Django is so popular among web developers because it comes with ‘Batteries included’: with a wide range of tools, libraries and features ready to handle common tasks on any web application. It is free and open source, has a thriving and active community and has great documentation.

Five large companies that use Django:

1. **Instagram**

Python’s practicality and simplicity make the language a top-tier choice for Instagram. But after the platform scaled, efficiency became a bigger priority. Using Django, Instagram Engineering was able to build custom tools to meet this goal.

2. **National Geographic**

Django eases the development of data-driven, complex websites like National Geographic’s website. Django also has its own content management system (CMS) called django CMS which the National Geographic's Education page relies on.

3. **Mozilla**

With Django, Firefox can tackle large amounts of traffic and API hits in a more efficient manner.

4. **Spotify**.

Spotify is an audio streaming provider. Users listen to digital music (and sometimes podcasts) around the world through the internet. Python is used for Spotify’s back-end services and data analysis. And Spotify uses a Django app or two to increase functionality.

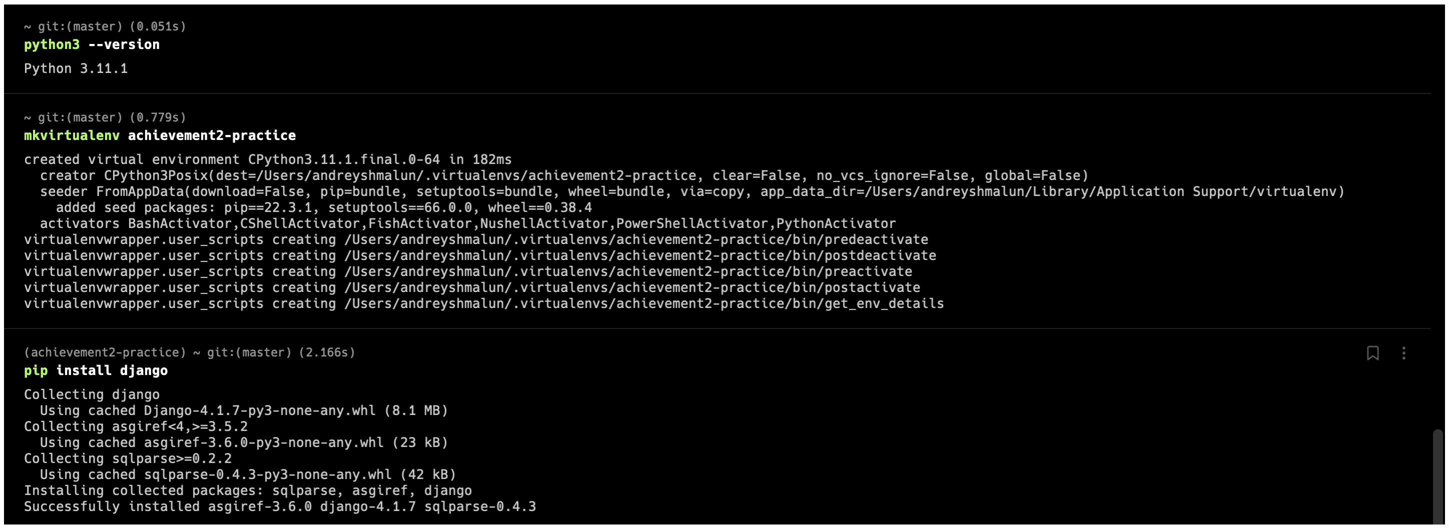
5. **Pinterest**

Pinterest is another social media platform like Instagram. But although you can share images just the same, Pinterest’s primary concern is that users gain inspiration for topics related to fashion, home, cooking, etc. The platform’s user-friendly interface is to blame for much of its appeal. Django’s open-source capacity means Pinterest can modify the framework for its needs.

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.
- You need fast deployment and the ability to make changes as you proceed.
- You need to build a basic application that doesn’t require any database access or file operations.
- You want to build an application from scratch and want a lot of control over how it works.
- You’re about to start working on a big project and are afraid of getting stuck and needing additional support.

Django is a great choice for the first scenario since it allows for multiple user accounts and provides a secure framework for building web applications. Django is also a great choice for the second scenario since it has fast deployment and makes it easy to make changes as you proceed. For the third scenario, Django is not the best choice since it requires a database and file operations, which are not needed for a very basic application. For the fourth scenario, Django is a great choice since it allows for a lot of control over how the application is built and works. For the fifth scenario, Django is a great choice since it has a large community of developers and provides a lot of support if you ever get stuck.



```
~ git:(master) (0.051s)
python3 --version
Python 3.11.1

~ git:(master) (0.779s)
mkvirtualenv achievement2-practice
created virtual environment CPython3.11.1.final.0-64 in 182ms
  creator CPython3Posix(dest=/Users/andreyshmalun/.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/andreyshmalun/Library/Application Support/virtualenv)
    added seed packages: pip==22.3.1, setuptools==66.0.0, wheel==0.38.4
  activators BashActivator,CShellActivator,FishActivator,NushellActivator,PowerShellActivator,PythonActivator
virtualenvwrapper.user_scripts creating /Users/andreyshmalun/.virtualenvs/achievement2-practice/bin/predeactivate
virtualenvwrapper.user_scripts creating /Users/andreyshmalun/.virtualenvs/achievement2-practice/bin/postdeactivate
virtualenvwrapper.user_scripts creating /Users/andreyshmalun/.virtualenvs/achievement2-practice/bin/preactivate
virtualenvwrapper.user_scripts creating /Users/andreyshmalun/.virtualenvs/achievement2-practice/bin/postactivate
virtualenvwrapper.user_scripts creating /Users/andreyshmalun/.virtualenvs/achievement2-practice/bin/get_env_details

(achievement2-practice) ~ git:(master) (2.166s)
pip install django
Collecting django
  Using cached Django-4.1.7-py3-none-any.whl (8.1 MB)
Collecting asgiref<4,>=3.5.2
  Using cached asgiref-3.6.0-py3-none-any.whl (23 kB)
Collecting sqlparse>=0.2.2
  Using cached sqlparse-0.4.3-py3-none-any.whl (42 kB)
Installing collected packages: sqlparse, asgiref, django
Successfully installed asgiref-3.6.0 django-4.1.7 sqlparse-0.4.3
```