Section 8

Week of April 2

Outline

- A few notes
- Django: Admin Features
- Django: More
- Testing
- CI/CD

A few notes:

- Project 3 due on April 15th
- Last project before the final project

Django: Admin Features

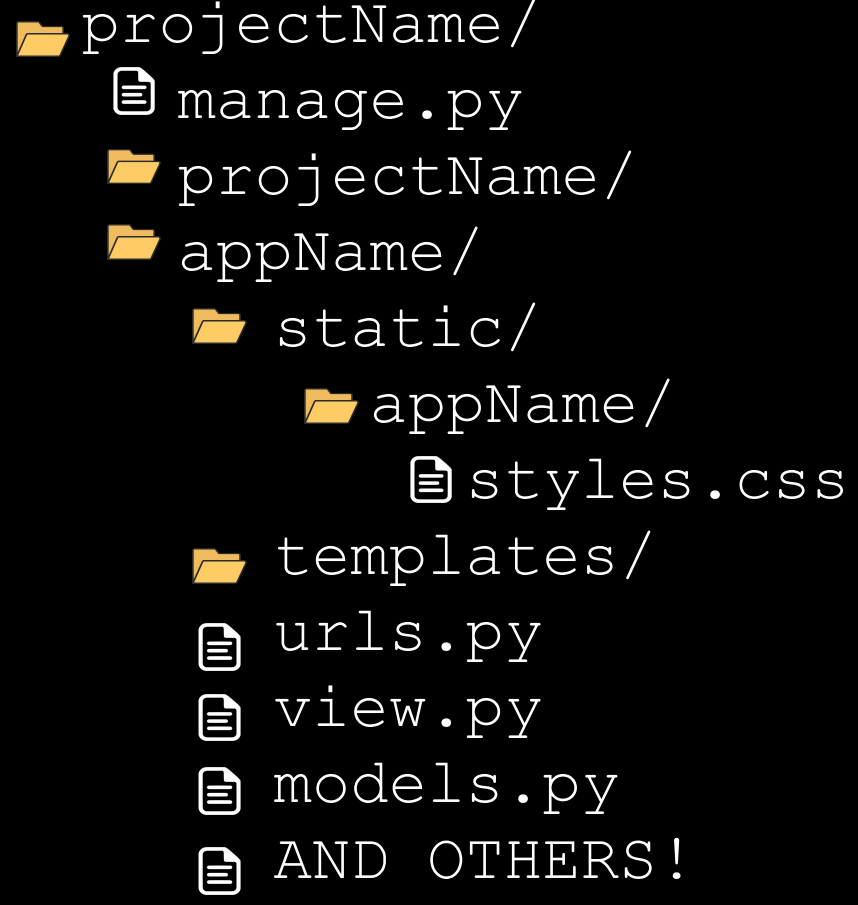
- https://docs.djangoproject.com/en/2.2/ref/contrib/admin/
- Customize your admin page for easier data manipulation

```
from django.contrib import admin
from .models import [name]

@admin.register([name])
class [name]Admin(admin.ModelAdmin):
    fields = ([col-1], [col-2], [col-3])
    filter_horizontal = ([many-to-many-field])
```

Django: Static Files

```
<link rel="stylesheet" href="{% static 'appName/styles.css' %}">
```



Any Django questions related to project 3?

Testing

- *Test-driven development*: Write specific test cases first and tune the development to those cases

How to Write Good Tests:

- Don’t think tests will work for your code? You wrote your code wrong
- Correctness
- Independent, Repeatable
  - Testing random numbers? Testing non-determinate functions?
- Well-organized
  - How to maintain a good code base
  - Figure out what code is doing via tests
- Portable, Reusable
- Provide Information
- Fast

Testing: Unittest

- *Unittest*: a python library with testing capabilities
- https://docs.python.org/3/library/unittest.html

```
import unittest

class Tests(unittest.TestCase):
    def testName(self):
        “””A descriptive sentence to print with the test“””
        # Write your test here

if __name__ == “__main__”:
    unittest.main()
```

Testing: Selenium

- https://www.seleniumhq.org/docs/03_webdriver.jsp#introducing-webdriver
- *Selenium:* a library for simulated browser interaction in python
  - Used combined with unittest to create browser tests

CI/CD

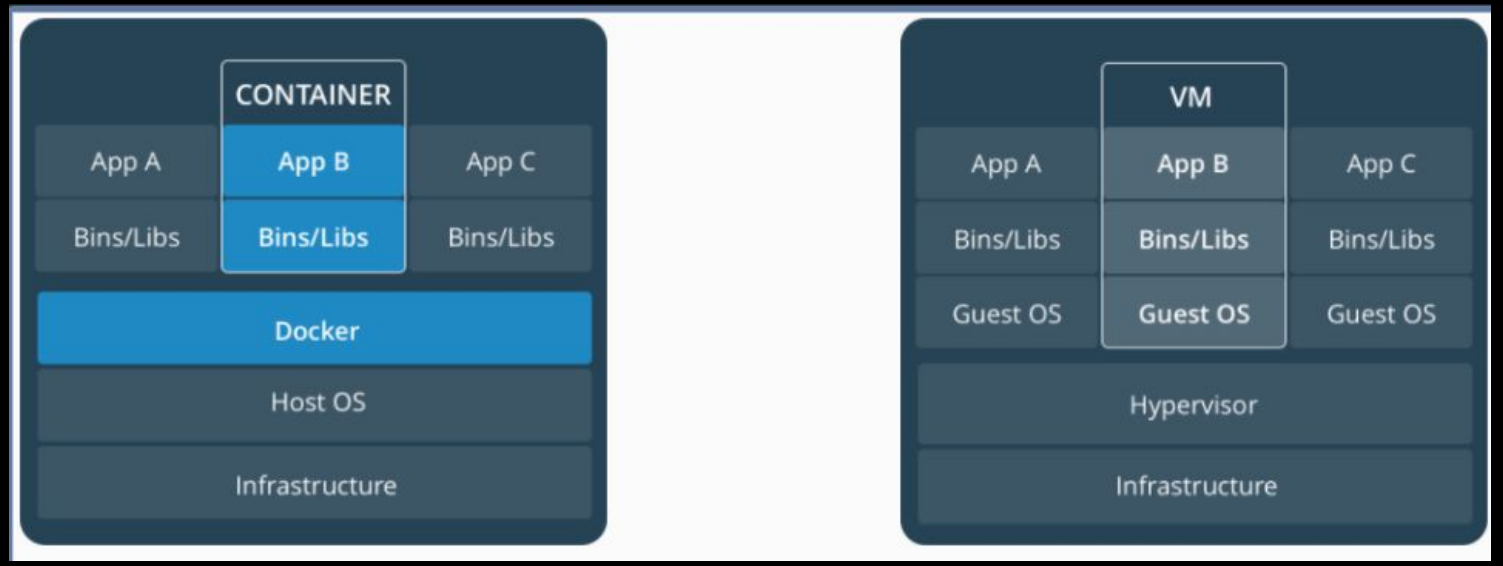
- *Continuous Integration*: continually merge small changes into the main branch, generally after they have passed test cases
- *Continuous Deployment*: deploy small changes frequently
- *Travis*: a tool for automatically running tests once pushed to github
  - Create a Travis account and link to your github
  - Travis pulls code when changes occur to run tests
  - Create a YAML file to configure
  - https://travis-ci.org/

YAML: A file format for configuration files

```
key1: value 1
key2: value 2
key 3:
    - value 1
    - value 2
    - value 3
```

Docker

- *Docker:* a program which creates *containers*, each of which has all the files and settings necessary to run an app
- *Containerization*: creating these containers, or *images*
- Controlled using a Dockerfile and a YAML file
- https://docs.docker.com/engine/docker-overview/
- `docker compose up` (Command to run a dockerfile)



Any remaining questions?