**Flask:**

A framework that works in conjunction with Python

To use: create a python file (.py) headed by the following:

**from flask import Flask**, …. , (other modules), render\_template, redirect, request, session, flash

**app=Flask(\_\_name\_\_)** ---- this instantiates your app with an object constructor

**@app.route(‘/’)**--- creates a route to your index page using the text decorator @, a reference to your app and the method route. To refer to another page, replace ’/’ with ‘/other\_page’

**def index():**

**return render\_template(‘index.html’)**----- will load a page titled index.html upon initializing

---- insert a few more routing options to refer to different pages

**app.run(debug=True)**--- final command to run the program in debug mode

Organization is important. You will place all of your webpages in a folder called *templates* on the same level as your python file. Static files go in a folder called *static*. Place your css, images and js folders and files here.

To link to CSS, you need to use a static address like href="{{ url\_for('static', filename='css/styles.css') }}"

There are some issues updating the cache if you are changing CSS. Use ctr+shf+r for a hard reload to show changes

Helpful Modules:

render\_template- if you return render\_template at the end of your route, it will load a specified HTML template that you will have created and stored under templates render\_template(‘index.html’)

redirect- redirects to a selected path redirect(‘/’) or redirect(‘/submit’)

request- can request info in the form of an object from an HTML form with a ‘post’ method. It calls on the name property eg name=’result’ and in your python script you can access it by request.form[‘result’]. You may want to assign your data to a variable to use later.

session- without session, you won’t be able to save data in the long term. Most get and post requests are stateless. Session makes a cookie to recall everything stored in the session object. Session also requires a secret key to use like app.secret\_key = 'ThisIsSecret'

flash- flash messages allow you to validate data coming through the page. In your Python page, you can enter flash message strings. To display your messages, in your HTML you need to request the message by:

{% with messages = get\_flashed\_messages() %}

{% if messages %}

{% for message in messages %}

{{message}}

{% endfor %}

{% endif %}

{% endwith %}

**Variables:** you can post data on your HTML through a render-template return . In your HTML, enclose your variable in double braces {{Variable}}.

You can also create logic statements in your HTML through:

{% if value<10 %}

{{variable}}

{% else %}

{{other-variable}}

{% endif %}

Finally run the program in terminal by navigating to the directory with your python file in terminal and running: python file\_name.py you can access your program through localhost:5000