Lab 1 – Flask Intro

Task – Create a basic Flask Application

Prerequisites (Same for all exercises): Have an EC2 with port 22, 80, and 5000 open. Install python and flask through a python environment with the following commands:

```
`sudo apt update`
`sudo apt install python3 python3-pip python3-venv`
`python3 -m venv venv`
`source venv/bin/activate`
`pip3 install flask`
```

1) Create a new file called app.py and enter the following:

```
from flask import Flask

app = Flask(__name__)

@app.route('/')
def hello_internet():
    return "Hello Internet!"

if __name__=='__main__':
    app.run(debug=True, host='0.0.0.0', port=5000)
```

2) Create a new route that listens for /name and returns 'hello <name>' where your name is a variable within the app being returned

3) Access this route by either curl localhost:5000/name or accessing this through a browser

Lab 2 – Flask Routes

Task – Add more routes to your flask app

Create a unique route for each type of request (GET, POST, PUT, DELETE) that returns text saying what type of route it is.

Use Postman.com to send the requests to your flask app.

Stretch goal – Create a single route that can take in any of the methods. Depending on the method sent, it should return different text.

Lab 3 – Request Data

Task – Using Data Requests

Create the following routes within your flask app using Request Bodies, URL params, and query params. The route should return the data being sent, and you should use your best judgement about which method(s) of sending request data:

- Create a new record
- Get record by ID
- Get all records by a query
- Delete a record by name
- Update a record by ID

Stretch goal – Use the Postman dynamic variables to make your requests more interesting.