Web Development & Django

1. 2.1: Getting Started with Django

Task deliverables

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 a robust framework with built-in features, such as authentication, database management, and URL routing. It emphasizes a clean and pragmatic design, helping developers create scalable and secure web applications quickly. Django's active community and comprehensive documentation also make it a go-to choice for many developers.

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.**

   Instagram uses Django to manage its massive user base and handle rapid development needs. Spotify employs Django for its web app and backend services, especially in handling music data and user interactions. Pinterest relies on Django for scaling and serving millions of users efficiently. Dropbox uses Django for its web-based file storage and sharing features. Lastly, Mozilla incorporates Django in its projects for quick development and maintenance of web applications.

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.**

   Django is ideal for a web application with multiple users because it provides built-in tools like user authentication and session management.

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

   For fast deployment and iterative changes, Django is a good choice as it allows for rapid prototyping and easy updates with its reusable components.

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

   If building a very basic application without database or file operations, Django might be overkill since its features are designed for more complex use cases.

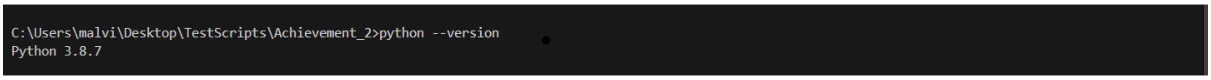
d. **You want to build an application from scratch and want a lot of control over how it works.**

   For building an application from scratch with a lot of control, Django might not be the best fit as its structure enforces certain conventions that limit flexibility.

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

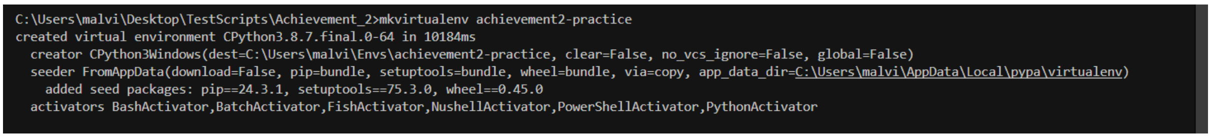
   For large projects where additional support may be needed, Django is excellent due to its strong community, extensive libraries, and well-documented framework.

4. Python version



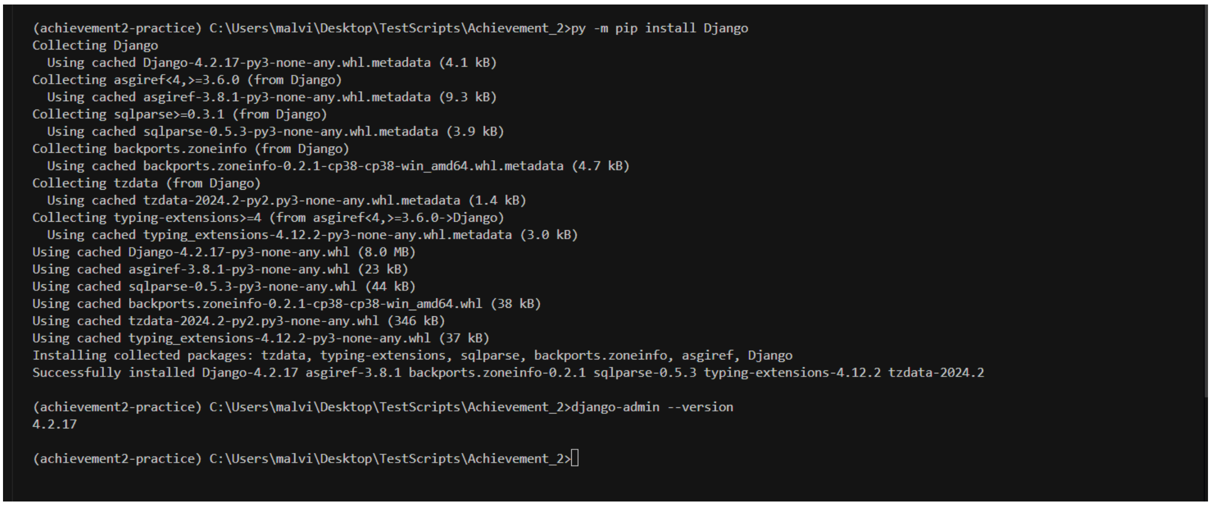
```
C:\Users\malvi\Desktop\TestScripts\Achievement_2>python --version
Python 3.8.7
```

5. Virtual environment



```
C:\Users\malvi\Desktop\TestScripts\Achievement_2>mkvirtualenv achievement2-practice
created virtual environment CPython3.8.7.final.0-64 in 10184ms
  creator CPython3Windows(dest=C:\Users\malvi\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\malvi\AppData\Local\pypa\virtualenv)
    added seed packages: pip==24.3.1, setuptools==75.3.0, wheel==0.45.0
  activators BashActivator,BatchActivator,FishActivator,NushellActivator,PowerShellActivator,PythonActivator
```

6. Django



```
(achievement2-practice) C:\Users\malvi\Desktop\TestScripts\Achievement_2>py -m pip install Django
Collecting Django
  Using cached Django-4.2.17-py3-none-any.whl.metadata (4.1 kB)
Collecting asgiref<4,>=3.6.0 (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-0.5.3-py3-none-any.whl.metadata (3.9 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-2024.2-py2.py3-none-any.whl.metadata (1.4 kB)
Collecting typing-extensions>=4 (from asgiref<4,>=3.6.0->Django)
  Using cached typing_extensions-4.12.2-py3-none-any.whl.metadata (3.0 kB)
Using cached Django-4.2.17-py3-none-any.whl (8.0 MB)
Using cached asgiref-3.8.1-py3-none-any.whl (23 kB)
Using cached sqlparse-0.5.3-py3-none-any.whl (44 kB)
Using cached backports.zoneinfo-0.2.1-cp38-cp38-win_amd64.whl (38 kB)
Using cached tzdata-2024.2-py2.py3-none-any.whl (346 kB)
Using cached typing_extensions-4.12.2-py3-none-any.whl (37 kB)
Installing collected packages: tzdata, typing-extensions, sqlparse, backports.zoneinfo, asgiref, Django
Successfully installed Django-4.2.17 asgiref-3.8.1 backports.zoneinfo-0.2.1 sqlparse-0.5.3 typing-extensions-4.12.2 tzdata-2024.2

(achievement2-practice) C:\Users\malvi\Desktop\TestScripts\Achievement_2>django-admin --version
4.2.17

(achievement2-practice) C:\Users\malvi\Desktop\TestScripts\Achievement_2>
```