Greenie Web Task

A Linux & Windows Command Line app that calculates average power consumption for a specific app without the background processes. App uses 2 test methods:

1. Instantaneous Test: Checks current usage of the app and evaluates it's approximate consumption.
2. Interval Based Test: Provide an time interval (in seconds) and conclude the test during this interval by collecting data at different intervals.

Usage

```
> pip install -r requirements.txt
> python main.py
```

Design Document

The idea is to use calibrated data for power consumption based on model of multiple components, check the usage percent of the selected app for each component, and multiply it with it's power model.

Approach:

- Save manufacturer data about some hardware components and use it to calibrate equations. (Database is required)
- If data wasn't present, we default to some approximated values. (Improvising)
- Understand how other parameters contribute to the power consumption such as hardware bus speed. (Testing & Observation is required)
- Tested app should be using resources to be able to calculate it's actual power consumption, or we will be calculating a theoretical value based on idle basis of the hardware. (This was concluded after testing)

Limitations:

- I don't have access to multiple hardware parts to conclude a generalized base model, So I had to improvise with approximating.
- Time span wasn't enough to conclude multiple hardware parts like disk usage, I/O, network usage, ..etc. so currently the app is calculating based on CPU & RAM usage.
- My access to hardware was based on a Laptop & a PC both based on Intel processor. So any testing mentioned below was based on observing same tests on both devices for Intel components only.

Testing:

- Windows: Most hardware monitor applications rely on win32 APIs on a low level to get hardware specifications and completes any missing data from its database with data serializing to extract relevant info.
- Power monitor tools aren't always giving the same readings at the same timings, readings had a margin of 0.5~5 Watts. (many screenshots were taken to make sure.)
- Relating CPU usage to it's TDP although not linear but gives very approximated results with the different power monitor tools readings. Reference P.7
- RAM power consumption is minimal and we can say the same across models. Reference
- My testing gave margin errors between 0.9~2.5W when measuring the overall usage for CPU & RAM and compare it to the readings with the power monitoring tools. (Used HWMonitor, TaskManager, HWInfo, PowerTop, CPU-Z)

- I've read many other articles and researches, only thing to conclude is that the app should be using resources before attempting to measure its consumption.
- RAM in idle state uses between 0.5~1 Watt to preserve its ON state and save current data. so an app in idle state won't be using resources. To attempt to measure, It should be using It.

Work:

- Days 1&2: just researching and evaluating equations and values using screenshots and calculations.
- Day 3: Taking notes and testing how to get required data on both Windows and Linux.

  > Accessing motherboard sensors isn't an easy task on Windows unlike Linux due to limitations of win32 API, so I sticked with the data that are guaranteed to be obtained on both kernels. Third party apps like PowerAPI weren't Windows compatible or not always guaranteed to get these data such as LibreHardwareMonitor. So I sticked with the data that are guaranteed to be obtained.

- Day 4&5: Designing the program flow and Developing the app.
- Day 6: Doing more tests to ensure cross-platform functionality & value approximations.
- Day 7: Writing the documentation and preparing the submission files.

Final Notes:
Such task needs more researching and hardware evaluation to conclude a guaranteed and specific power consumption measurement. Time span wasn't enough to implement more functionalities related to Integrated GPU or use S.M.A.R.T to analyze hard disks.