Exercise 2.1 Task

Django is popular among developers for more than a few reasons. A few that stand out to me are that when installed it already has most of the essentials that you need to effectively develop your application. Another reason is that Django is an open source, which allows you to easily get support when you need. Lastly the scalability and security that Django helps provide makes it easy for developers to work with.

* List 5 large companies that use Django and specify the products/service is and what they use Django for.

- YouTube: Django helps them add new features and upgrades to current features are done as fast as possible.

- Spotify: Uses Django framework to make their app work by combining its backend that is developed using Python.

- Dropbox: Django framework is used to allow their users to store large amounts of files as well as share them to multiple users.

- NASA: Uses the Django framework to make sure that all of the sites run efficiently so that information can be shared.

- Instagram: Django helps them control the large amount of users and data. With people uploading stuff constantly Djangos framework and Fast Processing allows the high speed sharing aspect of Instagram.

* For each of the following, would you use Django or not? (Explain)

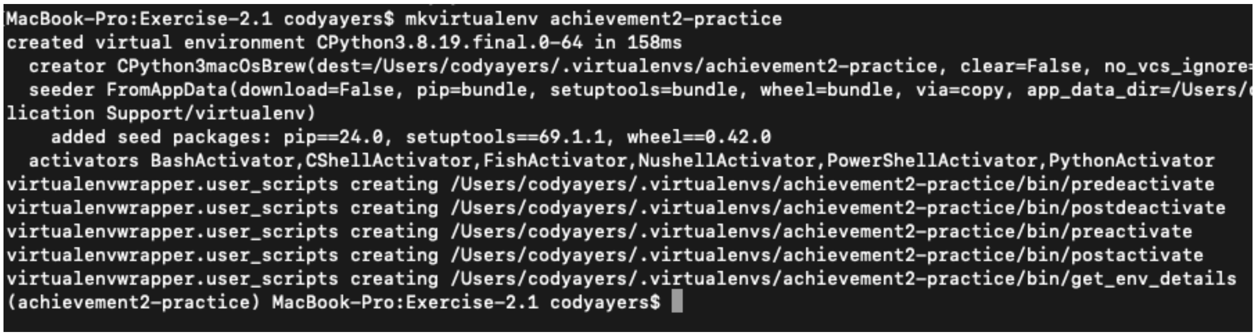
* You need to develop a web application with multiple users:
    * - I would use Django. With multiple users you would need a large database and be able to secure information.

* You need fast deployment and the ability to make changes as you proceed:
    * - Yes I would use Django. Because Django uses MVT it allows for a fast and easy development for your application.

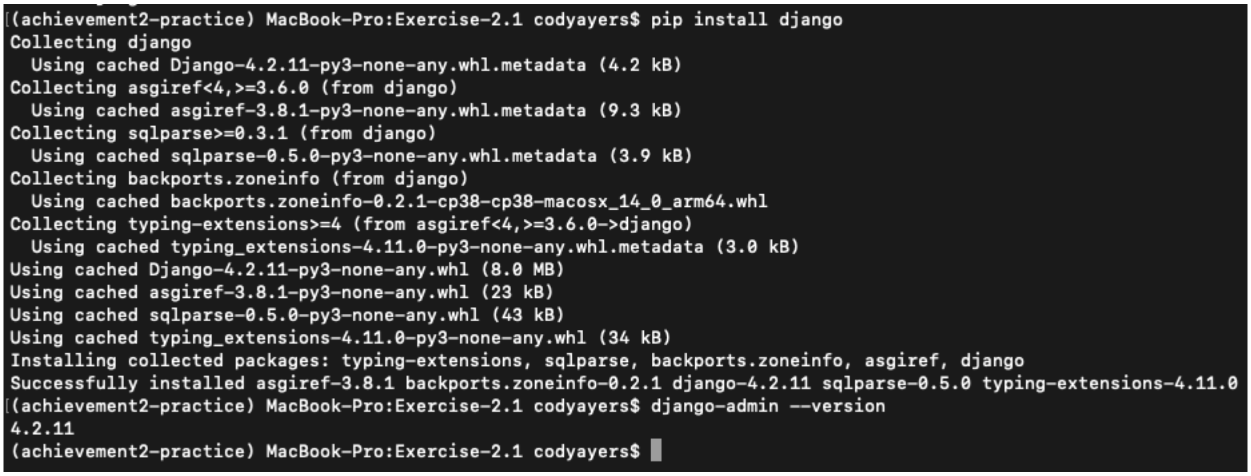
* You need to build a very basic application, which doesn't require any database access or file operations:
    * - No I would not use Django. This would slow down your application as it would have multiple things that you wouldn't need for such a simple application.

* You want to build an application from scratch and want a lot of control over how it works:
    * - If you want control over how things work in your application then Django would not be what I would use. With Django you would have to follow its rules and do it the "The Django Way".

* You're about to start working on a big project and are afraid of getting stuck and needing additional support:
    * - I would use Django for this project because it is an open source framework that has a large community that would allow you to get support when you need it.



```
MacBook-Pro:Exercise-2.1 codyayers$ mkvirtualenv achievement2-practice
created virtual environment CPython3.8.19.final.0-64 in 158ms
  creator CPython3macOsBrew(dest=/Users/codyayers/.virtualenvs/achievement2-practice, clear=False, no_vcs_ignore=
  seeder FromAppData(download=False, pip=bundle, setuptools=bundle, wheel=bundle, via=copy, app_data_dir=/Users/
lication Support/virtualenv)
    added seed packages: pip==24.0, setuptools==69.1.1, wheel==0.42.0
  activators BashActivator,CShellActivator,FishActivator,NushellActivator,PowerShellActivator,PythonActivator
virtualenvwrapper.user_scripts creating /Users/codyayers/.virtualenvs/achievement2-practice/bin/predeactivate
virtualenvwrapper.user_scripts creating /Users/codyayers/.virtualenvs/achievement2-practice/bin/postdeactivate
virtualenvwrapper.user_scripts creating /Users/codyayers/.virtualenvs/achievement2-practice/bin/preactivate
virtualenvwrapper.user_scripts creating /Users/codyayers/.virtualenvs/achievement2-practice/bin/postactivate
virtualenvwrapper.user_scripts creating /Users/codyayers/.virtualenvs/achievement2-practice/bin/get_env_details
(achievement2-practice) MacBook-Pro:Exercise-2.1 codyayers$ 
```



```
[(achievement2-practice) MacBook-Pro:Exercise-2.1 codyayers$ pip install django
Collecting django
  Using cached Django-4.2.11-py3-none-any.whl.metadata (4.2 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_14_0_arm64.whl
Collecting typing-extensions>=4 (from asgiref<4,>=3.6.0->django)
  Using cached typing_extensions-4.11.0-py3-none-any.whl.metadata (3.0 kB)
Using cached Django-4.2.11-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 typing_extensions-4.11.0-py3-none-any.whl (34 kB)
Installing collected packages: typing-extensions, sqlparse, backports.zoneinfo, asgiref, django
Successfully installed asgiref-3.8.1 backports.zoneinfo-0.2.1 django-4.2.11 sqlparse-0.5.0 typing-extensions-4.11.0
[(achievement2-practice) MacBook-Pro:Exercise-2.1 codyayers$ django-admin --version
4.2.11
(achievement2-practice) MacBook-Pro:Exercise-2.1 codyayers$ 
```