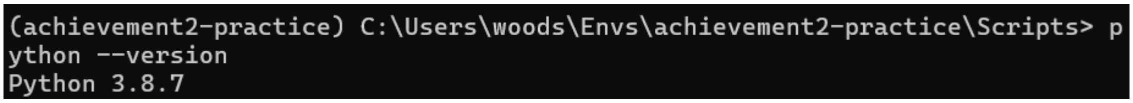
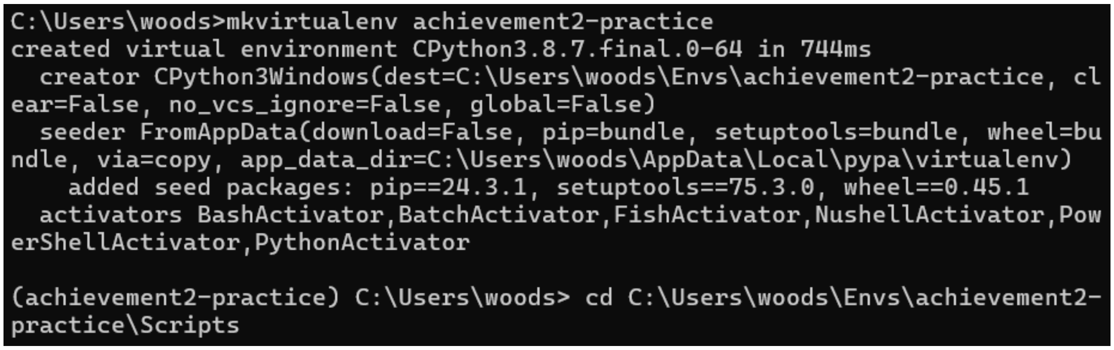
1. Why is Django so popular among web developers?
    - Django is popular among web developers because of its many benefits. Django is a batteries-included web development framework that promotes fast processing and security. It is also open source and has a huge community of contributors, which means it’s usually easy to get support if it’s needed.
2. 5 large companies that use Django, what they use Django for, and what the company’s product or service is.
    - Spotify is an audio streaming and media service that uses Django to ensure optimal functionality and manage backend systems, like user profiles and content recommendations.
    - Pinterest is a social media service for publishing and discovery of information that uses Django to manage backend systems, like user profiles and content management.
    - YouTube is a social media and online video sharing platform that uses Django to ensure features and upgrades are implemented as fast as possible.
    - Instagram is a photo and video sharing social networking service that uses Django for handling a large amounts of user interactions and to manage backend systems, like user authentication and content management.
    - The Washington Post is an American daily newspaper that uses Django to manage backend systems, like content management.
3. Explain if you would use Django in each of the following scenarios, including why or why not.
    - You need to develop a web application with multiple users
        - I would use Django in this scenario because it has built-in support for user authentication, permissions, and session management that make it ideal for a web application with multiple users.
    - You need fast deployment and the ability to make changes as you proceed
        - I would use Django in this scenario because it’s built and structured using DRY principles and is considered a “batteries-included” framework, both of which help ensure development is fast and easy.
    - You need to build a very basic application, which doesn't require any database access or file operations
        - I would not use Django in this scenario because it’s generally server intensive and has many built-in features that are for database access and file operations, so using it would be excessive and unnecessary.

- You want to build an application from scratch and want a lot of control over how it works
    - I would not use Django in this scenario because it uses prewritten code, which results in a loss of control over the internals of your system.
- You're about to start working on a big project and are afraid of getting stuck and needing additional support
    - I would use Django in this scenario because it is open source and has a huge community of contributors which means its usually easy to get support if you need it.

4. Python version screenshot
    - 

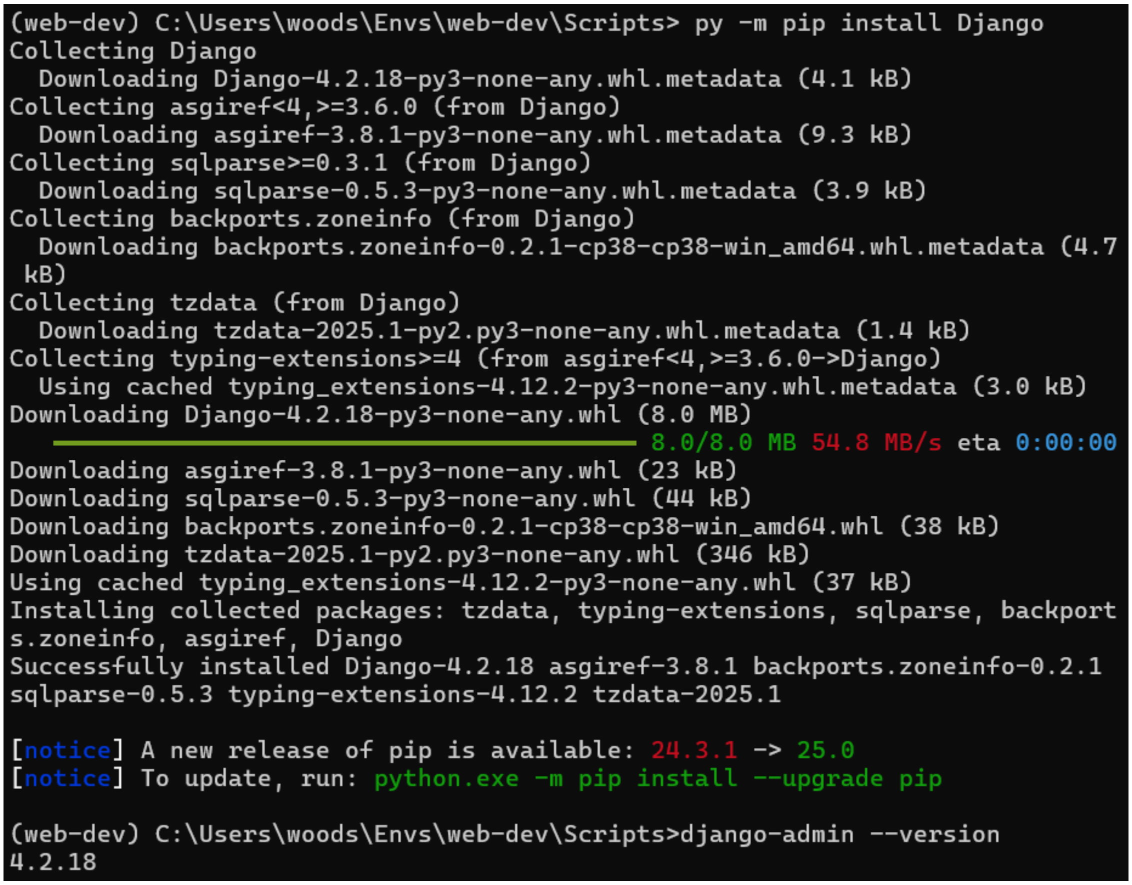
```
(achievement2-practice) C:\Users\woods\Envs\achievement2-practice\Scripts> p
ython --version
Python 3.8.7
```

5. Activated new virtual environment: achievement2-practice
    - 

```
C:\Users\woods>mkvirtualenv achievement2-practice
created virtual environment CPython3.8.7.final.0-64 in 744ms
  creator CPython3Windows(dest=C:\Users\woods\Envs\achievement2-practice, cl
ear=False, no_vcs_ignore=False, global=False)
  seeder FromAppData(download=False, pip=bundle, setuptools=bundle, wheel=bu
ndle, via=copy, app_data_dir=C:\Users\woods\AppData\Local\pypa\virtualenv)
    added seed packages: pip==24.3.1, setuptools==75.3.0, wheel==0.45.1
  activators BashActivator,BatchActivator,FishActivator,NushellActivator,Pow
erShellActivator,PythonActivator

(achievement2-practice) C:\Users\woods> cd C:\Users\woods\Envs\achievement2-
practice\Scripts
```

6. Django installation and version screenshot

- 

```
(web-dev) C:\Users\woods\Envs\web-dev\Scripts> py -m pip install Django
Collecting Django
  Downloading Django-4.2.18-py3-none-any.whl.metadata (4.1 kB)
Collecting asgiref<4,>=3.6.0 (from Django)
  Downloading asgiref-3.8.1-py3-none-any.whl.metadata (9.3 kB)
Collecting sqlparse>=0.3.1 (from Django)
  Downloading sqlparse-0.5.3-py3-none-any.whl.metadata (3.9 kB)
Collecting backports.zoneinfo (from Django)
  Downloading backports.zoneinfo-0.2.1-cp38-cp38-win_amd64.whl.metadata (4.7
 kB)
Collecting tzdata (from Django)
  Downloading tzdata-2025.1-py2.py3-none-any.whl.metadata (1.4 kB)
Collecting typing-extensions>=4 (from asgiref<4,>=3.6.0->Django)
  Using cached typing_extensions-4.12.2-py3-none-any.whl.metadata (3.0 kB)
Downloading Django-4.2.18-py3-none-any.whl (8.0 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 8.0/8.0 MB 54.8 MB/s eta 0:00:00
Downloading asgiref-3.8.1-py3-none-any.whl (23 kB)
Downloading sqlparse-0.5.3-py3-none-any.whl (44 kB)
Downloading backports.zoneinfo-0.2.1-cp38-cp38-win_amd64.whl (38 kB)
Downloading tzdata-2025.1-py2.py3-none-any.whl (346 kB)
Using cached typing_extensions-4.12.2-py3-none-any.whl (37 kB)
Installing collected packages: tzdata, typing-extensions, sqlparse, backport
s.zoneinfo, asgiref, Django
Successfully installed Django-4.2.18 asgiref-3.8.1 backports.zoneinfo-0.2.1
sqlparse-0.5.3 typing-extensions-4.12.2 tzdata-2025.1

[notice] A new release of pip is available: 24.3.1 -> 25.0
[notice] To update, run: python.exe -m pip install --upgrade pip

(web-dev) C:\Users\woods\Envs\web-dev\Scripts>django-admin --version
4.2.18
```