

Task 2.1

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

Django is popular because it simplifies web development by offering ready-to-use features, saving developers time and effort. Also, Django's MVT architecture ensures easy scalability by simplifying the addition of new hardware and components. Another benefit of Django is its big community of contributors, so it's easy to get support if needed.

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.

- i. Instagram: Django is used for managing user data, content delivery, and handling interactions between users.
- ii. Pinterest: Django is used for managing user accounts, content organization, and search functionality.
- iii. Spotify: Django is used for user authentication, playlist management, and content delivery.
- iv. YouTube: Django is used for backend services such as user authentication, content management, and analytics.
- v. Dropbox: Django is used for user account management, file synchronization, and access control.

3. For each of the following scenarios, explain if you would use Django (and why or why not):

- a. *You need to develop a web application with multiple users.*

Yes, I would use Django, because it provides built-in authentication features.

- b. *You need fast development and the ability to make changes as you proceed.*

Yes, I would use Django, because it is ideal for fast development and making changes as I proceed due to its MVT architecture, where view and model are just loosely coupled, allowing for relatively independent development and modification of these components.

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

No, I would not use Django because it's not necessary for building a very basic application that doesn't require database access or file operations.

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

No, I would not use Django, because it's a batteries-included system and works the specific Django way, so I won't have a lot of control over how it works.

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

Yes, I would use Django, because Django has a large and active community, so it will be easy to get additional support if I need it.

4. Screenshot Version of Python

```
(web-dev) C:\>python -V
Python 3.8.7

(web-dev) C:\>|
```

5. Screenshot Activated Environment

```
(achievement2-practice) C:\>|
```

6. Screenshot Install Django

```
(achievement2-practice) C:\>py -m pip install Django
Collecting Django
  Using cached Django-4.2.11-py3-none-any.whl.metadata (4.2 kB)
Collecting asgiref<4,>=3.6.0 (from Django)
  Using cached asgiref-3.7.2-py3-none-any.whl.metadata (9.2 kB)
Collecting sqlparse>=0.3.1 (from Django)
  Using cached sqlparse-0.4.4-py3-none-any.whl.metadata (4.0 kB)
Collecting backports.zoneinfo (from Django)
  Using cached backports.zoneinfo-0.2.1-cp38-cp38-win_amd64.whl.metadata (4.7 kB)
Collecting tzdata (from Django)
  Using cached tzdata-2024.1-py2.py3-none-any.whl.metadata (1.4 kB)
Collecting typing_extensions>=4 (from asgiref<4,>=3.6.0->Django)
  Using cached typing_extensions-4.10.0-py3-none-any.whl.metadata (3.0 kB)
Using cached Django-4.2.11-py3-none-any.whl (8.0 MB)
Using cached asgiref-3.7.2-py3-none-any.whl (24 kB)
Using cached sqlparse-0.4.4-py3-none-any.whl (41 kB)
Using cached backports.zoneinfo-0.2.1-cp38-cp38-win_amd64.whl (38 kB)
Using cached tzdata-2024.1-py2.py3-none-any.whl (345 kB)
Using cached typing_extensions-4.10.0-py3-none-any.whl (33 kB)
Installing collected packages: tzdata, typing_extensions, sqlparse, backports.zoneinfo, asgiref, Django
Successfully installed Django-4.2.11 asgiref-3.7.2 backports.zoneinfo-0.2.1 sqlparse-0.4.4 typing_extensions-4.10.0 tzdata-2024.1

(achievement2-practice) C:\>djadmin --version
4.2.11
```