2. Django is based on Python, which means that it inherits many of the inherent benefits of developing using that language, such as easy to learn/simple syntax and wide variety of available modules and libraries to import into an application. Secondly, Django has a wide variety of built in features that simply the process of creating a web application, which as providing built in methods for handling forms, session handling, and user authentication, just to name a few.

3.

a. Instagram - Social media photo and video sharing platform. Instagram uses Django to power its entire backend infrastructure. This allowed for the app to continue to perform even as the user-base increased in size.
b. Spotify - music streaming service. Spotify uses Django for backend services and data analysis, which allows them to handle their large user base, and to recommend music to users based on their behavior, via their machine learning algorithms.
c. Disqus - Service powering comments/polls on websites. Django was the primary framework used in creating the website, allowing for quick development and scaling.
d. Pinterest - social media platform allowing user's to share boards of "pinned" posts. Originally developed with Django, currently the framework powers user's ability to subscribe to other users and share their own boards.
e. The Washington Post - Major news organization. WP uses Django to ensure the scalability of their website, allowing millions of users to access large amounts of data without issue.

4.

a. Yes, I would use Django. Django's built in user management systems (authentication, authorization, session management, etc.) would make developing this application easier.
b. Yes, I would use Django. This is an ideal use case, and Django's built in features and components make development much quicker, and allow for changes to be made quickly.
c. No, I would not use Django. This would be overkill, as many of the advantages of Django would not be relevant here, and the structure of the application would be unnecessarily complex for a simple project.
d. No, I would not use Django. There is a certain loss of control with using Django, as the framework requires some things to be done in certain ways (for example, the project structure, the implementation of some of the built-in components), and if a developer wants total control of everything Django would not be a good fit.
e. Yes, I would use Django. Django has a number of benefits in this area. First, there is a track record of success, with many extremely successful applications having been created with Django. Secondly, there are many resources for assisting with Django development, including Django's own documentation, and many user created resources due to Django's popularity.

Screenshot showing Python version, new virtual environment, Django installation, and Django version:

```
(cf-python-base) andrewhenry@Andrews-MBP Career_Foundry % python --version
Python 3.8.7
(cf-python-base) andrewhenry@Andrews-MBP Career_Foundry % deactivate
andrewhenry@Andrews-MBP Career_Foundry % mkvirtualenv achievement2-practice
created virtual environment CPython3.8.7.final.0-64 in 926ms
  creator CPython3Posix(dest=/Users/andrewhenry/.virtualenvs/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=/Users/andrewhenry/Library/Application Supp
ort/virtualenv)
    added seed packages: pip==25.0.1, setuptools==75.3.2, wheel==0.45.1
  activators BashActivator,CShellActivator,FishActivator,NushellActivator,PowerShellActivator,PythonActivator
virtualenvwrapper.user_scripts creating /Users/andrewhenry/.virtualenvs/achievement2-practice/bin/predeactivate
virtualenvwrapper.user_scripts creating /Users/andrewhenry/.virtualenvs/achievement2-practice/bin/postdeactivate
virtualenvwrapper.user_scripts creating /Users/andrewhenry/.virtualenvs/achievement2-practice/bin/preactivate
virtualenvwrapper.user_scripts creating /Users/andrewhenry/.virtualenvs/achievement2-practice/bin/postactivate
virtualenvwrapper.user_scripts creating /Users/andrewhenry/.virtualenvs/achievement2-practice/bin/get_env_details
(achievement2-practice) andrewhenry@Andrews-MBP Career_Foundry % pip install django
Collecting django
  Using cached django-4.2.26-py3-none-any.whl.metadata (4.2 kB)
Collecting asgiref<4,>=3.6.0 (from django)
  Using cached asgiref-3.8.1-py3-none-any.whl.metadata (9.3 kB)
Collecting backports.zoneinfo (from django)
  Using cached backports.zoneinfo-0.2.1-cp38-cp38-macosx_10_14_x86_64.whl.metadata (4.7 kB)
Collecting sqlparse>=0.3.1 (from django)
  Using cached sqlparse-0.5.4-py3-none-any.whl.metadata (4.7 kB)
Collecting typing-extensions>=4 (from asgiref<4,>=3.6.0->django)
  Using cached typing_extensions-4.13.2-py3-none-any.whl.metadata (3.0 kB)
Using cached django-4.2.26-py3-none-any.whl (8.0 MB)
Using cached asgiref-3.8.1-py3-none-any.whl (23 kB)
Using cached sqlparse-0.5.4-py3-none-any.whl (45 kB)
Using cached backports.zoneinfo-0.2.1-cp38-cp38-macosx_10_14_x86_64.whl (35 kB)
Using cached typing_extensions-4.13.2-py3-none-any.whl (45 kB)
Installing collected packages: typing-extensions, sqlparse, backports.zoneinfo, asgiref, django
Successfully installed asgiref-3.8.1 backports.zoneinfo-0.2.1 django-4.2.26 sqlparse-0.5.4 typing-extensions-4.13.2
(achievement2-practice) andrewhenry@Andrews-MBP Career_Foundry % django-admin --version
4.2.26
```