

TASK 2.1 Django

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

Django is a framework and uses its own libraries and features without having to install external libraries. Its ability to be scalable, secure and fast allows web developers to concentrate on the project logic and not the web development. It follows the DRY (Don't Repeat Yourself) principle, which helps to keep the code efficient and non-repetitive.

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.
 - YouTube: the Django framework allows the YouTube developers to act quickly & flawlessly to implement new features and to maintain the speed of the website.
 - Instagram: Django helped to scale the application, process huge amounts of data and manage a greater number of interactions between users every single second.
 - Spotify: The huge library of this application is accessible everywhere and on any device. It contains a vast amount of data and to handle this data the application uses Python alongside Django. There were mainly two reasons for choosing this framework the Fast backend and Machine learning options.
 - Mozilla: since moving to Python+Django the benefits to the platform is much more capable to handle hundreds of millions of views per month, and even more API hits per day. Today, all the Mozilla **support site** and the **add-ons** are powered by Django.
 - Dropbox: Dropbox developers choose Django to add the user history option, synchronise an account across different devices, and to add various types of file-sharing options.
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.

I would use Django here, because it can be quickly deployed, is scalable and with security in mind.

- You need fast deployment and the ability to make changes as you proceed.

I would use Django here, so long as it's for a medium to large application or one that may need to be scaled up later. The Django framework can help applications run faster and are scalable.

- You need to build a very basic application, which doesn't require any database access or file operations.

I would not use Django here, because Django is great for large scale systems and databases, however for small systems the preinstalled code is overkill on their server and bandwidth.

- You want to build an application from scratch and want a lot of control over how it works.

I would not use Django in this instance, because Django framework is highly structured and follows its own rules and doesn't allow that much flexibility outside of these.

- You're about to start working on a big project and are afraid of getting stuck and needing additional support.

I would use Django here, because the Django community is large. the documentation is open source and if support is needed it's usually easy to get.

Python Version

```
C:\Windows\System32>py --version
Python 3.8.7
```

Django version and venv achievement2_practice

```
(achievement2-practice) F:\Users\Andrew\Documents\Career Foundry\Work submitted\Python\Python\Exercise_2.1>django-admin
--version
4.2.2
```