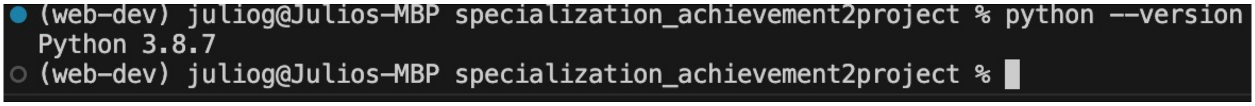
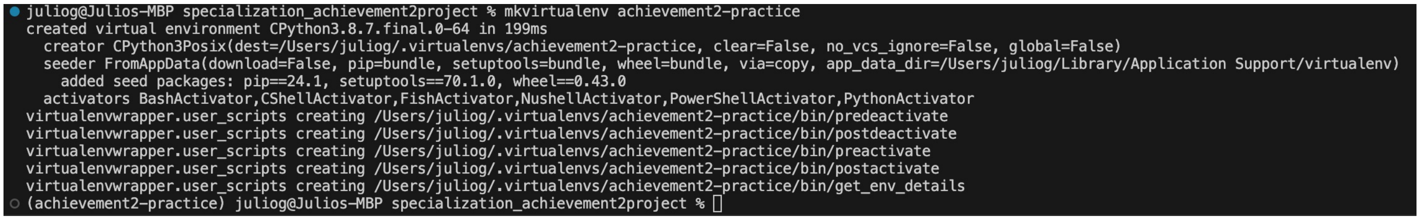
Achievement 2
Exercise 2.1

1. Write two to three sentences on why Django is so popular among web developers.
    a. Django is popular because it is a batteries-included framework with most of what's needed for effective development. It's Model View Template architecture allows Django to handle network transmission and content delivery at a high speed. It helps to optimize code structure and reduce repetitive code following DRY principles.
2. After some research, list five large companies that use Django. Specify what the company's product or service is and what they use Django for.
    a. Instagram: a popular social network, uses Django for its web version to handle large amounts of data and user interactions.
    b. Spotify: an audio streaming platform, uses Django for a fast backend and for machine learning options.
    c. YouTube: a video hosting platform, uses Django due to the rapid growth of the audience and the need for fast implementation of new features.
    d. The Washington Post: a news platform, uses Django because of the scalability and to handle large amounts of data generated daily by their audience.
    e. Pinterest: a visual discovery engine, uses Django much like Instagram in that they're able to handle large amounts of data and lots of user interations.
    f. Source: https://intellisoft.io/15-famous-sites-built-with-django-that-you-probably-know/
3. For each of the following scenarios, explain if you would use Django (and why or why not):
    - You need to develop a web application with multiple users.
        - Yes, Django can handle multiple users and sessions at once. It includes authentication and authorization.
    - You need fast deployment and the ability to make changes as you proceed.
        - Yes, it streamlines development helping to get applications off the ground faster and allows for changes to happen quickly.
    - You need to build a very basic application, which doesn't require any database access or file operations.
        - Django would be too feature-rich for this type of application, and another framework would be better suited.
    - You want to build an application from scratch and want a lot of control over how it works.
        - Django has a very rigid structure and rules that must be followed. It is very difficult to have control over how the application gets put together.
    - You're about to start working on a big project and are afraid of getting stuck and needing additional support.
        - Because it is open source, Django has a large and active community, so there are others who may be able to help.
4. Download and install Python (if you haven't done so already).



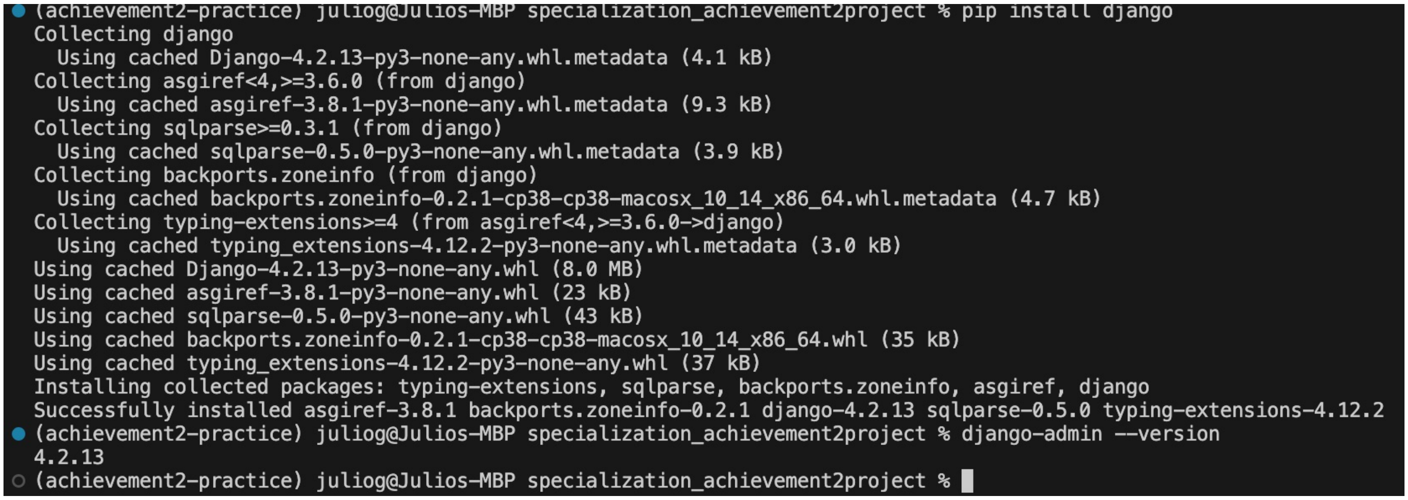
```
(web-dev) juliog@Julios-MBP specialization_achievement2project % python --version
Python 3.8.7
(web-dev) juliog@Julios-MBP specialization_achievement2project %
```

5. If you haven't done so already, set up and create a virtual environment and name it achievement2-practice.



```
juliog@Julios-MBP specialization_achievement2project % mkvirtualenv achievement2-practice
created virtual environment CPython3.8.7.final.0-64 in 199ms
  creator CPython3Posix(dest=/Users/juliog/.virtualenvs/achievement2-practice, clear=False, no_vcs_ignore=False, global=False)
  seeder FromAppData(download=False, pip=bundle, setuptools=bundle, wheel=bundle, via=copy, app_data_dir=/Users/juliog/Library/Application Support/virtualenv)
    added seed packages: pip==24.1, setuptools==70.1.0, wheel==0.43.0
  activators BashActivator,CShellActivator,FishActivator,NushellActivator,PowerShellActivator,PythonActivator
virtualenvwrapper.user_scripts creating /Users/juliog/.virtualenvs/achievement2-practice/bin/predeactivate
virtualenvwrapper.user_scripts creating /Users/juliog/.virtualenvs/achievement2-practice/bin/postdeactivate
virtualenvwrapper.user_scripts creating /Users/juliog/.virtualenvs/achievement2-practice/bin/preactivate
virtualenvwrapper.user_scripts creating /Users/juliog/.virtualenvs/achievement2-practice/bin/postactivate
virtualenvwrapper.user_scripts creating /Users/juliog/.virtualenvs/achievement2-practice/bin/get_env_details
(achievement2-practice) juliog@Julios-MBP specialization_achievement2project %
```

6. Install Django and verify the installation by checking the version.



```
(achievement2-practice) juliog@Julios-MBP specialization_achievement2project % pip install django
Collecting django
  Using cached Django-4.2.13-py3-none-any.whl.metadata (4.1 kB)
Collecting asgiref<4,>=3.6.0 (from django)
  Using cached asgiref-3.8.1-py3-none-any.whl.metadata (9.3 kB)
Collecting sqlparse>=0.3.1 (from django)
  Using cached sqlparse-0.5.0-py3-none-any.whl.metadata (3.9 kB)
Collecting backports.zoneinfo (from django)
  Using cached backports.zoneinfo-0.2.1-cp38-cp38-macosx_10_14_x86_64.whl.metadata (4.7 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)
Using cached Django-4.2.13-py3-none-any.whl (8.0 MB)
Using cached asgiref-3.8.1-py3-none-any.whl (23 kB)
Using cached sqlparse-0.5.0-py3-none-any.whl (43 kB)
Using cached backports.zoneinfo-0.2.1-cp38-cp38-macosx_10_14_x86_64.whl (35 kB)
Using cached typing_extensions-4.12.2-py3-none-any.whl (37 kB)
Installing collected packages: typing-extensions, sqlparse, backports.zoneinfo, asgiref, django
Successfully installed asgiref-3.8.1 backports.zoneinfo-0.2.1 django-4.2.13 sqlparse-0.5.0 typing-extensions-4.12.2
(achievement2-practice) juliog@Julios-MBP specialization_achievement2project % django-admin --version
4.2.13
(achievement2-practice) juliog@Julios-MBP specialization_achievement2project %
```