User Guide

This User Guide details the user about how to use the software provided for use with the Raspberry Pi with Temperature, Humidity, Air Pressure and Particulate sensors.

Contents

What is this product? 2
Accessing the web app 2
User Interface 2
The meaning of each value in the tables, graphs and gauges? 8
Installation 10
Common Q&As 10
References 11

What is this product?

This product consists of a Raspberry Pi (RPi) that has a full suite of sensors attached to its headers. These sensors consist of an Adafruit DHT22 Temperature and Humidity Sensor, a Honeywell or a Nova SDS-011 particulate sensor and an Adafruit BMP180 for temperature and air pressure readings. These are all included on the sensor, creating a small package that can be placed indoors or in a place that is protected from the weather (the sensor node is not waterproof).

Included in this package are various Python files that help you connect your RPi to the internet and send data collected to a DynamoDB table in Amazon Web Services (AWS). The data can then be viewed via a web app that shows various graphs, tables and gauges to display the data.

Multiple sensor nodes can be attached to the project, meaning you can observe multiple areas at once and have them all send data to a central database to look at.

This product is recommended for families who want to observe air quality in their home including low levels of dust and smoke pollution that cannot be seen eg from cooking or log fires.

Accessing the web app

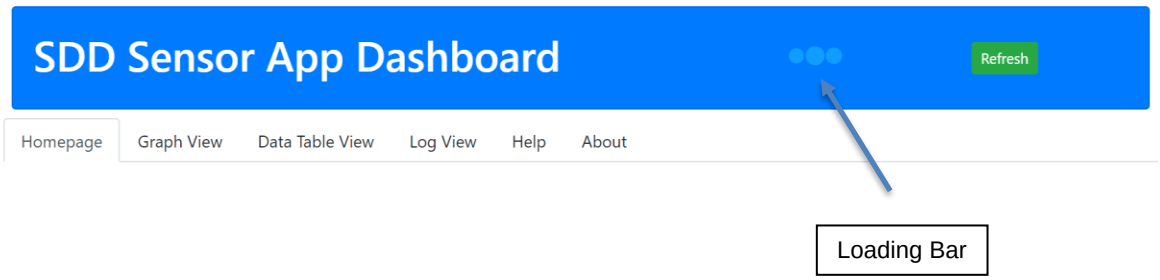
This app can be acced with the URL http://sensorapp.ap-southeast-2.elasticbeanstalk.com/. You will be greeted with an authentication box. The username and password can be obtained from the creator.

User Interface

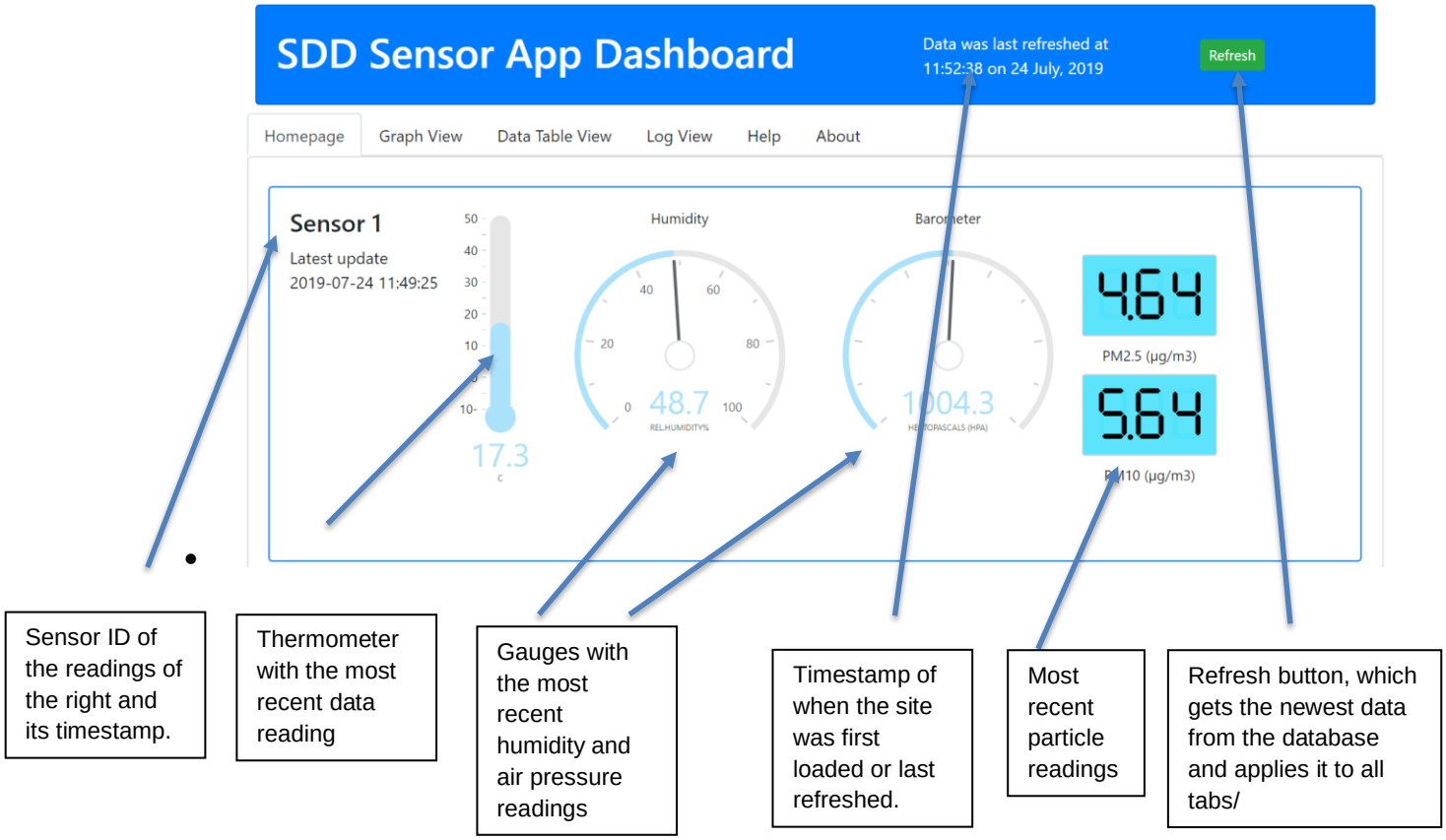
- When the web app first loads, you will see the following message for a few seconds:

Loading...

- After that, the web app header will appear and it will continue to load data from the database. The loading of data is indicated by the three pulsating circles in the loading Bar.



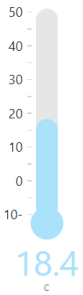
- When the site has fully loaded, you will see a page that contains sections for each sensor node that look like this:



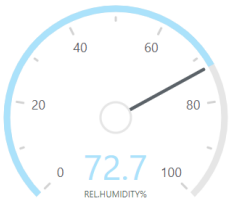
- Further Homepage view with other sensors readings at a glance.

Sensor 2

Latest update
2019-07-04 13:06:31



Humidity



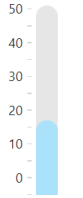
Barometer

PM2.5 (µg/m3)

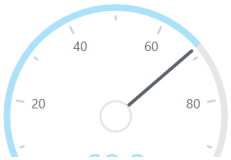
PM10 (µg/m3)

Sensor 3

Latest update
2019-07-24 11:50:54



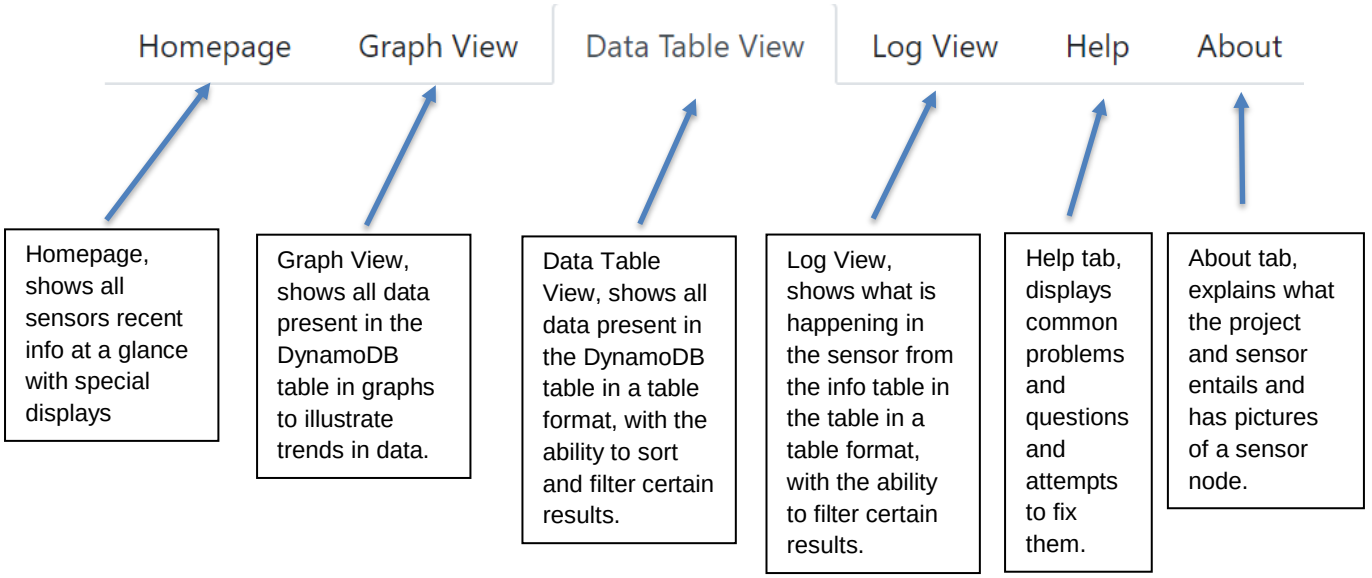
Humidity



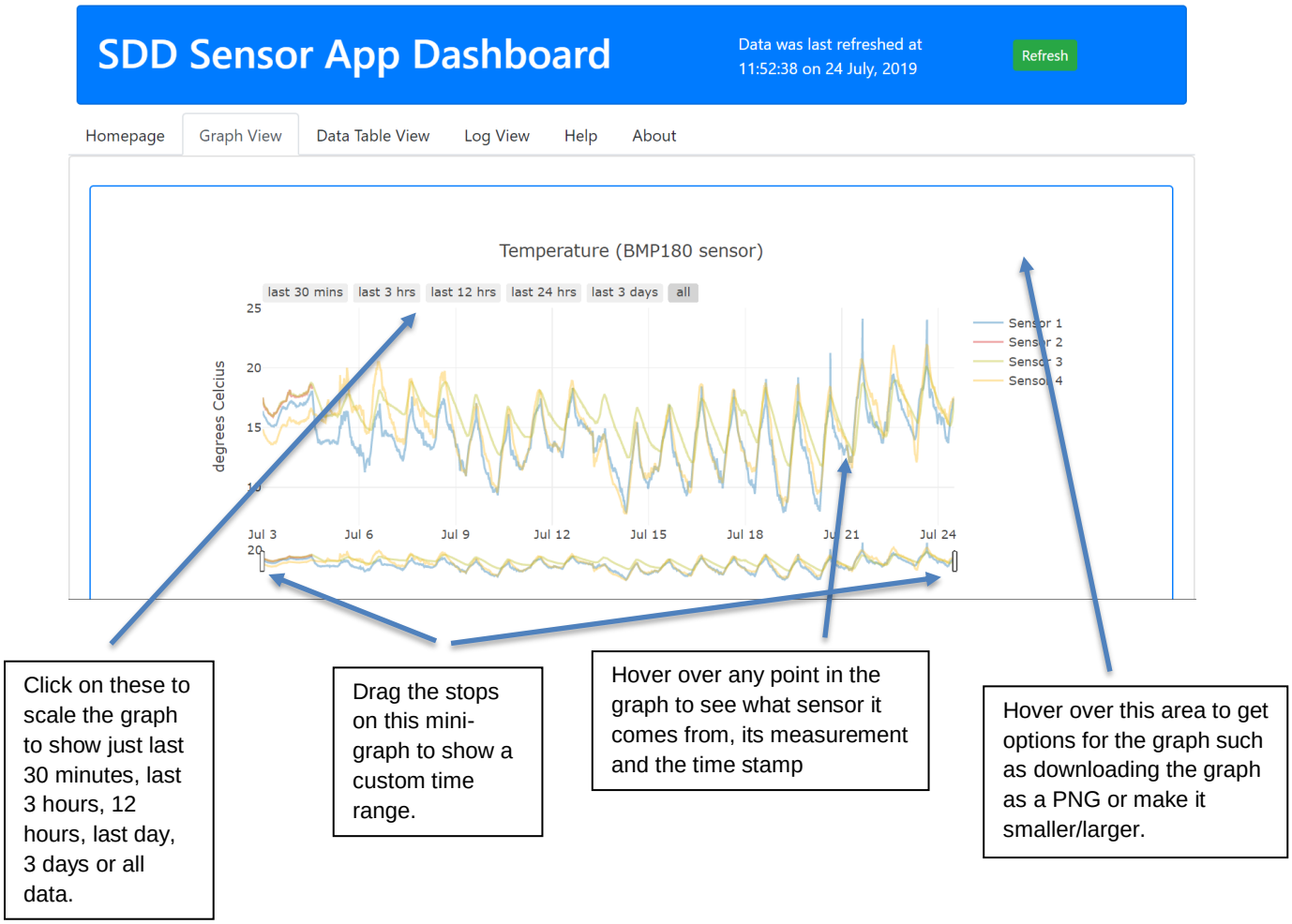
Barometer

PM2.5 (µg/m3)

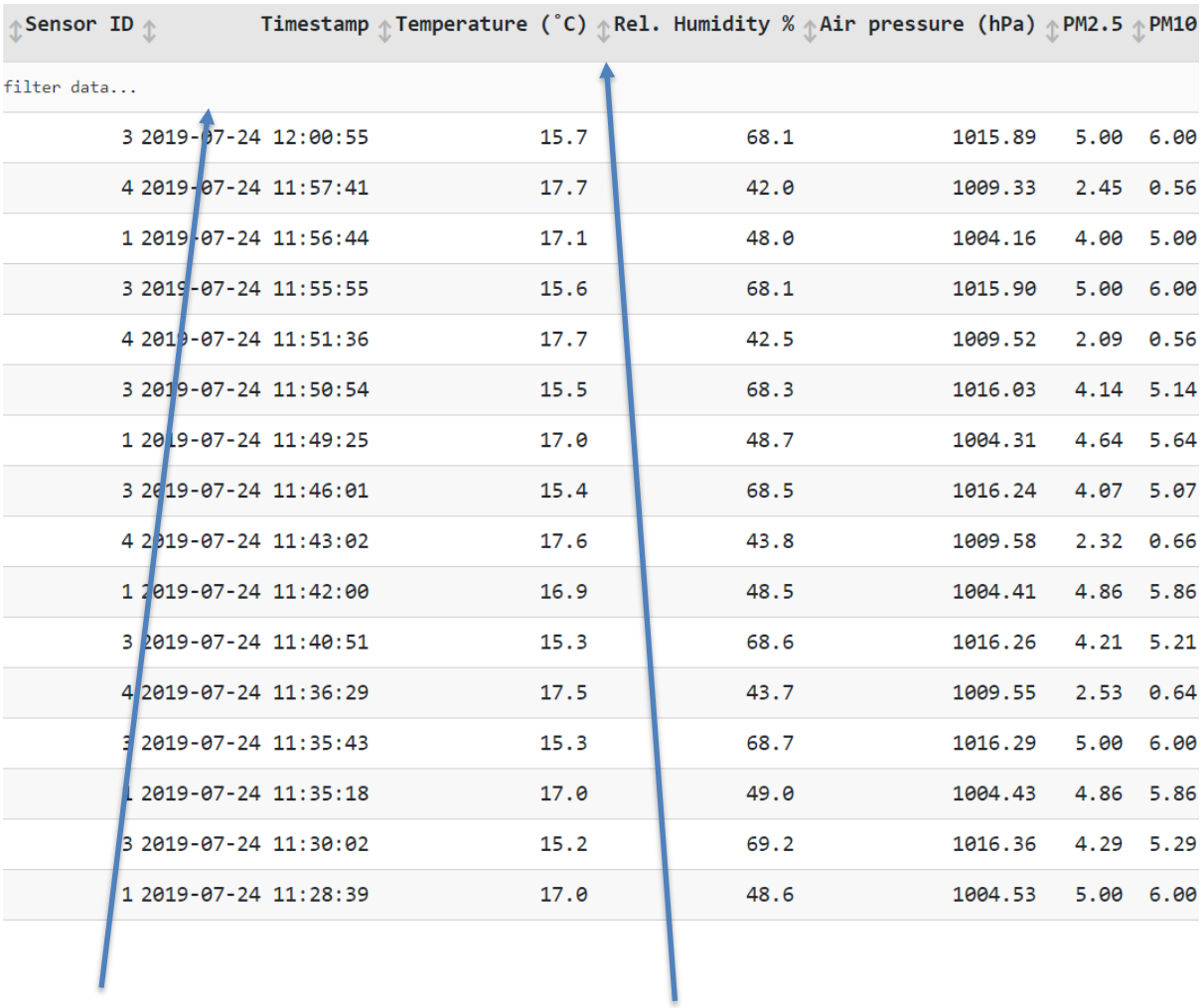
- The Tabs, below the header bar are used to swap to the different views, help file and about file.



- Graph View, shows all data present in the DynamoDB table in time-series graphs to illustrate trends in data.



- The Data Table View, shows all data present in the DynamoDB table in a table format, with the ability to filter results and sort by columns.

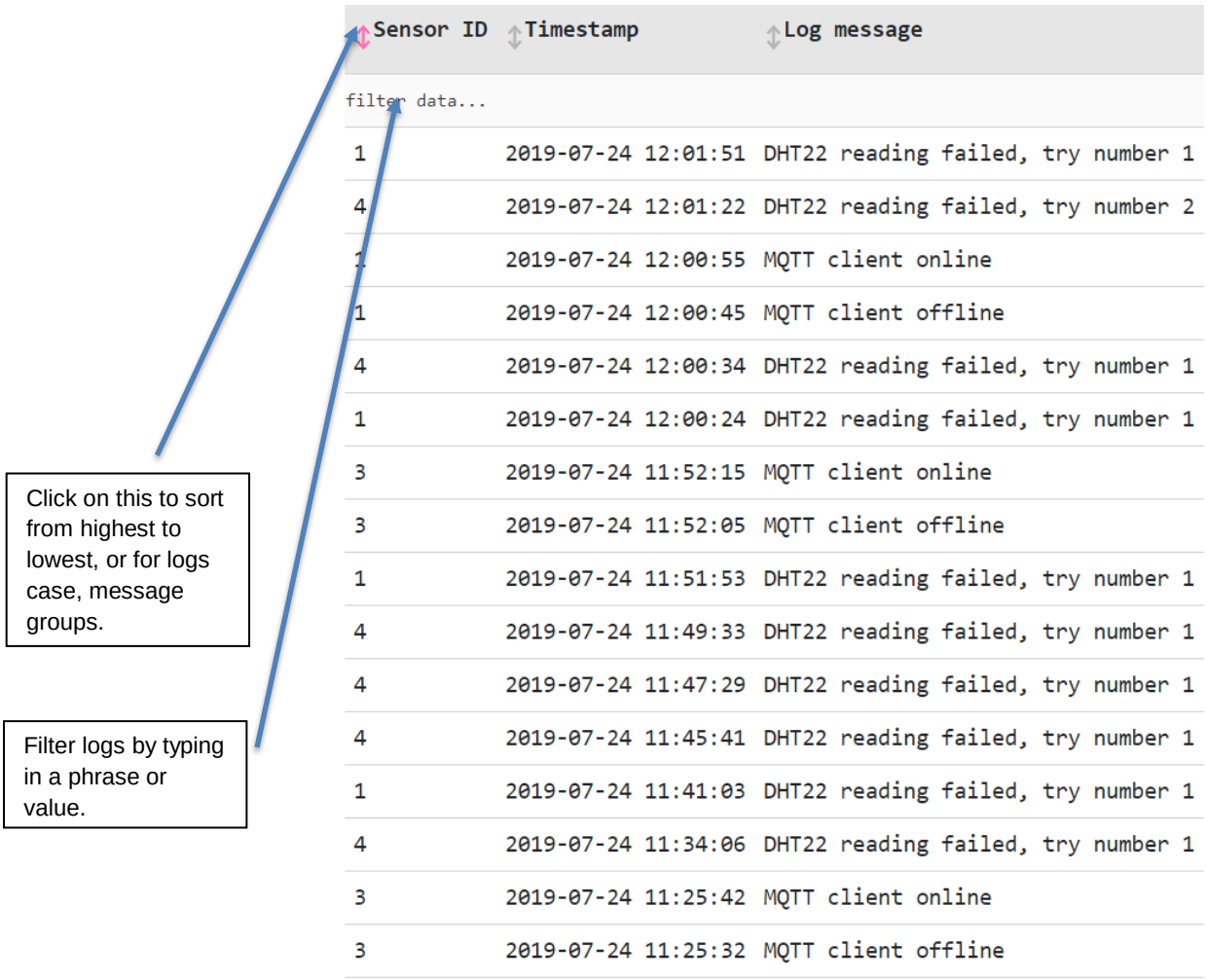


Sensor ID	Timestamp	Temperature (˚C)	Rel. Humidity %	Air pressure (hPa)	PM2.5	PM10
filter data...						
3	2019-07-24 12:00:55	15.7	68.1	1015.89	5.00	6.00
4	2019-07-24 11:57:41	17.7	42.0	1009.33	2.45	0.56
1	2019-07-24 11:56:44	17.1	48.0	1004.16	4.00	5.00
3	2019-07-24 11:55:55	15.6	68.1	1015.90	5.00	6.00
4	2019-07-24 11:51:36	17.7	42.5	1009.52	2.09	0.56
3	2019-07-24 11:50:54	15.5	68.3	1016.03	4.14	5.14
1	2019-07-24 11:49:25	17.0	48.7	1004.31	4.64	5.64
3	2019-07-24 11:46:01	15.4	68.5	1016.24	4.07	5.07
4	2019-07-24 11:43:02	17.6	43.8	1009.58	2.32	0.66
1	2019-07-24 11:42:00	16.9	48.5	1004.41	4.86	5.86
3	2019-07-24 11:40:51	15.3	68.6	1016.26	4.21	5.21
4	2019-07-24 11:36:29	17.5	43.7	1009.55	2.53	0.64
3	2019-07-24 11:35:43	15.3	68.7	1016.29	5.00	6.00
1	2019-07-24 11:35:18	17.0	49.0	1004.43	4.86	5.86
3	2019-07-24 11:30:02	15.2	69.2	1016.36	4.29	5.29
1	2019-07-24 11:28:39	17.0	48.6	1004.53	5.00	6.00

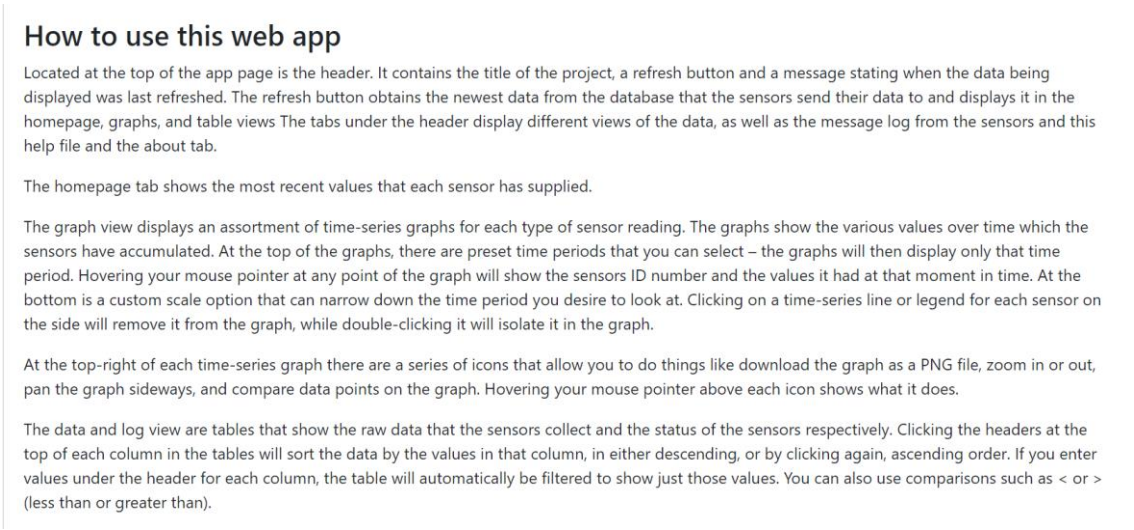
Type something in here to show results relating to the phrase (i.e. type 1 in Sensor ID column to get all sensor 1 results). Must follow the format in the table, so for timestamp it must be YYYY-MM-DD HH:MM:SS. You can also use > and < (greater than and less than) operators, or >= etc.

Click on these arrows to sort the data from highest to lowest. Click again to reverse the order.

- The Log View shows what is happening on each sensor node based on messages sent to the info table in the central database. This information may be useful in troubleshooting problems or checking for issues in each sensor. Like the data view table, you can filter and sort the log view.



- The Help tab, displays a condensed version of this user guide as well as common questions and answers and some useful troubleshooting information.



How to use this web app

Located at the top of the app page is the header. It contains the title of the project, a refresh button and a message stating when the data being displayed was last refreshed. The refresh button obtains the newest data from the database that the sensors send their data to and displays it in the homepage, graphs, and table views The tabs under the header display different views of the data, as well as the message log from the sensors and this help file and the about tab.

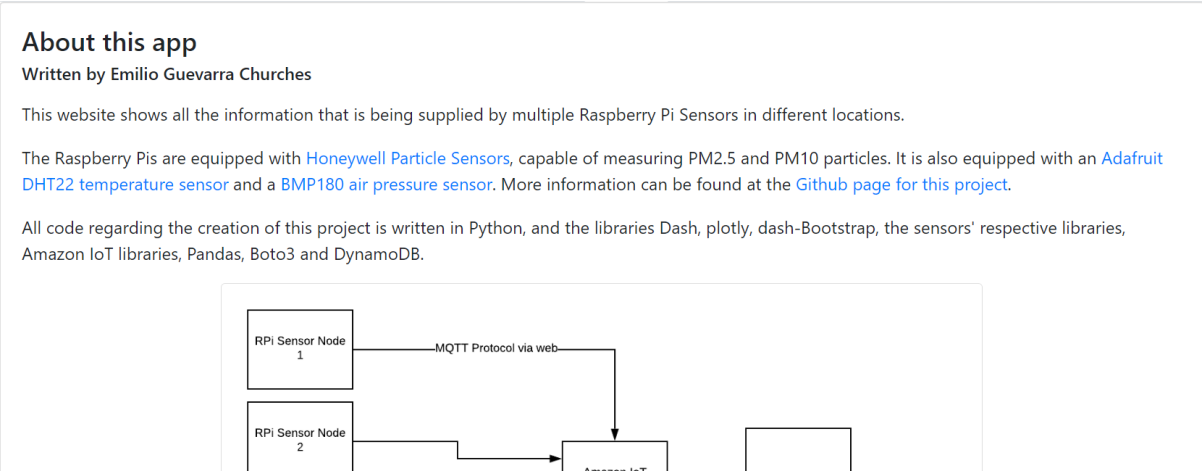
The homepage tab shows the most recent values that each sensor has supplied.

The graph view displays an assortment of time-series graphs for each type of sensor reading. The graphs show the various values over time which the sensors have accumulated. At the top of the graphs, there are preset time periods that you can select – the graphs will then display only that time period. Hovering your mouse pointer at any point of the graph will show the sensors ID number and the values it had at that moment in time. At the bottom is a custom scale option that can narrow down the time period you desire to look at. Clicking on a time-series line or legend for each sensor on the side will remove it from the graph, while double-clicking it will isolate it in the graph.

At the top-right of each time-series graph there are a series of icons that allow you to do things like download the graph as a PNG file, zoom in or out, pan the graph sideways, and compare data points on the graph. Hovering your mouse pointer above each icon shows what it does.

The data and log view are tables that show the raw data that the sensors collect and the status of the sensors respectively. Clicking the headers at the top of each column in the tables will sort the data by the values in that column, in either descending, or by clicking again, ascending order. If you enter values under the header for each column, the table will automatically be filtered to show just those values. You can also use comparisons such as < or > (less than or greater than).

- The About tab, explains what the project is about and how the sensor nodes send data to the database and then to the web app.



The meaning of each value in the tables, graphs and gauges?

- Temperature, defaulting to the values obtained by the Adafruit BMP180 sensor is shown in the Temperature column in Data View, the Temperature graph and the Thermometer on the Homepage. Temperature refers to the air temperature in the vicinity of the sensor node, measured in °C.
- Sensor ID, refers to the internal number of the sensor node that is providing the information.
- Humidity, obtained with the Adafruit DHT22 sensor, is shown in the Rel. Humidity column in Data View, the Rel. Humidity graph and the Rel. Humidity% gauge on the homepage. Humidity is the amount of water vapour present in air. Measured in Relative Humidity %, which adjusts for the air temperature (which is also measured by the DHT22 sensor but is not displayed separately in the web app, just used by the DHT22 driver library to calculate relative humidity)
- Air Pressure, also known as atmospheric pressure or barometric pressure, is obtained with the Adafruit BMP180 sensor and is shown in the Air Pressure column in Data View, the Air Pressure graph and the Air pressure gauge on the Homepage. Measured in hectopascals (hPa)
- PM2.5 and PM10, obtained with the Honeywell or the Nova SDS-001 particulate sensor and is shown in PM2.5 and PM10 column in Data View, the PM2.5 and PM10 graph and the PM2.5 and PM10 counters respectively on the Homepage. PM is the measure of particulate matter in the air, with 2.5 and 10 referring to the size of the particles, being in micrometres. The measures are in micrograms of particles per cubic metre of air.

- Timestamp, Timestamp refers to when the data was sent to the server by each sensor node. It uses the time zone that was set on each sensor. In this project, all sensor nodes were set to the Sydney time zone.
- Log Message, shows any problems or events that are currently happening with the sensor. However, if a problem on the sensor node prevents it from sending data, then the problems obviously won't show up in the Log table.

Installation

Please refer to the installation guide on either the GitHub (https://github.com/emiliog-c/SDD-Sensor-App) or in the directory that you are reading this in for help.

Common Q&As

Q: Where should I place the sensor?

- **A:** Ensure you place your sensor in a wide open space to allow it to be used at maximum effectiveness. Placing it outside if you want to know the outside temperature and air quality would be a good choice. Placing it inside is also a good choice. However, the sensor nodes are NOT waterproof and MUST be placed where they will not get wet. They also need 5V USB power so they need to be near a power socket, and they need to be in range of the wi-fi network. For these reasons, it is recommended to place them under cover on verandahs or patios is they must be outside. Really the sensors are designed for indoor use.

Q: Is it safe for the sensor node just to be turned off?

- **A:** In general, yes. Ideally the sensor should be shut down using a linux command (in the linux terminal console, type “sudo shutdown -h now”) before the power is removed, but in practice it should survive just having the power turned off OK.

Q: If the sensor disconnects from the internet due to an outage, will it reconnect and what will happen to the data collected when it was offline?

- **A**: It should automatically reconnect, and the data will have been stored locally until it can be sent to the cloud, which it will do automatically when it reconnects. However, if the sensor loses power before it reconnects, the stored data will be lost because it is only held in RAM memory on the RPi.

Q: How do I know if the sensor is down?

- **A**: Inspect the home page of the dash app to see when the sensor last sent data. The Log View tab can be used to see if there are any log messages from the sensor which might indicate what is wrong, although if the wi-fi connection for the sensor is down, then log messages cannot be sent either.

Q: If I want to assign a sensor to a new DynamoDB table, how would I do that?

- **A**: Follow the installation guide regarding the setup of the DynamoDB table.

Q: What do I need to run this sensor?

- **A**: Python version 3.6 or later, the libraries that need to be installed detailed in the installation guide, an Amazon Web Services account and a working internet connection.

Q: What's the difference between the sensor_run.py file and the sensorDashApp.py file?

- **A**: The sensor_run.py file runs the sensors on the sensor Raspberry Pi devices and sends all the information via MQTT to the DynamoDB tables made with AWS services. The sensorDashApp.py file takes the data from the DynamoDB tables, places it in graphs and presents it as a website.

If there is a problem, refer to the help file.

References

Libraries used: boto3, jsonschema, numpy, pandas, Flask, Flask-Compress, Flask-Cors, Flask-SeaSurf, dash, dash-auth, dash-bootstrap-components, dash-core-components, dash-daq, dash-html-components, dash-renderer, dash-table, plotly, dynamodb-json

Resources Used:
https://dash.plot.ly/
https://dash-bootstrap-components.opensource.faculty.ai/
https://aws.amazon.com/developers/getting-started/python/
https://aws.amazon.com/sdk-for-python/
https://learn.adafruit.com/dht/connecting-to-a-dhtxx-sensor