

Achievement #2- Django and Web Development

Sapna Bolikal

Exercise 2.1

Reasons Django is so popular among web developers:

Django has a batteries included framework, which means it comes with many tools out of the box, making it easier to use. It allows for rapid development of projects because of the tools it has. Applications run faster on Django as well. Additionally it follows DRY, don't repeat yourself principles so things are straightforward and not repetitive. It is highly scalable, highly secure and has a large community of support.

5 large companies that use Django:

Spotify is the the global music streaming service and it uses Django to support its backend, specifically for web applications and APIs.

Mozilla, which develops Firefox web browser, uses Django for various internal tools and some of their public-facing websites.

Dropbox is a file hosting and collaboration service, and it has used Django for its web applications, helping to manage millions of users and their data.

Instagram is one of the world's leading social media platforms, and it uses Django to handle its massive user base and the high volume of data and interactions.

Pinterest is a popular visual discovery engine, uses Django to manage its extensive content and user interactions.

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- I would say that you could use Django, because it supports large scale web application including multiple users.
- You need fast deployment and the ability to make changes as you proceed- I believe Django would be good at this, as it supports fast deployment.
- You need to build a very basic application, which doesn't require any database access or file operations- Django would not be good for this, as it is used for applications that are larger and use a database.
- You want to build an application from scratch and want a lot of control over how it works- This would not be good for Django, as a lot of the control is not by the developer.
- You're about to start working on a big project and are afraid of getting stuck and needing additional support- Django would work well here, as it is good for big projects and has a large community that could support developers who need help along the way.

Python Version

4.2.15

```
[(web-dev) sapnabolikal@MacBook-Pro ~ % python --version  
Python 3.8.19
```

Virtual Environment and Activation

```
sapnabolikal@MacBook-Pro python % mkvirtualenv achievement2-practice  
created virtual environment CPython3.8.19.final.0-64 in 193ms  
creator CPython3macOsBrew(dest=/Users/sapnabolikal/.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/sapnabolikal/Libr  
ary/Application Support/virtualenv)  
added seed packages: pip==24.1.2, setuptools==71.0.4, wheel==0.43.0  
activators BashActivator,CShellActivator,FishActivator,NushellActivator,PowerShellActivator,PythonActivator  
virtualenvwrapper.user_scripts creating /Users/sapnabolikal/.virtualenvs/achievement2-practice/bin/predeactivate  
virtualenvwrapper.user_scripts creating /Users/sapnabolikal/.virtualenvs/achievement2-practice/bin/postdeactivate  
virtualenvwrapper.user_scripts creating /Users/sapnabolikal/.virtualenvs/achievement2-practice/bin/preactivate  
virtualenvwrapper.user_scripts creating /Users/sapnabolikal/.virtualenvs/achievement2-practice/bin/postactivate  
virtualenvwrapper.user_scripts creating /Users/sapnabolikal/.virtualenvs/achievement2-practice/bin/get_env_details  
(achievement2-practice) sapnabolikal@MacBook-Pro python %
```

```
virtualenvwrapper.user_scripts creating /Users/sapnabolikal/.Virtualenvs/achievement2-prac  
(achievement2-practice) sapnabolikal@MacBook-Pro python % workon achievement2-practice  
(achievement2-practice) sapnabolikal@MacBook-Pro python %
```

Install Django

```
(achievement2-practice) sapnabolikal@MacBook-Pro python % pip install django  
Collecting django  
Using cached Django-4.2.15-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-0.5.1-py3-none-any.whl.metadata (3.9 kB)  
Collecting backports.zoneinfo (from django)  
Using cached backports.zoneinfo-0.2.1-cp38-cp38-macosx_13_0_arm64.whl  
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-4.2.15-py3-none-any.whl (8.0 MB)  
Using cached asgiref-3.8.1-py3-none-any.whl (23 kB)  
Using cached sqlparse-0.5.1-py3-none-any.whl (44 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-4.2.15 sqlparse-0.5.1 typing-extensions-4.12.2
```

Django version

```
(achievement2-practice) sapnabolikal@MacBook-Pro python % django-admin --version  
4.2.15
```