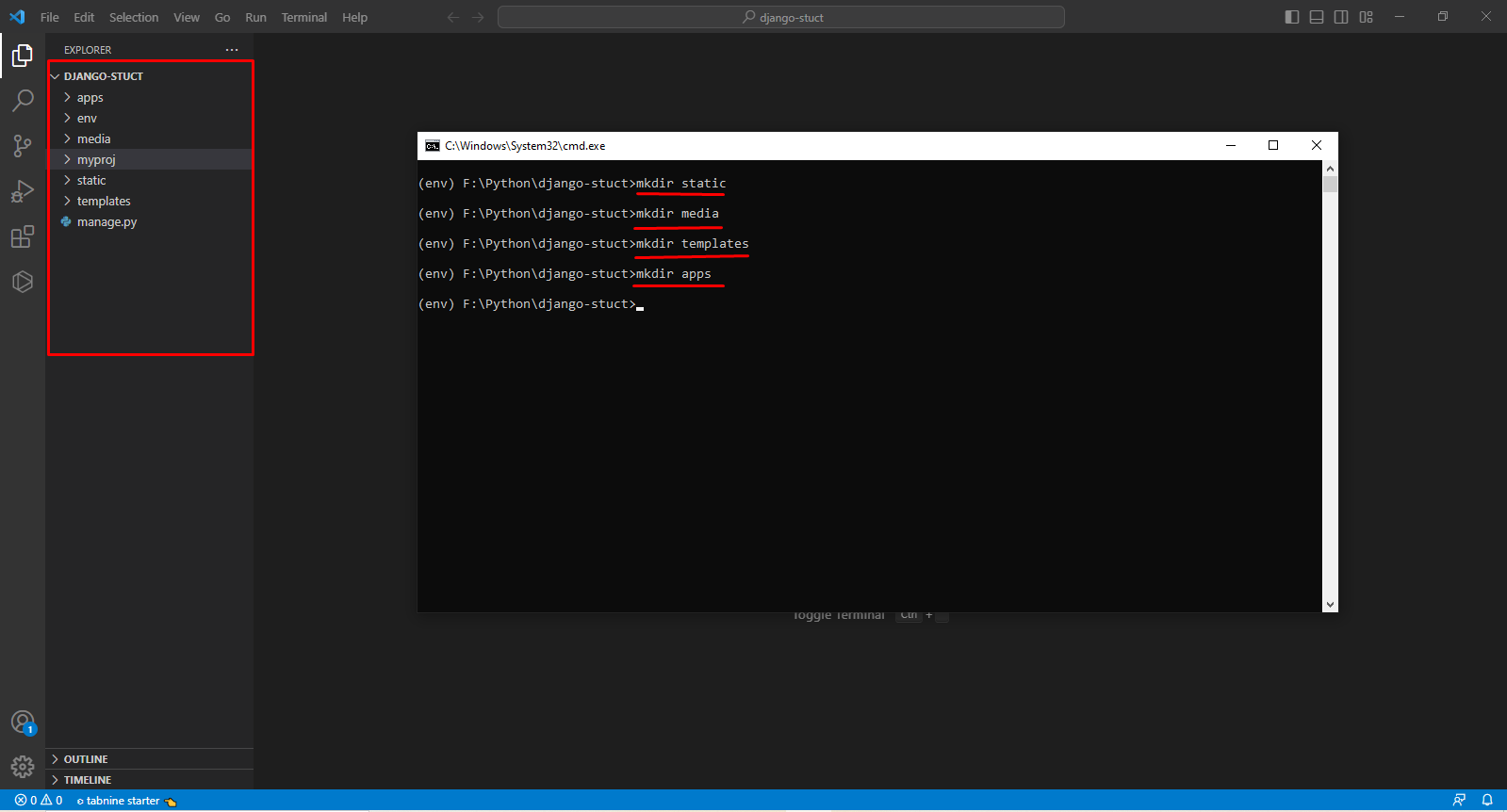


- Now create a Django Project using the below command

django-admin startproject myproj .



- Now that we have created a Django project, within the same directory we will be creating apps, media, static and templates folders.



* Add the following lines in **setting.py**  file

STATIC\_URL = '/static/'

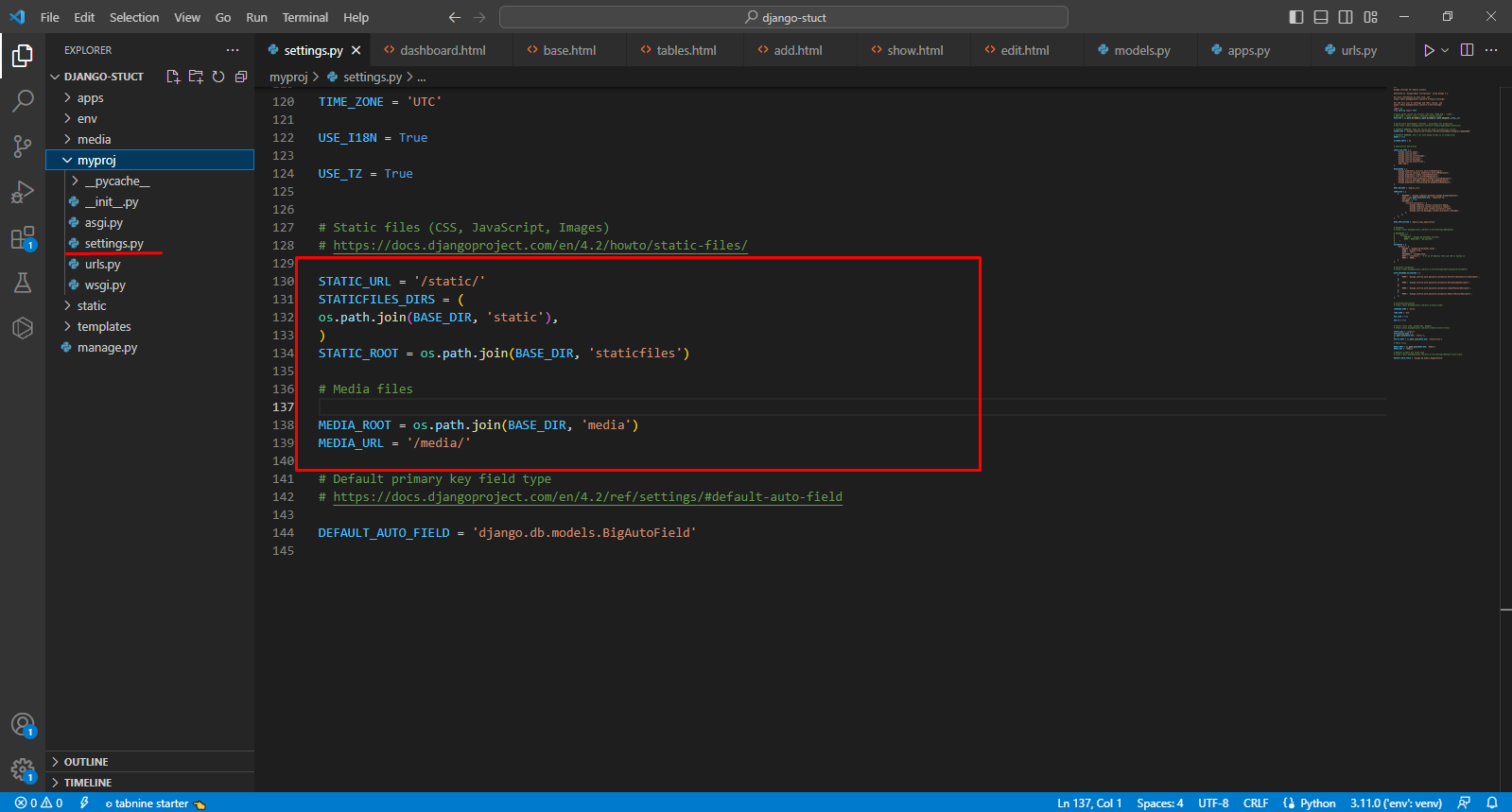
STATICFILES\_DIRS = (os.path.join(BASE\_DIR, 'static'), )

STATIC\_ROOT = os.path.join(BASE\_DIR, 'staticfiles')

# Media files

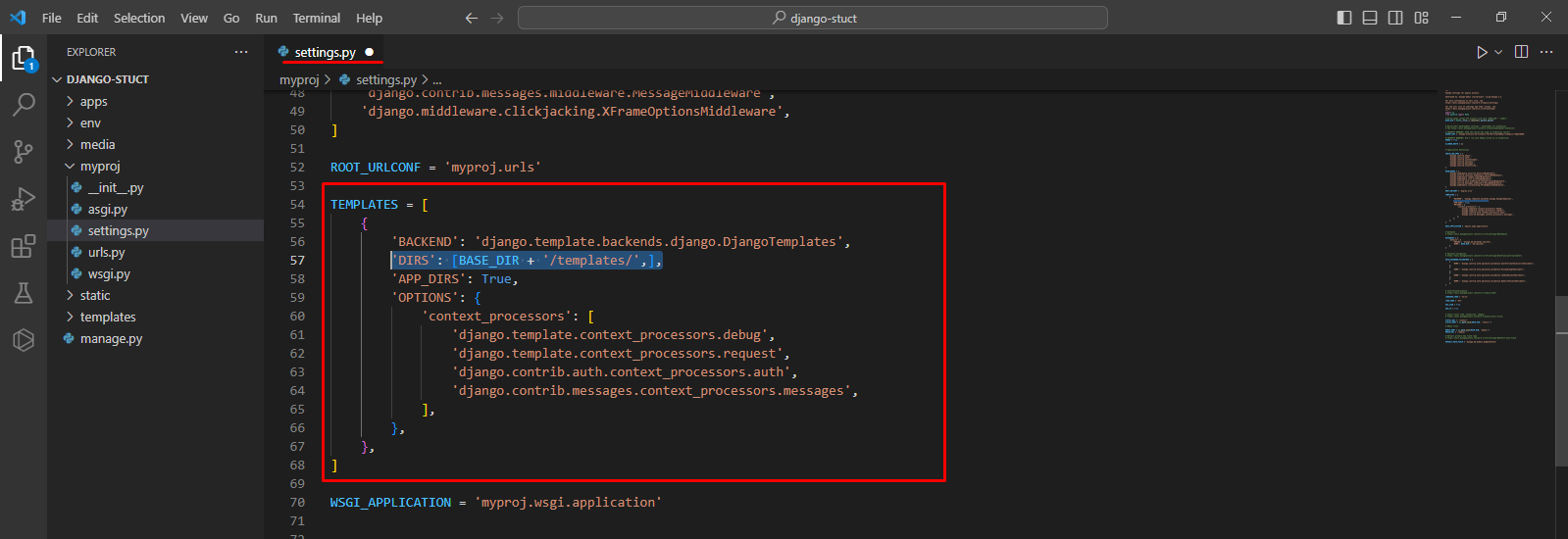
MEDIA\_ROOT = os.path.join(BASE\_DIR, 'media')

MEDIA\_URL = '/media/'



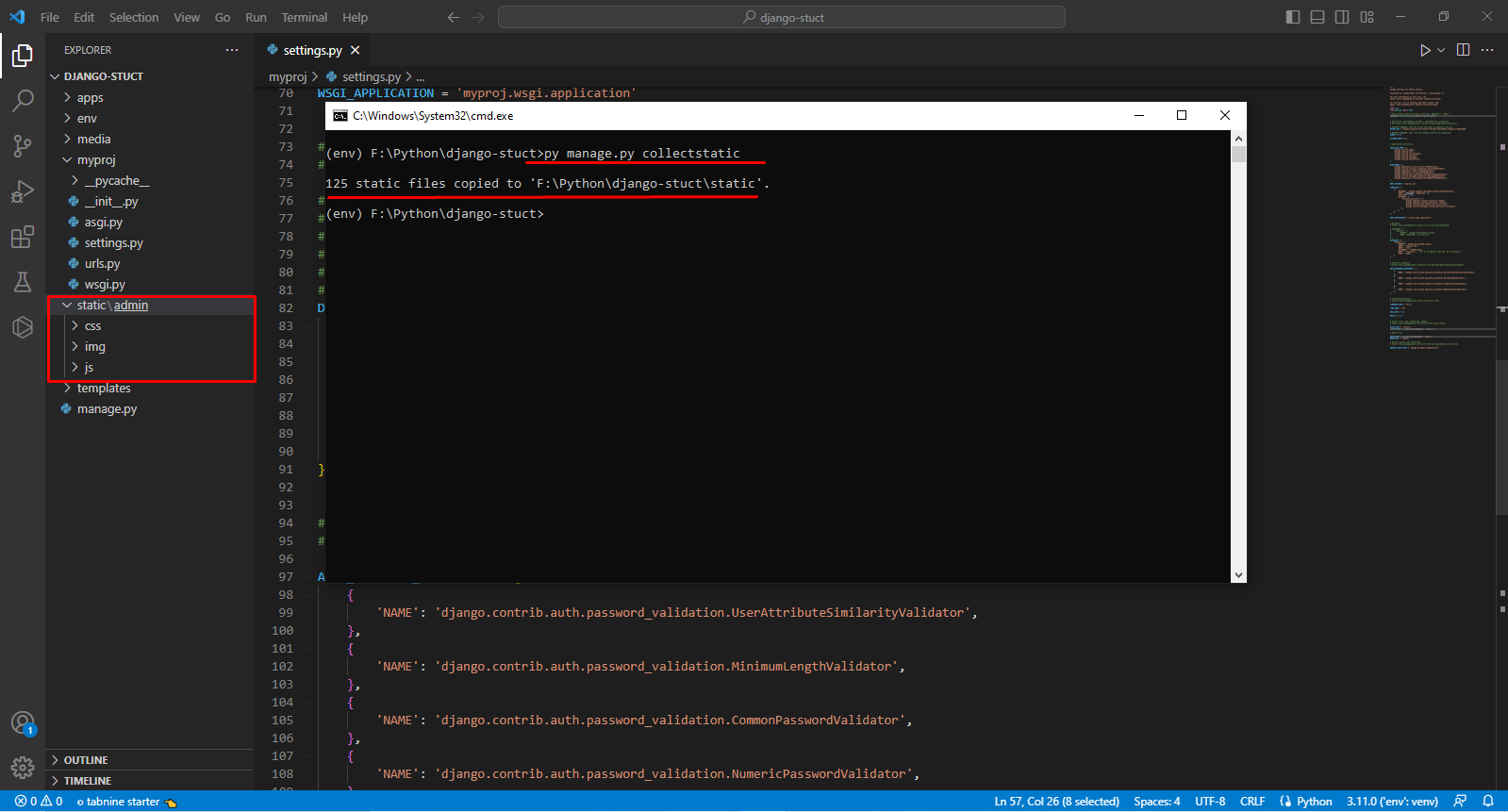
* Add the template path in **settings.py** file

'DIRS': [BASE\_DIR,'/templates/',],



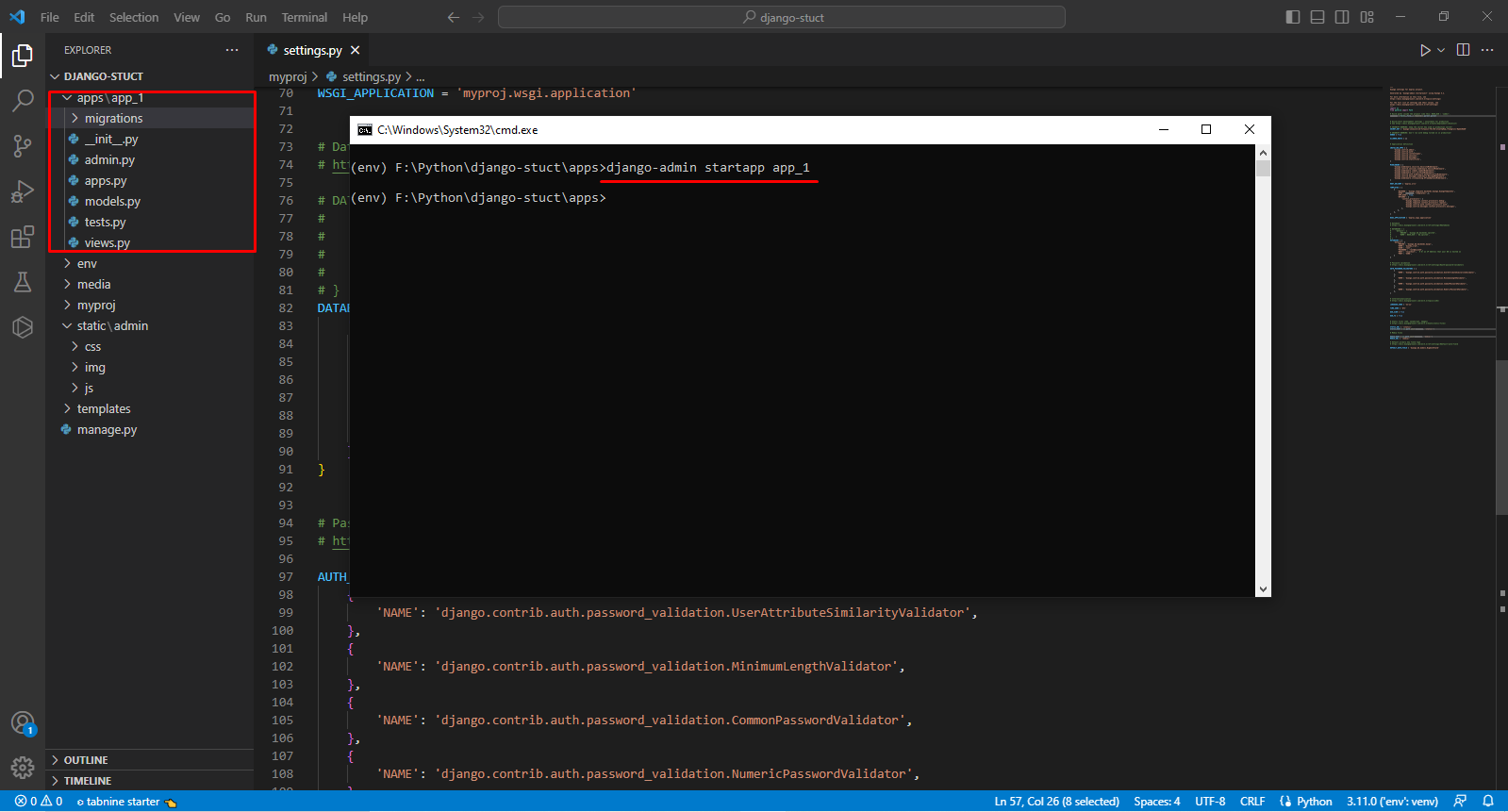
**django.contrib.staticfile**s provides a way of **gathering static files in a single directory**, so you can serve them easily. We have set it above STATIC\_ROOT. Now once we will **run the collectstatic command**, it will copy all static files to STATIC\_ROOT Folder.

py manage.py collectstatic



- For **creating Django project apps** you need to change your directory, **you need to go to apps folders** and inside apps folder, run the following command

django-admin startapp app\_1



- Add the app name which we have created to **INSTALLED\_APPS** list- in **setting.py** file

INSTALLED\_APPS = [

'django.contrib.admin',

'django.contrib.auth',

'django.contrib.contenttypes',

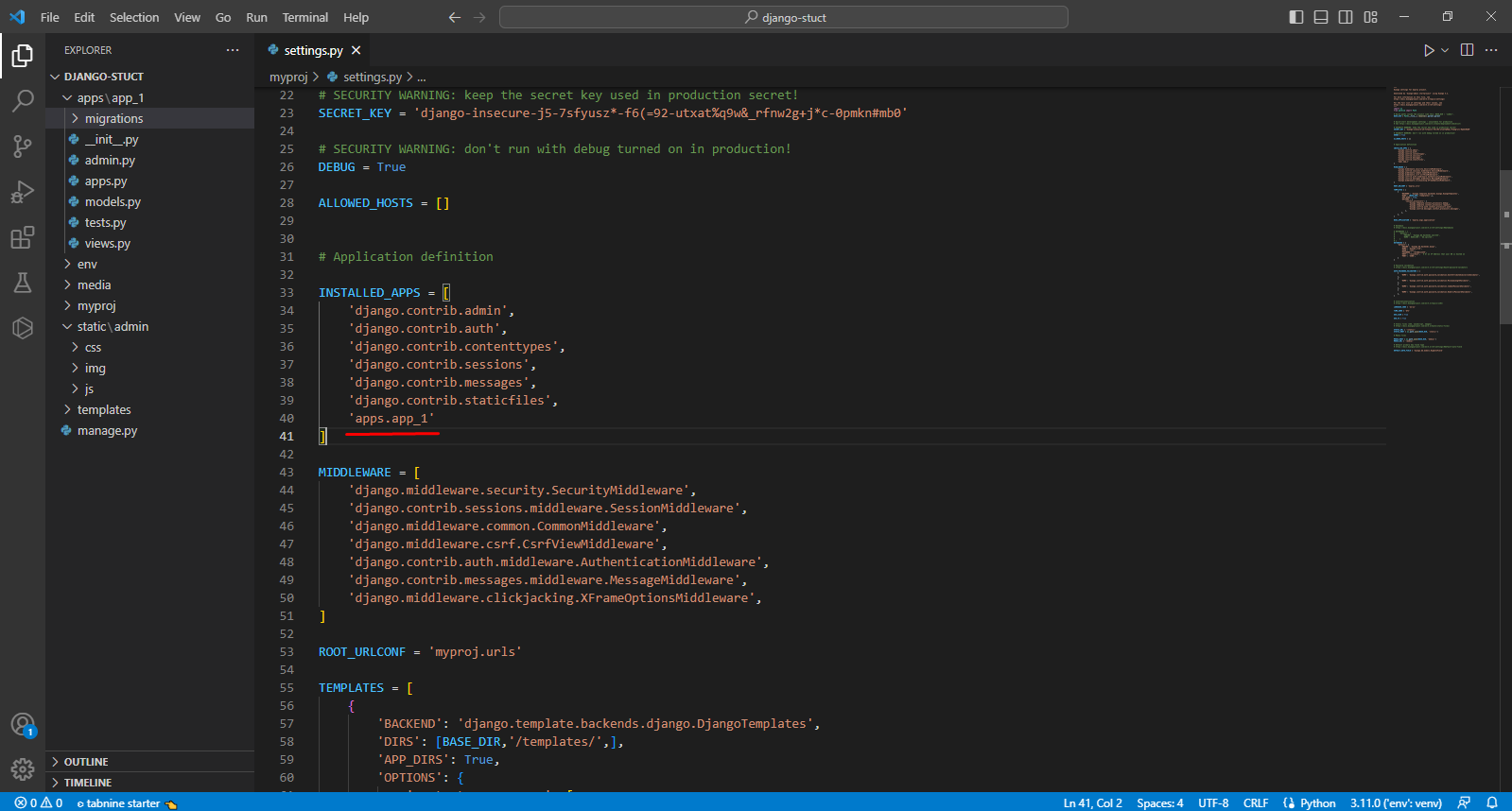
'django.contrib.sessions',

'django.contrib.messages',

'django.contrib.staticfiles',

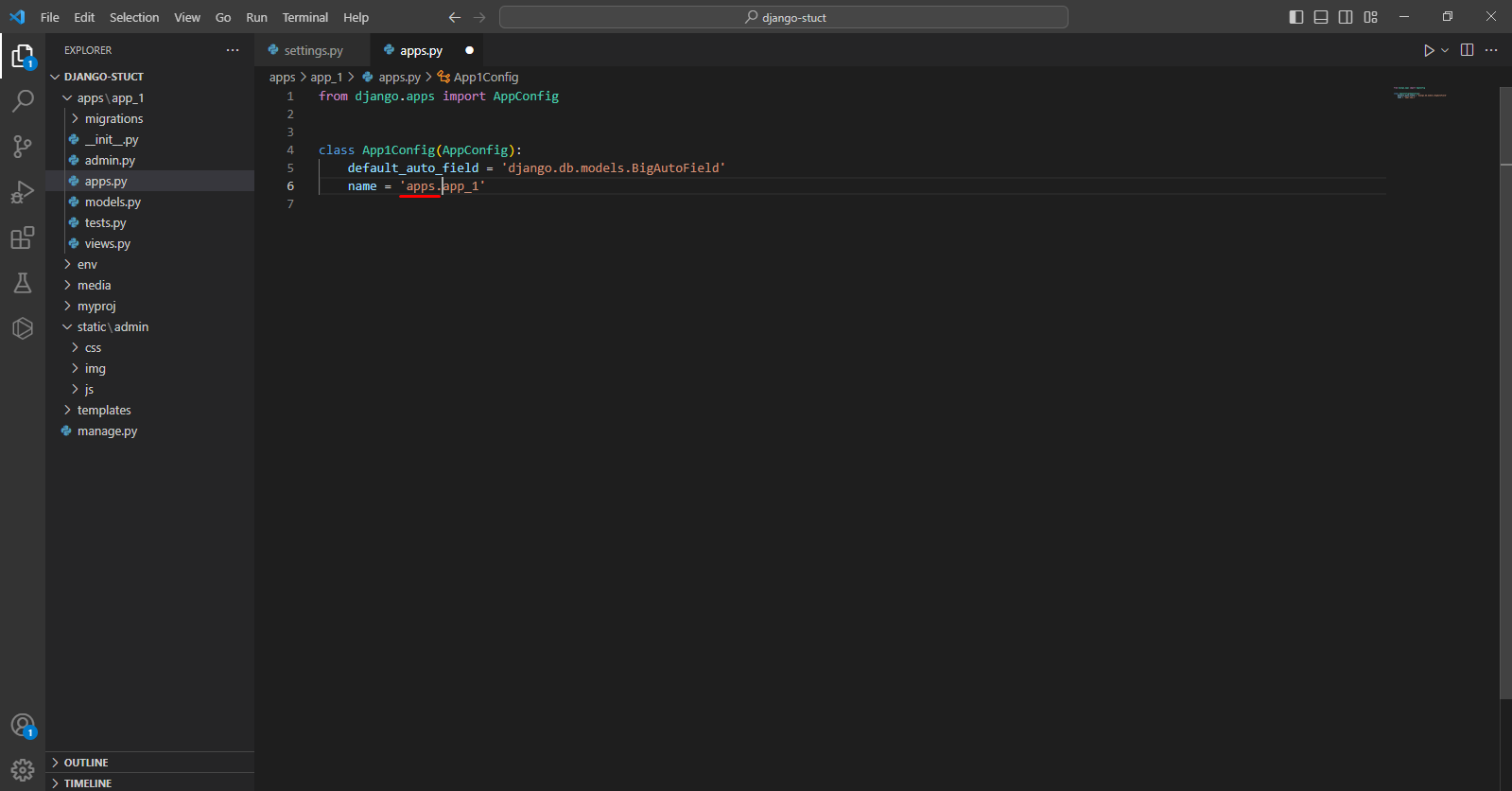
'apps.app\_1'

]



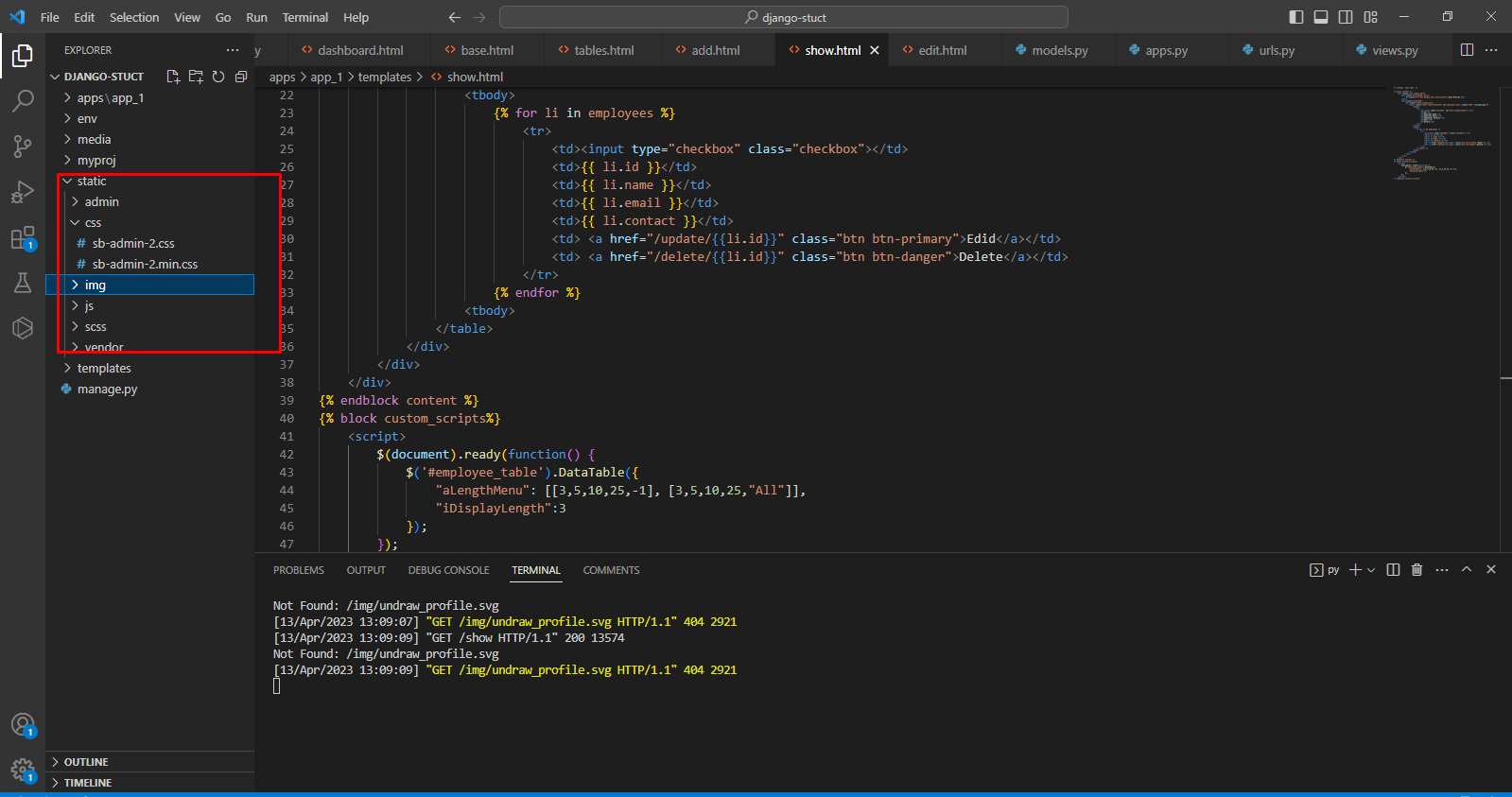
-Add the **apps** folder name in app config class name variable apps.py file

name = 'apps.app\_1’



## Django Theme Integration:

* copy theme css file and script,plugin file place to static folder project root directory.

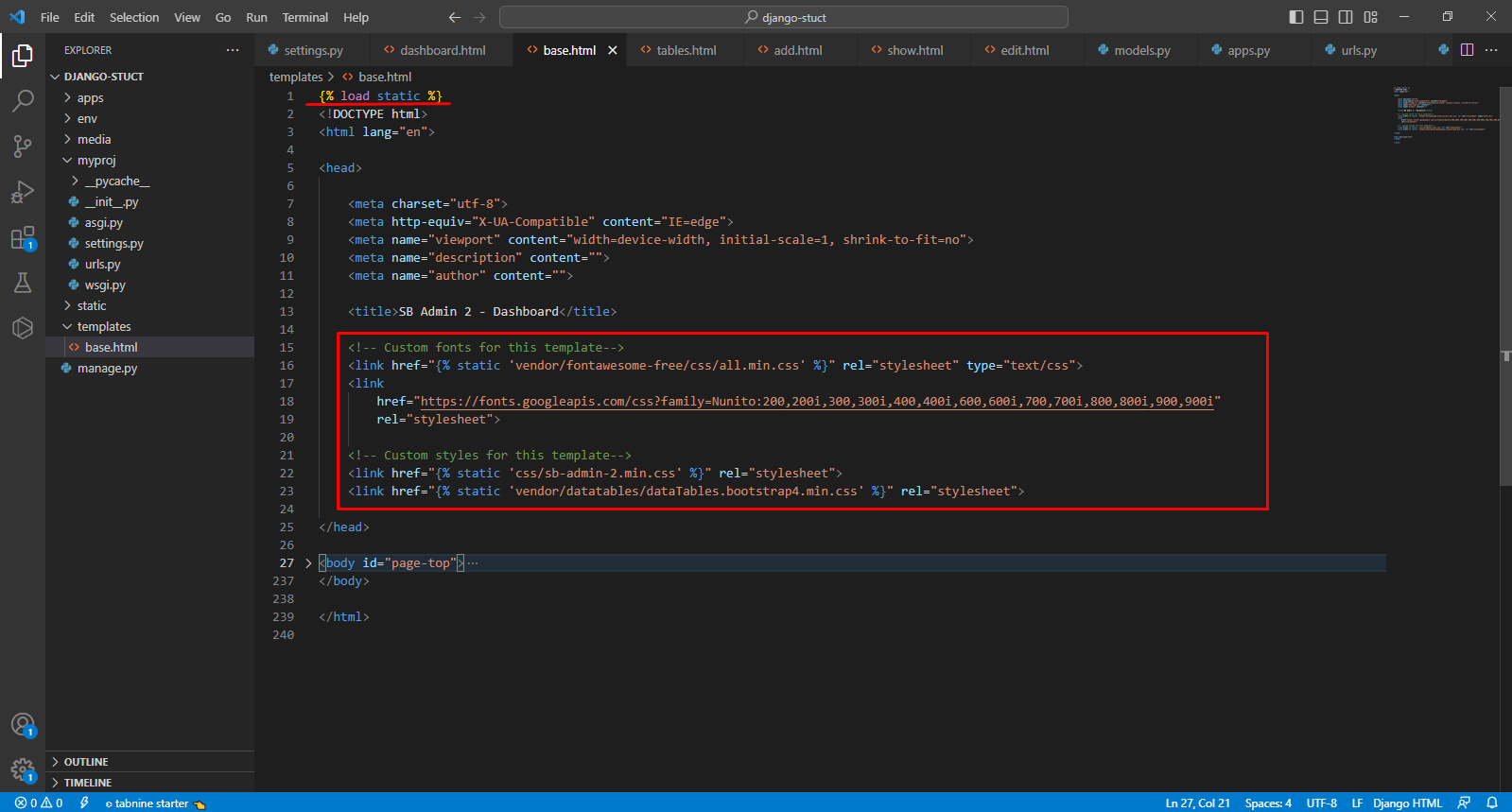


- For loading static content, we need to load it in our template. Edit your *base.html* and add the below code at the top above file

{% load static %}

- you have to use the static {% static ' ' %} command for loading all images, js, and CSS from the static folder.

<link href="{% static 'css/sb-admin-2.min.css' %}" rel="stylesheet">



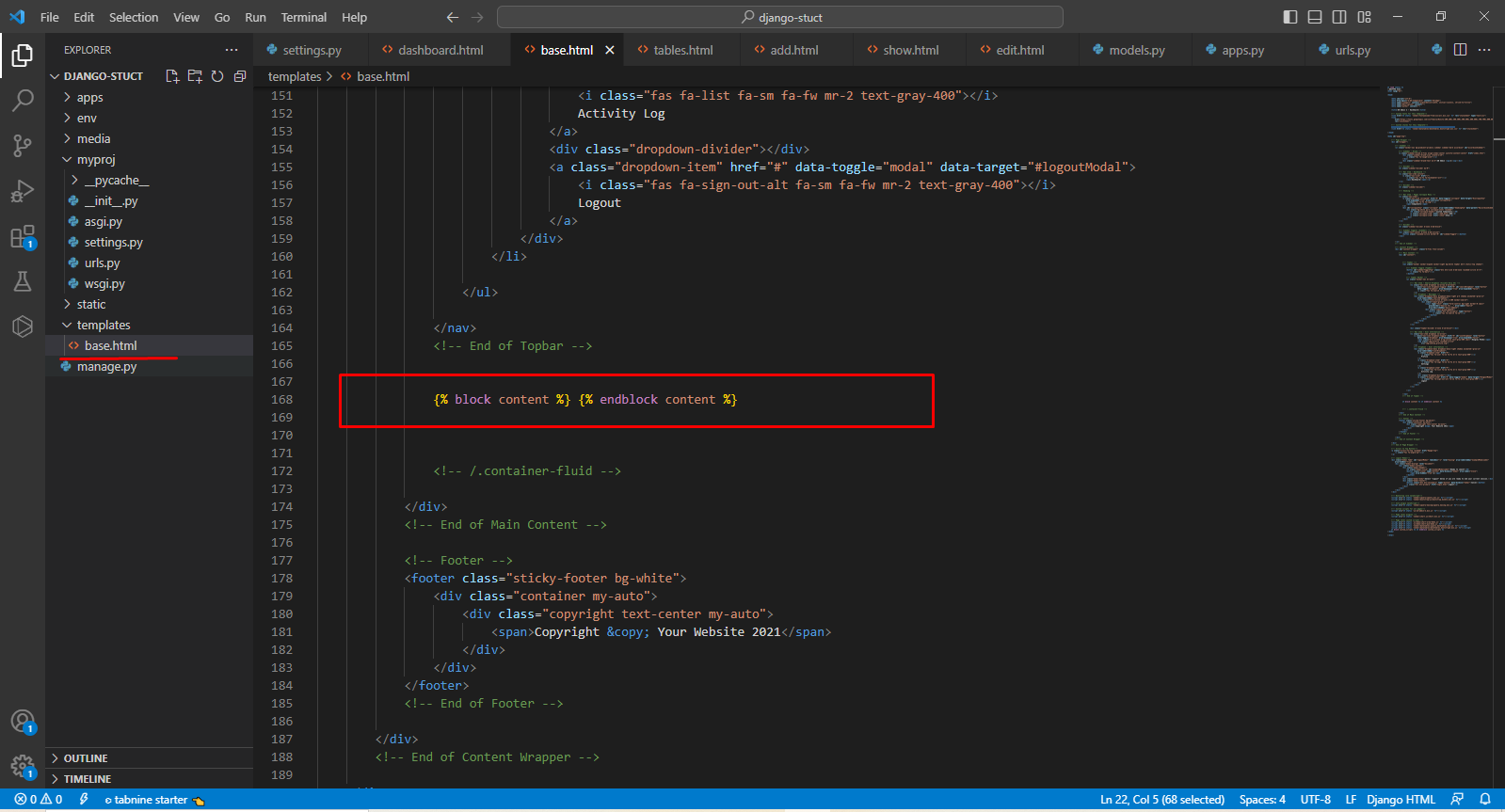
Templates Extending

- You can also extend the template using the **extends** command. This will help your template to be DRY.

Let’s take an example. Suppose you are extending **show.html** with **base.html**

* Edit your base.html file to add following commend

{% block content %} {% endblock content %}



* Edit your show.html file to add following command to extends the base file

{% extends 'base.html' %}

{% block content %}

<!--Extended Content -->

{% endblock content %}

