app.py

```python
import dht_sensor
import wind_breaker
import asyncio
import time
import requests
from datetime import datetime
from zoneinfo import ZoneInfo
```

secţiunea în care se importă toate modulele externe şi librăriile necesare

```python
async def read_sensors(breaker_time=60) -> tuple[float, float, float]:
    breaks_task = asyncio.create_task(wind_breaker.get_breaks(breaker_time))
    await asyncio.sleep(0)  # actually start breaks_task
    humidity, temperature = dht_sensor.get_both()
    breaks = await breaks_task

    return (humidity, temperature, breaks)
```

funcţia asincronă care realizează citirea senzorilor
aceasta iniţial porneşte thread-ul care numără numărul de întreruperi care s-au întâmplat de-a lungul minutului pentru care se execută funcţia, în timp ce threadul principal realizează citirea senzorului de umiditate şi temperatură

```python
if __name__ == "__main__":
    while True:
        try:
            t = time.time()

            values = []
            breaks = 0

            print(f"started at {t}")
            while True:
                humidity, temperature, crt_breaks = asyncio.run(
                    read_sensors(breaker_time=60)
                )

                values.append([humidity, temperature])
                breaks += crt_breaks
                print(
                    f"humidity: {humidity}, temperature: {temperature}, breaks: {crt_breaks}"
                )

                if crt_hour != datetime.now().hour:
                    break

            wind_speed = get_wind_speed(breaks)
            averages = [sum(x) / len(values) for x in zip(*values)]
            humidity = averages[0]
            temperature = averages[1]

            crt_hour = datetime.now().hour  # VERY IMPORTANT
```

programul propriu-zis, care rulează întru-un ciclu infinit, începând cu citirea senzorilor, din minut în minut
apio, după 60 de minute, programul realizează o medie a valorilor obținute în ora respectivă

```python
            current_romania_datetime = datetime.now(tz=ZoneInfo("Europe/Bucharest"))
            current_datetime_string = current_romania_datetime.isoformat("T")
            data = [
                {
                    "time": current_datetime_string,
                    "value": temperature,
                    "dataType": "temp",
                },
                {
                    "time": current_datetime_string,
                    "value": humidity,
                    "dataType": "humidity",
                },
                {
                    "time": current_datetime_string,
                    "value": wind_speed,
                    "dataType": "wind",
                },
                {
                    "time": current_datetime_string,
                    "value": pressures[crt_hour],
                    "dataType": "pressure",
                },
            ]
```

în final, se crează fișierul json pentru post, și se realizează post-ul