

## Task 2.1

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

Django is popular among web developers for several reasons. First, its "batteries-included" philosophy provides developers with tools and features out of the box, streamlining the development process. Second, Django follows the Don't Repeat Yourself (DRY) principle, promoting code reuse and maintainability. Lastly, its emphasis on security practices and well-structured documentation contributes to its scalability and thriving community.

2. List five large companies that use Django. Specify what the company's product or service is and what they use Django for.

Instagram, a social media platform for sharing photos and videos, uses Django for its backend development. Django helps manage the extensive database and handle user interactions efficiently. Pinterest, a visual discovery and bookmarking platform, relies on Django for its backend infrastructure. Django supports the handling of user-generated content, search functionality, and platform stability. The music streaming service, Spotify, utilizes Django for managing user accounts, playlists, and content delivery. Dropbox, a cloud storage and file synchronization service, employs Django in its backend systems. Django facilitates user authentication, file organization, and collaborative features within the platform. Eventbrite, an online event management and ticketing platform, uses Django to handle event creation, ticket sales, and user interactions.

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 a suitable choice for developing web applications with multiple users. Its built-in authentication system, ORM (Object-Relational Mapping), and user management features simplify the development of these applications.

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

Yes, Django supports fast deployment, and its development framework allows for rapid changes. Features like Django's built-in development server, automatic admin interface, and extensive documentation contribute to this.

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

No, Django isn't good for a very basic application without database access or file operations. Its robust features are geared towards more complex applications. A lightweight framework might be a more suitable choice for these applications.

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

No, Django's "batteries-included" philosophy, while beneficial for rapid development, is not great for having a lot of control over the development. Developers seeking more control and flexibility in choosing components may prefer a more lightweight framework, which allows more for building applications from the ground up.

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

Yes, Django is a good choice for big projects due to its scalability, extensive community support, and a large ecosystem of reusable components. The Django community actively offers assistance and solutions for potential challenges that may arise.

#### 4. Download and install Python.

```
C:\Users\kamar>python -V
Python 3.8.7

C:\Users\kamar>virtualenv --version
virtualenv 20.25.0 from c:\users\kamar\appdata\local\programs\python\python38-32\lib\site-packages\virtualenv\__init__.py
```

#### 5. Set up and create a virtual environment and name it achievement2-practice.

```
(web-dev) C:\Users\kamar>mkvirtualenv achievement2-practice
created virtual environment CPython3.8.7.final.0-32 in 455ms
creator CPython3Windows(dest=C:\Users\kamar\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\kamar\AppData\Local\pypa\virtualenv)
added seed packages: pip==23.3.1, setuptools==69.0.2, wheel==0.42.0
activators BashActivator,BatchActivator,FishActivator,NushellActivator,PowerShellActivator,PythonActivator

(achievement2-practice) C:\Users\kamar>"C:\Users\kamar\Envs\achievement2-practice\Scripts\activate.bat"

(achievement2-practice) C:\Users\kamar>if defined _OLD_VIRTUAL_PYTHONPATH (set "PYTHONPATH=") else (set "_OLD_VIRTUAL_PYTHONPATH=")
```

#### 6. Install Django and verify the installation by checking the version.

```
(achievement2-practice) C:\Users\kamar>py -m pip install Django
Collecting Django
  Using cached Django-4.2.9-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 (41 kB)
Collecting backports.zoneinfo (from Django)
  Using cached backports.zoneinfo-0.2.1-cp38-cp38-win32.whl (36 kB)
Collecting tzdata (from Django)
  Using cached tzdata-2023.4-py2.py3-none-any.whl.metadata (1.4 kB)
Collecting typing-extensions>=4 (from asgiref<4,>=3.6.0->Django)
  Using cached typing_extensions-4.9.0-py3-none-any.whl.metadata (3.0 kB)
Using cached Django-4.2.9-py3-none-any.whl (8.0 MB)
Using cached asgiref-3.7.2-py3-none-any.whl (24 kB)
Using cached tzdata-2023.4-py2.py3-none-any.whl (346 kB)
Using cached typing_extensions-4.9.0-py3-none-any.whl (32 kB)
Installing collected packages: tzdata, typing-extensions, sqlparse, backports.zoneinfo, asgiref, Django
Successfully installed Django-4.2.9 asgiref-3.7.2 backports.zoneinfo-0.2.1 sqlparse-0.4.4 typing-extensions-4.9.0 tzdata-2023.4

[notice] A new release of pip is available: 23.3.1 -> 23.3.2
[notice] To update, run: python.exe -m pip install --upgrade pip

(achievement2-practice) C:\Users\kamar>django-admin --version
4.2.9
```

