CSC 103 - HW 7

Complete each of the following exercises. To receive full credit please ensure the following:

- Submission structure follows the following:

```
HW7
    -> data
            -> poke.csv
            . . . .
    -> Q1
            -> q1.py
            . . . .
    -> Q2
            -> q2.py
            . . . .
    -> Q10
            -> q10.py
            . . . .
```

- Exercise artifacts are uploaded to your GitHub repository under a folder called `HW7` (i.e. https://github.com/<username>/carlow-me)
- Exercise artifacts are uploaded to Brightspace as a compressed `.zip` file called `<LastName>_HW7.zip`.

It is crucial that you are naming artifacts and storing files in the format mentioned above. All files that need to reference datasets should use a relative reference that refers to it's location in the `data` directory and not an absolute reference that would only work on your machine. Failure to meet any of these requirements will result in loss of points for that and any following problems.

Exercises

1. Create a flask application that responds with “Hello, `<Pokemon>`!”. Where <Pokemon> is the name of a random pokemon found in the `pokemon.csv` data file that we’ve been working with in previous assignments. I should be able to hit your application at `http://localhost:5690` and receive my desired response.

2. Given the following code:

```
from flask import Flask

app = Flask(__name__)

@app.route('/')
def index():
    return "Hello World!"

app.run(host='0.0.0.0', port=9999, debug=True)
```

If I attempted to run two copies of the application on the same computer, what do you believe would happen? If you believe there to be problems, what would you do to resolve the issue. Provide your answer in a file called **Q2.txt**

Hint: Run two applications of the application on your computer and it should enlighten you with some insights into this question.

3. Given the following code:

```
from flask import Flask

app = Flask(__name__)

@app.route('/')
def index():
    return "Hello World!"

app.run(host='0.0.0.0', port=9999, debug=True)
```

If I wanted to ensure that this was only available on my local machine and not any public IP address available on my machine, what would I change? Provide your answer in a file called **Q3.txt**

4. Write a flask application that responds “Hello, User!” when accessing the root of your application from a browser. However, when the root of your application is accessed via a **POST** action, respond with “Hello, Computer!”. I should be able to access this application at http://localhost:4356.
5. Write a flask application that represents a pokemon team management system. There should be endpoints `create`, `list` and `delete` that allow you to generate a random team, show that random team and delete that random team. Each of the endpoints should accept only their respective HTTP method. If a random team already exists, you

should return a message telling the user that a team already exists. I should be able to access these endpoints at http://localhost:8989.

6. Expanding on **Q5**, let's start doing some basic logging to a file. Using tooling that we already know, let's write each action that our API takes to a file called **out.log**. When an action is taken (create, list, or delete) write a log following this example:

   Action - HTTP_METHOD - <pokmon> (added/deleted/etc)