Task Questions

2. Django's popularity among web developers stems from its powerful features, such as its built-in admin interface, ORM (Object-Relational Mapping) system, and robust security features, which streamline development and ensure efficient, secure web applications. Additionally, Django's emphasis on simplicity, scalability, and adherence to best practices, coupled with its extensive documentation and vibrant community, make it an attractive choice for building complex web projects with ease.

3.

- Instagram: A social media platform for sharing photos and videos. Django powers its backend infrastructure, including user authentication and content management.
- Pinterest: A visual discovery and social media platform. Django supports various backend functions, including user profiles and content management, enhancing user experience.
- Spotify: A digital music streaming service. Django manages backend systems, such as user accounts and content recommendations, ensuring personalized experiences.
- Dropbox: A cloud storage service. Django facilitates backend operations like user authentication and file management, ensuring secure file storage and sharing.

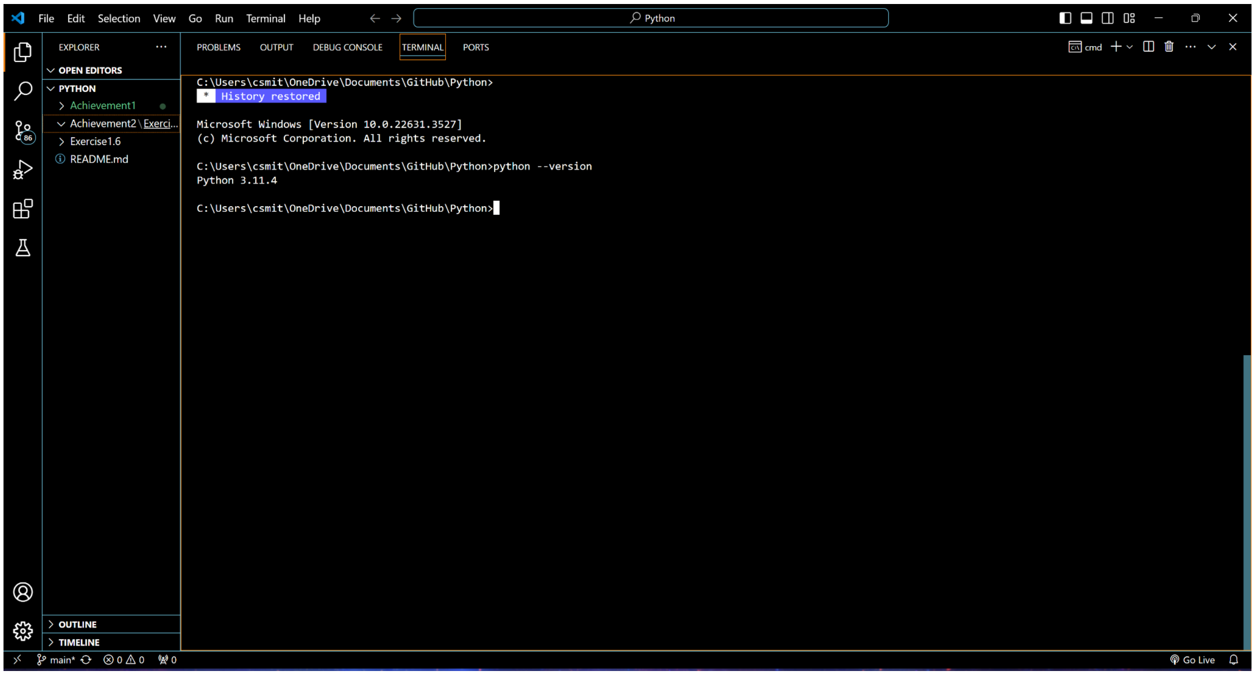
4.

- Developing a web application with multiple users: Yes, Django is an excellent choice for this scenario due to its built-in authentication system, user management features, and scalability. Its robust security measures and support for handling concurrent user sessions make it well-suited for building web applications with multiple users while ensuring their data remains secure.
- Fast deployment and the ability to make changes as you proceed: Absolutely, Django's "batteries-included" approach and emphasis on rapid development through reusable components make it ideal for fast deployment and iterative development. Its modular design allows developers to make changes easily as the project progresses without sacrificing performance, ensuring flexibility and agility throughout the development process.
- Building a very basic application without database access or file operations: No, Django might not be the best choice for this scenario since it's primarily designed for building database-driven web applications. In this case, a lightweight framework like Flask would be more suitable as it provides the necessary flexibility without the overhead of Django's ORM and file handling capabilities.
- Building an application from scratch and wanting a lot of control over how it works: Yes, Django offers developers a high level of control over their applications through its customizable components and flexibility. Its extensive ecosystem of third-party packages and comprehensive documentation empower developers to tailor their applications

precisely to their needs, providing the desired level of control over functionality and performance.

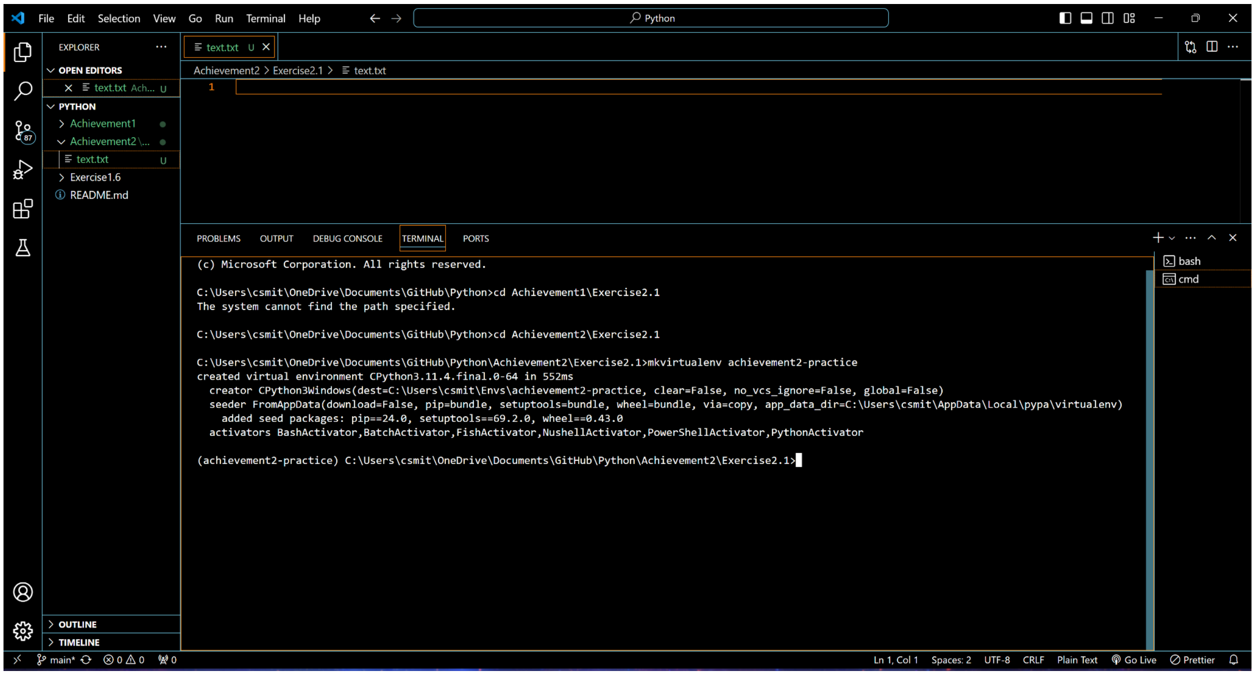
- Starting a big project and being afraid of getting stuck and needing additional support: Yes, Django is an excellent choice for big projects where support might be needed, thanks to its active community, extensive documentation, and vast ecosystem of third-party packages. The availability of experienced developers and resources makes it easier to troubleshoot issues and find solutions, reducing the risk of getting stuck and ensuring the project's success in the long run.

Python Version



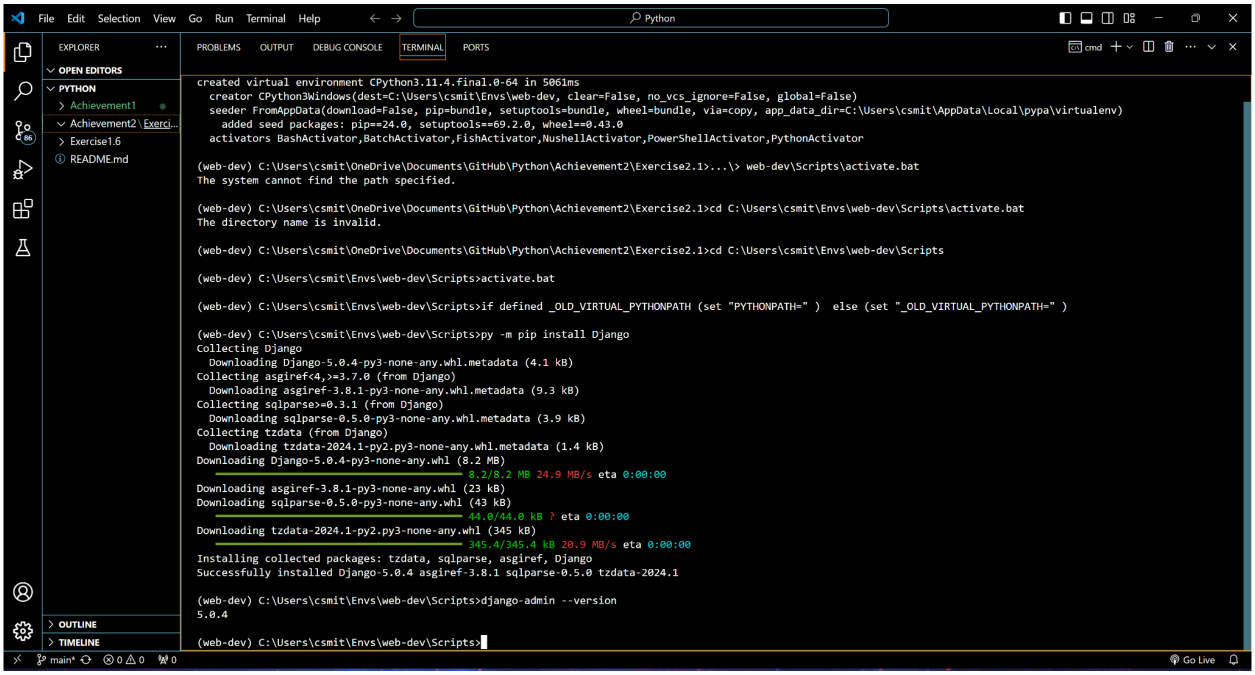
```
C:\Users\csmit\OneDrive\Documents\GitHub\Python>
 *  History restored

Microsoft Windows [Version 10.0.22631.3527]
(c) Microsoft Corporation. All rights reserved.

C:\Users\csmit\OneDrive\Documents\GitHub\Python>python --version
Python 3.11.4

C:\Users\csmit\OneDrive\Documents\GitHub\Python>
```

Virtual Environment:


File Edit Selection View Go Run Terminal Help
Python
EXPLORER
OPEN EDITORS
text.txt Ach... U
PYTHON
Achievement1
Achievement2\...
text.txt U
Exercise1.6
README.md
OUTLINE
TIMELINE
text.txt U
Achievement2 > Exercise2.1 > text.txt
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
bash
cmd
main* 0 0 0
Ln 1, Col 1 Spaces: 2 UTF-8 CRLF Plain Text Go Live Prettier

```
(c) Microsoft Corporation. All rights reserved.

C:\Users\csmit\OneDrive\Documents\GitHub\Python>cd Achievement1\Exercise2.1
The system cannot find the path specified.

C:\Users\csmit\OneDrive\Documents\GitHub\Python>cd Achievement2\Exercise2.1

C:\Users\csmit\OneDrive\Documents\GitHub\Python\Achievement2\Exercise2.1>mkvirtualenv achievement2-practice
created virtual environment CPython3.11.4.final.0-64 in 552ms
  creator CPython3Windows(dest=C:\Users\csmit\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\csmit\AppData\Local\pypa\virtualenv)
    added seed packages: pip==24.0, setuptools==69.2.0, wheel==0.43.0
  activators BashActivator,BatchActivator,FishActivator,NushellActivator,PowerShellActivator,PythonActivator

(achievement2-practice) C:\Users\csmit\OneDrive\Documents\GitHub\Python\Achievement2\Exercise2.1>
```

Django Version:



```
created virtual environment CPython3.11.4.final.0-64 in 5061ms
  creator CPython3Windows(dest=C:\Users\csmit\Envs\web-dev, 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\csmit\AppData\Local\pypa\virtualenv)
    added seed packages: pip==24.0, setuptools==69.2.0, wheel==0.43.0
  activators BashActivator,BatchActivator,FishActivator,NushellActivator,PowerShellActivator,PythonActivator

(web-dev) C:\Users\csmit\OneDrive\Documents\GitHub\Python\Achievement2\Exercise2.1>...\> web-dev\Scripts\activate.bat
The system cannot find the path specified.

(web-dev) C:\Users\csmit\OneDrive\Documents\GitHub\Python\Achievement2\Exercise2.1>cd C:\Users\csmit\Envs\web-dev\Scripts\activate.bat
The directory name is invalid.

(web-dev) C:\Users\csmit\OneDrive\Documents\GitHub\Python\Achievement2\Exercise2.1>cd C:\Users\csmit\Envs\web-dev\Scripts

(web-dev) C:\Users\csmit\Envs\web-dev\Scripts>activate.bat

(web-dev) C:\Users\csmit\Envs\web-dev\Scripts>if defined _OLD_VIRTUAL_PYTHONPATH (set "PYTHONPATH=" )  else (set "_OLD_VIRTUAL_PYTHONPATH=" )

(web-dev) C:\Users\csmit\Envs\web-dev\Scripts>py -m pip install Django
Collecting Django
  Downloading Django-5.0.4-py3-none-any.whl.metadata (4.1 kB)
Collecting asgiref<4,>=3.7.0 (from Django)
  Downloading asgiref-3.8.1-py3-none-any.whl.metadata (9.3 kB)
Collecting sqlparse>=0.3.1 (from Django)
  Downloading sqlparse-0.5.0-py3-none-any.whl.metadata (3.9 kB)
Collecting tzdata (from Django)
  Downloading tzdata-2024.1-py2.py3-none-any.whl.metadata (1.4 kB)
Downloading Django-5.0.4-py3-none-any.whl (8.2 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 8.2/8.2 MB 24.9 MB/s eta 0:00:00
Downloading asgiref-3.8.1-py3-none-any.whl (23 kB)
Downloading sqlparse-0.5.0-py3-none-any.whl (43 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 44.0/44.0 kB ? eta 0:00:00
Downloading tzdata-2024.1-py2.py3-none-any.whl (345 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 345.4/345.4 kB 20.9 MB/s eta 0:00:00
Installing collected packages: tzdata, sqlparse, asgiref, Django
Successfully installed Django-5.0.4 asgiref-3.8.1 sqlparse-0.5.0 tzdata-2024.1

(web-dev) C:\Users\csmit\Envs\web-dev\Scripts>django-admin --version
5.0.4

(web-dev) C:\Users\csmit\Envs\web-dev\Scripts>
```