Write two to three sentences on why Django is so popular among web developers.

Python is easy to read, has a lot of powerful computational features and its MVT architecture ensures development is fast and easy. Django Follows DRY (Don't Repeat Yourself) Principles which helps you keep your code non-repetitive, non-redundant, and efficient. Since Django is open source, it has a huge community of contributors. As such, it's usually very easy to get support if you need it.

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.

Youtube-Django plays a key role in ensuring that new features are added and upgrades are implemented in the shortest time possible. Moreover, it eliminates mistakes and allows the YouTube team of developers to perform flawlessly.

Instagram-With 1 billion monthly active users, this app has incredibly large amounts of data and user interactions to process. The Django framework helps Instagram perform all these tasks seamlessly.

Spotify- Spotify uses Python for its backend services and development and combines it with the Django framework to ensure optimum functionality of the app.

Pinterest- Pinterest utilizes the Django framework to execute multiple tasks, scale, and maintain top-notch performance.

Udemy-Since Udemy has a lot of video content, it's necessary to have a framework that can keep up with its database demands. Django's architecture allows for this.

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.
Django would be good to use as it would help deploy the application quickly and scale it up in the future to improve the user's experience.

-You need fast deployment and the ability to make changes as you proceed.
I would use it as one of the main features of Django is its fast deployment speed and easy flawless scalability.

-You need to build a very basic application, which doesn't require any database access or file operations.
I don't think Django would be a good choice to use in this case as Django is better when most of the features it offers are used.

-You want to build an application from scratch and want a lot of control over how it works.
Django is a batteries-included system that works in a specific Django way which means you will lose some control over the internals in your system. For that reason I would not use it.

-You're about to start working on a big project and are afraid of getting stuck and needing additional support.
Django would be useful in this case as its built-in libraries and tools would be a big help if I got stuck and being able to scale down with minimal flaws is a plus.

Python Version:

```
C:\Users\Madis>python --version
Python 3.8.7
```

Creating Virtual Environment

```
C:\Users\Madis>mkvirtualenv achievement2-practice
created virtual environment CPython3.8.7.final.0-64 in 7223ms
  creator CPython3Windows(dest=C:\Users\Madis\Envs\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=C:\Users\Madis\
AppData\Local\pypa\virtualenv)
    added seed packages: pip==23.2, setuptools==68.0.0, wheel==0.40.0
  activators BashActivator,BatchActivator,FishActivator,NushellActivator,PowerShellActivator,PythonActivator

(achievement2-practice) C:\Users\Madis>
```

Django Install/Version

```
(achievement2-practice) C:\Users\Madis>django-admin --version
4.2.6
```