How we gonna do this fr irl tho

- See W5-2 slides for some resources to add here

- Python variable naming conventions are different than what we originally settled on

General

- Flask Tutorial in Visual Studio Code
- Need to download Python ide / environment - anaconda?
- Flask web dev for Python
- Are we supposed to use Bootstrap too or does Flask cover that?

Graphical representation

- Matplotlib: Visualization with Python

Notification system

- Twilio's email API for Python - Rohit mentioned something about free credits

Map

- Leaflet.js is what Rohit used

Database

- This is a pretty integral part, and I know nothing about it. Database peeps?
  Data management - Database
  - PyMongo? - i actually don't think this is what we should use
  - MySQL
    - Python/MySQL Connector
  - SQLite - flask supports
  - SQLAlchemy - flask supports

Server?

- How does web prog work

Testing

- Flask provides utilities for testing an application. (They use the pytest framework)

UI

- Bootstrap framework / other css framework? Otherwise we have to write css ourselves which can take forever
- This also makes the ui flexible
- Adding Bootstrap to Flask - this guy's tutorials seem really helpful actually

Production/Deployment

- Heroku??

Python/Virtual Environment Setup

Install a python interpreter:
Vs code tutorial

- Download it from Python.org - version 3.12.2, check the boxes “Use admin privileges when installing py.exe” and “Add python.exe to PATH”
- Also get the VSCode python extensions

Setting up Flask
vs code flask

-
- Tips for problems during the tutorial
- Selecting python interpreter: mine was in ~/appdata/programs/python/python312/python.exe
-

Everyone needs to create their own virtual environment, but we can run commands to make sure it is the same (pip freeze stuff)

What does Flask do?
Flask’s library provides methods and objects that act as code abstractions for components and processes of your web application. In other words, instead of having to understand how all this stuff works in Internet Architecture and all that, you can just do Python code to control your app.
> methods

- Redirect, abort
- url_for

> objects

- Flask - an object for the application itself (i think)
- Request - cookies, post and get methods, file uploads, other stuff
- Session
- Response

Flask is also the glue holding everything together. It’s called a ‘microframework’ because there is a lot it doesn’t do: HTML template rendering, database stuff, etc… BUT it provides extensions that allow you to use other tools/libraries alongside it (eg Jinja2, SQLAlchemy etc)

BASIC SEQUENCE DESCRIPTION

1. HTTP request sent to app, with a specific URL.
    - ‘GET’ -> asks for data from the server
    - ‘POST’ -> sends data to the server
2. App searches for the route decorator associated with that URL
3. App calls the view method associated with that route decorator
4. Method interacts with the database (sends/retrieves data) and does some logic
5. Method terminates in one of two ways:
    a. Redirects to another URL, thereby calling another view method, OR
    b. Step 6
6. Asks for a web page (ie html document) to be rendered, by calling render_template(*‘nameOfPage.html’, variablesFromDataBase…*)
7. The html document, called a template, contains Jinja2 syntax, which does two things:
    a. Allows the template to inherit elements from a base template, so you don’t have to re-write html for similar web pages
    b. Uses the passed variables to perform logic operations that decide exactly what html will be displayed
8. The Jinja2 template engine renders the html document so it can be displayed to the user as a web page