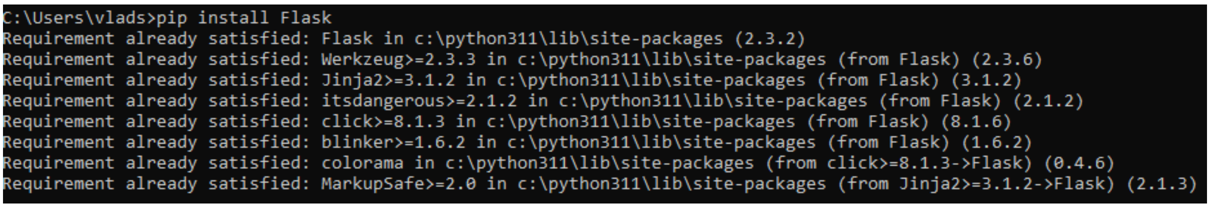
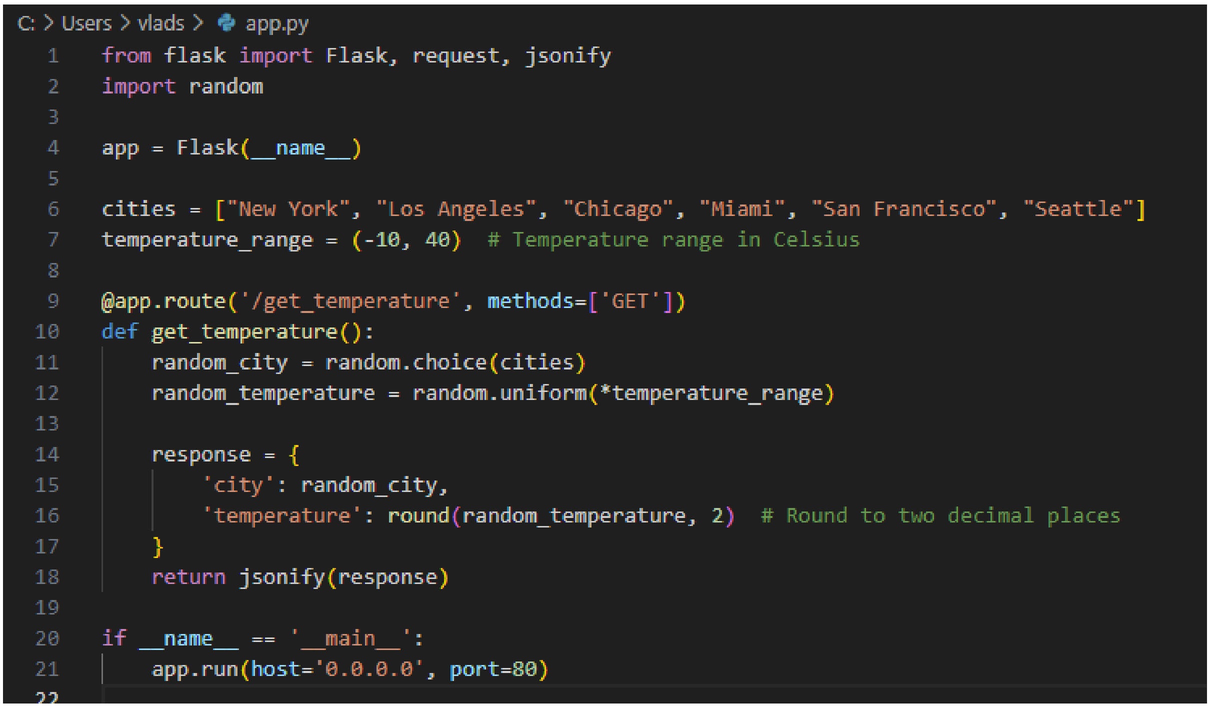
I created a web application that provides temperature information from a city using a Python Flask application.

I already had installed the Flask application:



```
C:\Users\vlads>pip install Flask
Requirement already satisfied: Flask in c:\python311\lib\site-packages (2.3.2)
Requirement already satisfied: Werkzeug>=2.3.3 in c:\python311\lib\site-packages (from Flask) (2.3.6)
Requirement already satisfied: Jinja2>=3.1.2 in c:\python311\lib\site-packages (from Flask) (3.1.2)
Requirement already satisfied: itsdangerous>=2.1.2 in c:\python311\lib\site-packages (from Flask) (2.1.2)
Requirement already satisfied: click>=8.1.3 in c:\python311\lib\site-packages (from Flask) (8.1.6)
Requirement already satisfied: blinker>=1.6.2 in c:\python311\lib\site-packages (from Flask) (1.6.2)
Requirement already satisfied: colorama in c:\python311\lib\site-packages (from click>=8.1.3->Flask) (0.4.6)
Requirement already satisfied: MarkupSafe>=2.0 in c:\python311\lib\site-packages (from Jinja2>=3.1.2->Flask) (2.1.3)
```

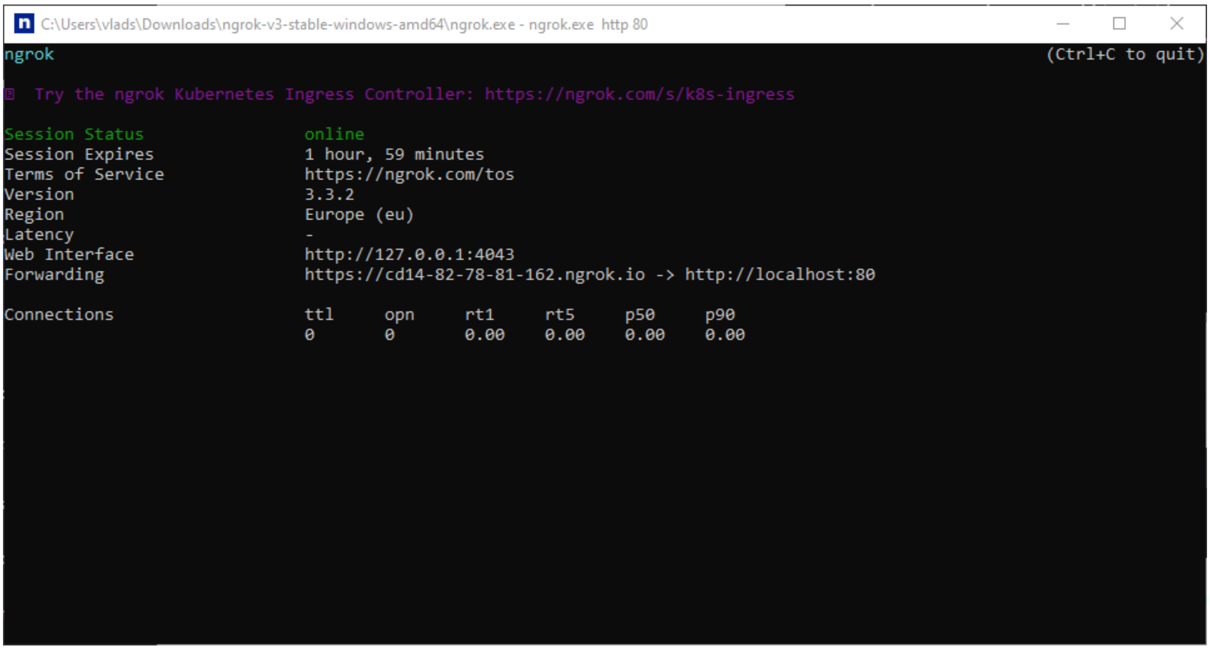
I created a file named app.py and added the following code:



```python
from flask import Flask, request, jsonify
import random

app = Flask(__name__)

cities = ["New York", "Los Angeles", "Chicago", "Miami", "San Francisco", "Seattle"]
temperature_range = (-10, 40)  # Temperature range in Celsius

@app.route('/get_temperature', methods=['GET'])
def get_temperature():
    random_city = random.choice(cities)
    random_temperature = random.uniform(*temperature_range)

    response = {
        'city': random_city,
        'temperature': round(random_temperature, 2)  # Round to two decimal places
    }
    return jsonify(response)

if __name__ == '__main__':
    app.run(host='0.0.0.0', port=80)
```

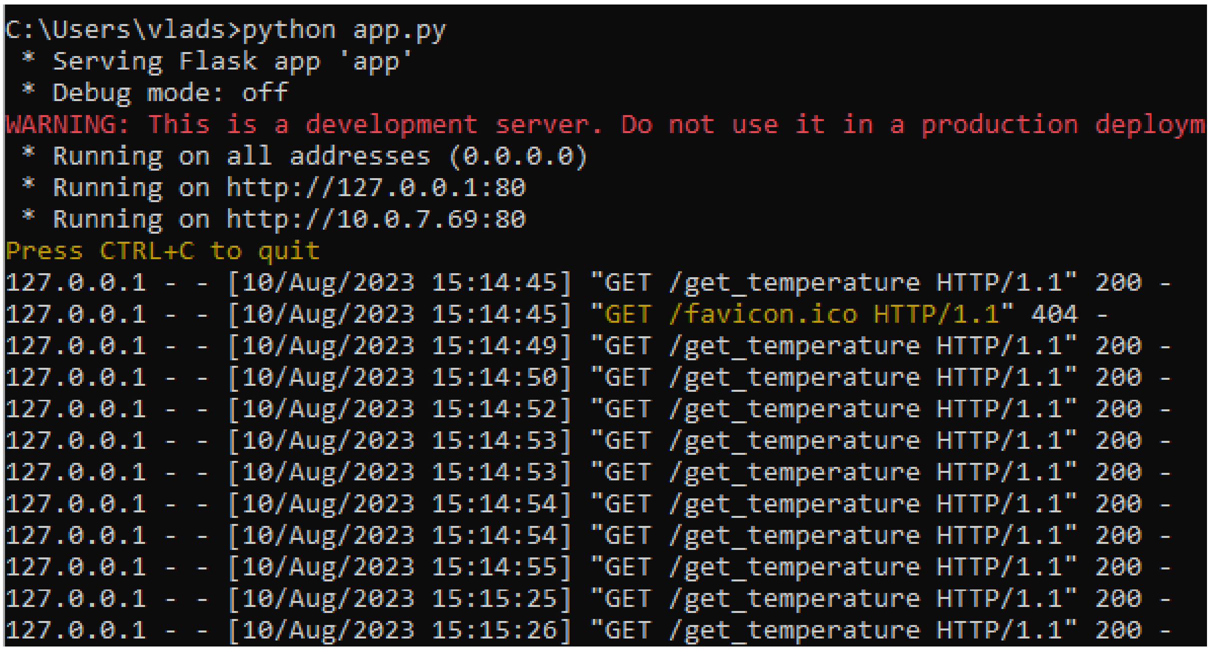
We have a list of cities and a temperature range defined. The get_temperature function now selects a random city from the list and generates a random temperature within the specified range. The temperature is rounded to two decimal places for cleaner presentation.

To make my Flask app accessible over the internet, I used a tool named Ngrok. This will provide me with a public URL that tunnels to my local Flask app. I wanted to run the Flask app in port 80 so I inserted the command: ngrok.exe http 80

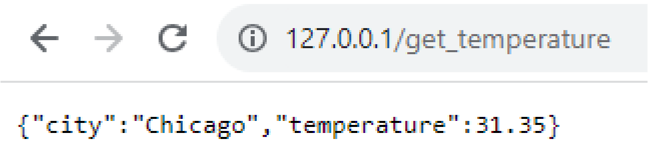

C:\Users\vlads\Downloads\ngrok-v3-stable-windows-amd64\ngrok.exe - ngrok.exe http 80

```
ngrok                                                                                          (Ctrl+C to quit)

  Try the ngrok Kubernetes Ingress Controller: https://ngrok.com/s/k8s-ingress

Session Status                online
Session Expires               1 hour, 59 minutes
Terms of Service              https://ngrok.com/tos
Version                       3.3.2
Region                        Europe (eu)
Latency                       -
Web Interface                 http://127.0.0.1:4043
Forwarding                    https://cd14-82-78-81-162.ngrok.io -> http://localhost:80

Connections                   ttl     opn     rt1     rt5     p50     p90
                              0       0       0.00    0.00    0.00    0.00
```

I ran the python app.py command and now I receive the results on the “http://127.0.0.1:80/get_temperature” address.



```
C:\Users\vlads>python app.py
 * Serving Flask app 'app'
 * Debug mode: off
WARNING: This is a development server. Do not use it in a production deploym
 * Running on all addresses (0.0.0.0)
 * Running on http://127.0.0.1:80
 * Running on http://10.0.7.69:80
Press CTRL+C to quit
127.0.0.1 - - [10/Aug/2023 15:14:45] "GET /get_temperature HTTP/1.1" 200 -
127.0.0.1 - - [10/Aug/2023 15:14:45] "GET /favicon.ico HTTP/1.1" 404 -
127.0.0.1 - - [10/Aug/2023 15:14:49] "GET /get_temperature HTTP/1.1" 200 -
127.0.0.1 - - [10/Aug/2023 15:14:50] "GET /get_temperature HTTP/1.1" 200 -
127.0.0.1 - - [10/Aug/2023 15:14:52] "GET /get_temperature HTTP/1.1" 200 -
127.0.0.1 - - [10/Aug/2023 15:14:53] "GET /get_temperature HTTP/1.1" 200 -
127.0.0.1 - - [10/Aug/2023 15:14:53] "GET /get_temperature HTTP/1.1" 200 -
127.0.0.1 - - [10/Aug/2023 15:14:54] "GET /get_temperature HTTP/1.1" 200 -
127.0.0.1 - - [10/Aug/2023 15:14:54] "GET /get_temperature HTTP/1.1" 200 -
127.0.0.1 - - [10/Aug/2023 15:14:55] "GET /get_temperature HTTP/1.1" 200 -
127.0.0.1 - - [10/Aug/2023 15:15:25] "GET /get_temperature HTTP/1.1" 200 -
127.0.0.1 - - [10/Aug/2023 15:15:26] "GET /get_temperature HTTP/1.1" 200 -
```



```
{"city":"Chicago","temperature":31.35}
```