

1. Write two to three sentences on why Django is so popular among web developers.
 - Django is highly popular among web developers due to its "batteries-included" approach, which provides built-in solutions for many common web development tasks. This framework promotes rapid development and clean, pragmatic design. Additionally, its robust documentation and large community support make it an appealing choice for developers of all skill levels.
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.
 1. **Instagram:** Uses Django due to its efficiency and scalability, which are crucial for handling the massive volume of user interactions on the platform.
 2. **Pinterest:** Uses Django to manage its large user base and data-intensive operations, benefiting from Django's capabilities in handling high traffic and data volumes.
 3. **Mozilla:** Mozilla utilizes Django to manage high volumes of user queries and interactions, leveraging Django's robustness and reliability.
 4. **Spotify:** Uses Django to optimize backend implementations and leverage machine learning capabilities, aiding in managing the service's large scale and complex functionalities.
 5. **Eventbrite:** This event management and ticketing platform chose Django to help manage its extensive user interactions and event handling functionalities. Django supports Eventbrite with features like URL routing and form building.
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 is designed to handle user authentication, authorization, and management out of the box. It can efficiently manage user accounts, groups, permissions, and cookie-based user sessions.
 - You need fast deployment and the ability to make changes as you proceed.
 - Yes, Django is suitable. Its design encourages rapid development and the ability to make changes quickly, which is facilitated by Django's automatic admin interface and its migrations system to adjust database schemas.

- You need to build a very basic application, which doesn't require any database access or file operations.
 - No, Django might not be the best choice. Django is a bit heavy for such simple needs due to its comprehensive structure.
- You want to build an application from scratch and want a lot of control over how it works.
 - Maybe, depending on the specifics. Django is very structured, which can sometimes limit deep customization without extensive work. For projects needing more control from the ground up, frameworks like Flask or even using a language like Go might offer more flexibility.
- You're about to start working on a big project and are afraid of getting stuck and needing additional support.
 - Yes, Django is ideal. Its large and active community, along with extensive documentation and third-party packages, ensures you have ample support and resources to tackle big projects.

Python Version

```
munkhjargalbatsaikhan@Munkhjargals-MacBook-Pro ~ % python3 --version
Python 3.12.3
munkhjargalbatsaikhan@Munkhjargals-MacBook-Pro ~ %
```

Set up virtual environment

```
munkhjargalbatsaikhan@Munkhjargals-MacBook-Pro GitHub % cd achievement2-practice
munkhjargalbatsaikhan@Munkhjargals-MacBook-Pro achievement2-practice % mkvirtualenv achievement2-practice
created virtual environment CPython3.12.3.final.0-64 in 96ms
  creator CPython3Posix(dest=/Users/munkhjargalbatsaikhan/.virtualenvs/achievement2-practice, clear=False, no_vcs_ignore=False, global=False)
  seeder FromAppData(download=False, pip=bundle, via=copy, app_data_dir=/Users/munkhjargalbatsaikhan/Library/Application Support/virtualenv)
    added seed packages: pip==24.0
  activators BashActivator,CShellActivator,FishActivator,NushellActivator,PowerShellActivator,PythonActivator
virtualenvwrapper.user_scripts creating /Users/munkhjargalbatsaikhan/.virtualenvs/achievement2-practice/bin/predeactivate
virtualenvwrapper.user_scripts creating /Users/munkhjargalbatsaikhan/.virtualenvs/achievement2-practice/bin/postdeactivate
virtualenvwrapper.user_scripts creating /Users/munkhjargalbatsaikhan/.virtualenvs/achievement2-practice/bin/preactivate
virtualenvwrapper.user_scripts creating /Users/munkhjargalbatsaikhan/.virtualenvs/achievement2-practice/bin/postactivate
virtualenvwrapper.user_scripts creating /Users/munkhjargalbatsaikhan/.virtualenvs/achievement2-practice/bin/get_env_details
(achievement2-practice) munkhjargalbatsaikhan@Munkhjargals-MacBook-Pro achievement2-practice %
```

Install Django

```
(achievement2-practice) munkhjargalbatsaikhan@Munkhjargals-MacBook-Pro achievement2-practice % pip3 install django
Collecting django
  Using cached Django-5.0.6-py3-none-any.whl.metadata (4.1 kB)
Collecting asgiref<4,>=3.7.0 (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)
Using cached Django-5.0.6-py3-none-any.whl (8.2 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)
Installing collected packages: sqlparse, asgiref, django
Successfully installed asgiref-3.8.1 django-5.0.6 sqlparse-0.5.0
(achievement2-practice) munkhjargalbatsaikhan@Munkhjargals-MacBook-Pro achievement2-practice % django-admin --version
5.0.6
(achievement2-practice) munkhjargalbatsaikhan@Munkhjargals-MacBook-Pro achievement2-practice %
```