COSC2674/2755 (PIoT) Semester 2, 2023
Assignment 1 Specification

Marks allocated:	30 (worth **30%** of the total score)
Deadline:	Sunday 20.08.2023 11:59 pm
Submit via:	Canvas and GitHub
Work mode:	**individual**
Submission format:	**URL**

1.1 Aim

The aim of this assignment is to write a small IoT application using Raspberry Pi and Sense HAT in Python language. Some of the tasks of this assignment will require self-exploration and research, you will not find the answer in lectures notes and/or tutorials.

You must attend a demo session to get assignment 1 marked. A schedule and a booking document will be published soon. You must submit the assignment prior to demo. **No submission → No demo → No marks**.

1.2 Important | *please read this section*

You must adhere to the following requirements:

a. Only Raspberry Pi model 4 or 3 may be used.

b. You must use Python >= 3.5 to complete the tasks. Older versions must not be used.

c. Whether working individually or in a group, you must use *GitHub* for version control.

d. You must stick to the standard for Python: https://www.python.org/dev/peps/pep-0008/

e. Your Python code must be object-oriented. Procedural code to fetch *zero*.

f. Your files must be under correct folders, e.g., all files in Task A must be under a folder named "TaskA". Otherwise, 50% penalty will be applied.

g. In the demo session, the marker will ask you to simply modify your code logic, e.g., adding random numbers or functions, for testing purpose. Failed to satisfy the modification will introduce 20~80% penalty.

1.3 Tasks

Coding Tasks

[Task A] (10 marks) Create a folder named "TaskA". Create a JSON config file will store a temperature and humidity range. This file should be called `config.json`:

```
{
    "cold_temperature_upper_limit": "15",
    "comfortable_temperature_range": "15-25",
    "hot_temperature_lower_limit": "25",
    "dry_humidity_upper_limit": "30",
    "comfortable_humidity_range": "30-60",
    "wet_humidity_lower_limit": "60"
}
```

Create a python file called `THNotifier.py` which will log the current time, temperature, temperature category (i.e., cold, comfortable or hot according to `config.json`), humidity and humidity category (i.e., dry, comfortable or wet according to `config.json`) to a database every 10 seconds (you can choose what type of database to use: **SQLite** or **MySQL**). Please note that you must use Python and read category data from `config.json` rather than hard coding the numbers in your Python file. The data from `config.json` should also be verified with proper responses.

> *In this course we cover SQLite & MySQL, if you want to use another database, please feel free to do so. Use of Cloud databases is not allowed for assignment 1, you must use a local database installed on your Raspberry Pi.*

After saving to the database, display the temperature and humidity in LED matrix on Sense HAT in turns. The LED light should be refreshed every 5 seconds (for example, start with data temperature=15 and humidity=50 → display "T15" for 5s → display "H50" for 5s → read new data → display…). The color of LED should satisfy the following rules:

1. ***cold → grey***
2. ***hot → red***
3. ***dry → purple***
4. ***wet → blue***
5. ***comfortable → green***

[Task B] (5 marks) Create a folder named "TaskB". Read data from the database in **Task A**, and use 2 different Python data visualisation libraries to create 4 images (e.g., png files). This is where it gets interesting, and you will need to do some research of your own.

You will now need to think:

- With the data that you have in the database, what kind of data visualisation graph should be used? (Scatter plots, Bar charts and Histograms, Line plots, Pie charts, etc.)
- What should be represented in the above graph? Why did you make that decision?
- Python supports numerous data visualisation libraries. Which two libraries did you narrow it down to?

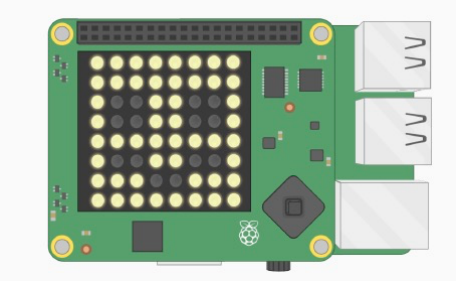
Create a python file called `analytics.py` that creates the above-mentioned images.

In addition to creating the images write a text file called `analytics.txt` comparing the different techniques used:

- Questions stated above and,
- an analytical comparison of the libraries used, advantages/disadvantages, simplicity/complexity, flexibility/configurations, anything you may think is necessary,
- Formal references if needed.

Use of JavaScript or any other language is prohibited for this task. You are only to use Python.

[Task C] (10 marks) Create a folder named "TaskC".
(1) Create a python file called `animatedEmoji.py` which will display 5 Emoji faces in the LED matrix on Sense HAT. You need to create your own Emojis with at least 3 colours for each. You also need to display them one by one with 2 seconds interval.



(2) Create a python file called `shakingEmoji.py` which will display one of the Emojis created in (1) in order when shaking the Pi. In this task, you need to be able to detect shaking motion by sensor and then display/refresh the Emoji.

Professionalism and Teamwork-related tasks

[Task D] (5 marks) Professional use GitHub for version control and how the code has been developed over time. Please read the assignment rubrics for details.

- You must use object-oriented python code (*procedural code will fetch a **zero** for the whole of assignment*).
- Note: You can add more python files in addition to the ones mentioned above if necessary.
- Note: Your code will be marked for its adherence to the PEP8 style guide for Python code.
- Note: necessary comments must be included in your files.
- Note: necessary information, such as student id and name, must be included in README file.

1.4 What and how to submit?

Submit your GitHub repository URL in Canvas. The repository should contain all the files:

- `config.json`
- `THNotifier.py`
- `analytics.py`
- `analyticsReport.txt`
- `animatedEmoji.py`
- `shakingEmoji.py` and,
- any other files that you may have created.

1.5 Late submission and Extension

a. A penalty of 10% per day (*which is 3 marks per day in case of this assignment*) of the total marks will apply for each day late, including both weekend and weekdays.

b. Extension requests should only be emailed to the lecturer (xiaoyu.xia@rmit.edu.au) before your (extended) deadline.

1.6 Plagiarism

All assignments will be checked with **plagiarism-detection software** and **AI detect tools**; any student found to have plagiarised will be subject to disciplinary action. Plagiarism includes:

- Submitting work that is not your own or submitting text that is not your own.
- Allowing others to copy your work via email, printouts, social media, GitHub, etc…
- Posting assignment questions (in full or partial) on external technical forums.
- Copying work from/of previous/current semester students.
- Sending or passing your work to your friends.
- Posting assignment questions on technical forums to get them solved.

A disciplinary action can lead to:

- A meeting with the disciplinary committee.
- A score of zero for the assignment.
- A permanent record of copying in your personal university records and/or
- Expulsion from the university, in some severe cases.

All plagiarism will be penalised. There are no exceptions and no excuses. You have been warned.