

Exercise 2.1 Getting Started with Django

Task 2. Why Django is popular

Django’s popularity stems from its "batteries-included" philosophy, providing built-in tools for authentication, database management, and security that allow developers to skip the repetitive "scaffold" work. Its focus on rapid development and clean, pragmatic design makes it the go-to choice for meeting tight deadlines without sacrificing code quality. Furthermore, because it is written in Python, it inherits a massive ecosystem of libraries for everything from AI to data science, making it incredibly versatile.

Task 3. Five companies using Django

Many industry giants rely on Django to handle impressive scale and complex data.

Company	Product/Service	Use of Django
Instagram	Social Media Platform	Handles the massive volume of user interactions (likes, photos, follows) and manages its API.
Spotify	Audio Streaming Service	Uses Django for its backend services, specifically for data analysis and high-speed performance.
Dropbox	Cloud Storage	Powers the backend for its web interface and handles the complex file-sharing logic.
NASA	Space Agency	Uses Django for various official websites and internal applications to manage high-security scientific data.
Pinterest	Visual Discovery Engine	Managed its initial rapid growth using Django's scalability to handle millions of images and user pins.

Task 4. Django Scenarios: Yes or No?

You need to develop a web application with multiple users.

- **Verdict:** Yes.
- **Why:** Django has a robust, built-in Authentication System out of the box. It handles user registration, logins, permissions, and groups with very little setup, making it ideal for multi-user environments.

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

- **Verdict:** Yes.
- **Why:** Known as the framework for "perfectionists with deadlines", Django's Model-View-Template (MVT) structure and automated admin panel allow me to build and iterate extremely quickly.

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

- **Verdict:** No.
- **Why:** Django is quite "heavy" for a simple static site. It forces a specific file structure and includes a lot of overhead (like the ORM and middleware) that I don't need in this case. A micro-framework like Flask would be a much lighter, faster choice here.

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

- **Verdict:** No.
- **Why:** Django is opinionated. It expects you to follow its specific way of doing things (the "Django way"). If I want to be able to hand-pick every library and design the architecture from the ground up, I will find myself fighting against Django's built-in features.

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

- **Verdict:** Yes.
- **Why:** Django has some of the best documentation in the software world. Beyond the official docs, it has a massive, mature community and thousands of third-party packages (via *Django Packages*) to solve almost any problem I might encounter.

Task 5. Python version

```
Command Prompt
(web-dev) C:\Users\dmkor>cd Scripts
The system cannot find the path specified.

(web-dev) C:\Users\dmkor>py -m pip install Django
Collecting Django
  Downloading django-6.0.2-py3-none-any.whl.metadata (3.9 kB)
Collecting asgiref<=3.9.1 (from Django)
  Downloading asgiref-3.11.1-py3-none-any.whl.metadata (9.3 kB)
Collecting sqlparse>=0.5.0 (from Django)
  Downloading sqlparse-0.5.5-py3-none-any.whl.metadata (4.7 kB)
Collecting tzdata (from Django)
  Downloading tzdata-2025.3-py2.py3-none-any.whl.metadata (1.4 kB)
Downloading django-6.0.2-py3-none-any.whl (8.3 MB)
 8.3/8.3 MB 10.6 MB/s 0:00:00
Downloading asgiref-3.11.1-py3-none-any.whl (24 kB)
Downloading sqlparse-0.5.5-py3-none-any.whl (46 kB)
Downloading tzdata-2025.3-py2.py3-none-any.whl (348 kB)
Installing collected packages: tzdata, sqlparse, asgiref, Django
Successfully installed Django-6.0.2 asgiref-3.11.1 sqlparse-0.5.5 tzdata-2025.3

[notice] A new release of pip is available: 25.3 -> 26.0.1
[notice] To update, run: python.exe -m pip install --upgrade pip

(web-dev) C:\Users\dmkor>django-admin --version
6.0.2

(web-dev) C:\Users\dmkor>python --version
Python 3.14.2

(web-dev) C:\Users\dmkor>
```

Task 7. Virtual environment

```
Command Prompt
Downloading asgiref-3.11.1-py3-none-any.whl (24 kB)
Downloading sqlparse-0.5.5-py3-none-any.whl (46 kB)
Downloading tzdata-2025.3-py2.py3-none-any.whl (348 kB)
Installing collected packages: tzdata, sqlparse, asgiref, Django
Successfully installed Django-6.0.2 asgiref-3.11.1 sqlparse-0.5.5 tzdata-2025.3

[notice] A new release of pip is available: 25.3 -> 26.0.1
[notice] To update, run: python.exe -m pip install --upgrade pip

(web-dev) C:\Users\dmkor>django-admin --version
6.0.2

(web-dev) C:\Users\dmkor>python --version
Python 3.14.2

(web-dev) C:\Users\dmkor>workon a2-practice

virtualenv "a2-practice" does not exist.
Create it with "mkvirtualenv "

C:\Users\dmkor>mkvirtualenv a2-practice
created virtual environment CPython3.14.2.final.0-64 in 373ms
creator CPythonWindows(dest=C:\Users\dmkor\Envs\a2-practice, clear=False, no_vcs_ignore=False, global=False)
seeder FromAppData(download=False, pip=bundle, via=copy, app_data_dir=C:\Users\dmkor\AppData\Local\pypa\virtualenv)
added seed packages: pip==25.3
activators BashActivator,BatchActivator,FishActivator,NushellActivator,PowerShellActivator,PythonActivator

(a2-practice) C:\Users\dmkor>cd Documents/cf-projects/cf-python

(a2-practice) C:\Users\dmkor\Documents\cf-projects\cf-python>
```

Task 8. Django installation

```
Command Prompt
creator CPython3Windows(dest=C:\Users\dmkor\Envs\A2-practice, clear=False, no_vcs_ignore=False, global=False)
seeder FromAppData(download=False, pip=bundle, via=copy, app_data_dir=C:\Users\dmkor\AppData\Local\pypa\virtualenv)
added seed packages: pip==25.3
activators BashActivator,BatchActivator,FishActivator,NushellActivator,PowerShellActivator,PythonActivator

(a2-practice) C:\Users\dmkor>cd Documents\cf-projects\cf-python

(a2-practice) C:\Users\dmkor\Documents\cf-projects\cf-python>py -m pip install Django
Collecting Django
  Using cached django-6.0.2-py3-none-any.whl.metadata (3.9 kB)
Collecting asgiref<=3.9.1 (from Django)
  Using cached asgiref-3.11.1-py3-none-any.whl.metadata (9.3 kB)
Collecting sqlparse<=0.5.0 (from Django)
  Using cached sqlparse-0.5.5-py3-none-any.whl.metadata (4.7 kB)
Collecting tzdata (from Django)
  Using cached tzdata-2025.3-py2.py3-none-any.whl.metadata (1.4 kB)
Using cached django-6.0.2-py3-none-any.whl (8.3 MB)
Using cached asgiref-3.11.1-py3-none-any.whl (24 kB)
Using cached sqlparse-0.5.5-py3-none-any.whl (46 kB)
Using cached tzdata-2025.3-py2.py3-none-any.whl (348 kB)
Installing collected packages: tzdata, sqlparse, asgiref, Django
Successfully installed Django-6.0.2 asgiref-3.11.1 sqlparse-0.5.5 tzdata-2025.3

[notice] A new release of pip is available: 25.3 -> 26.0.1
[notice] To update, run: python.exe -m pip install --upgrade pip

(a2-practice) C:\Users\dmkor\Documents\cf-projects\cf-python>django-admin --version
6.0.2

(a2-practice) C:\Users\dmkor\Documents\cf-projects\cf-python>
```