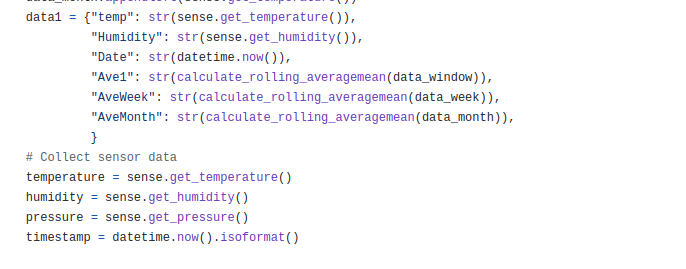
Overview:

This project was ideally suited for me, I have a passion for the environment and it also gave me a great opportunity to learn about some aspects of IOT that i wasn’t very knowledgeable about e.g. Mqtt and also work on improving my ability to write programs using Python.

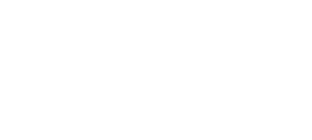
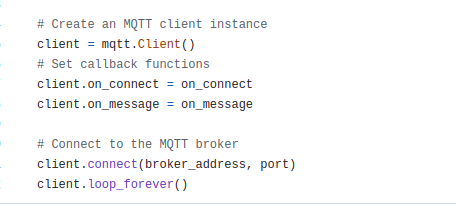
When we received Pauls datafile the first thing I did to it was correct it. I converted it from excell to CSV format and then i used Python code to combine the date and time collumns into a single date-time collum. The Python file is in the folder attached (datetime1.py).

**Use and develop sensors and actioners**

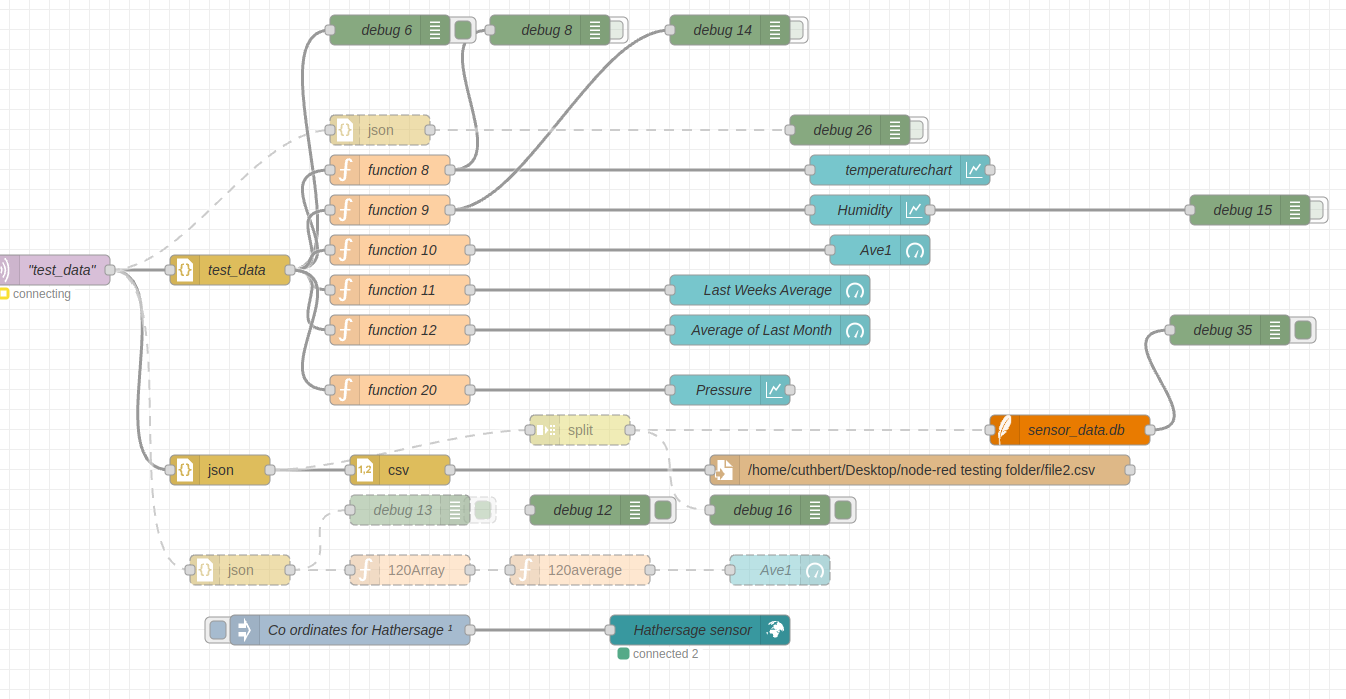
Below is a screenshot of the code in my program that collects the sensor data from my Raspberry Pi:

**Communication:**

Below is a screenshot of the code my program uses to communicate with my other raspberry pi via mqtt:



I have also used node-red on my other raspberry pi to subscribe to the mqtt broker and get display data being published by the raspberry pi.



**UX Design:**

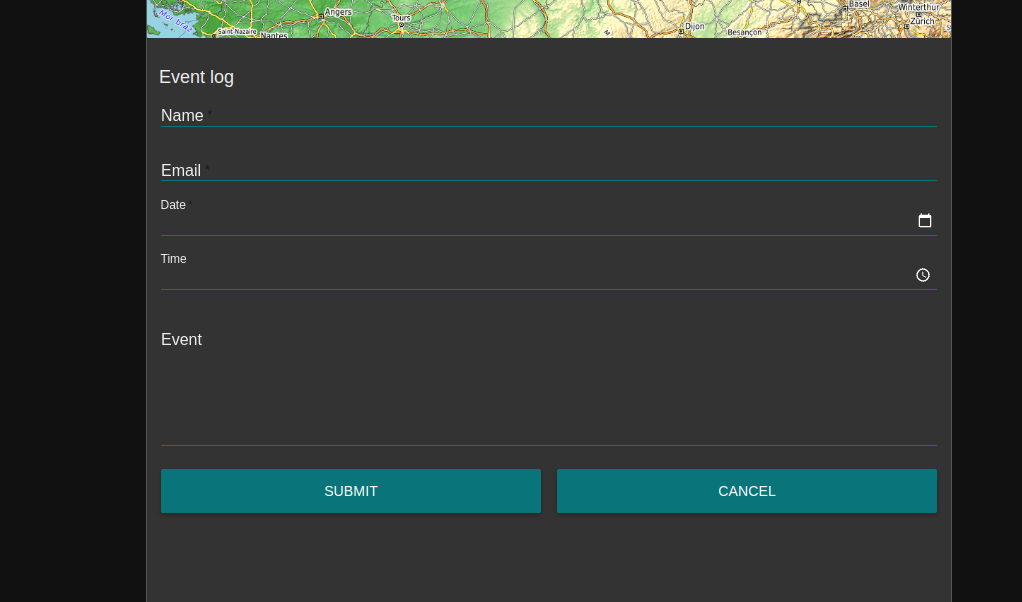
I took notes and sketched diagrams of designs of dashboards and visualized my idea of our systems’ architecture to show you so I can ask you about a potential technical issue I could foresee. I would break down every big portion of the project into it’s constituent incremental tasks that I could then work out how to solve each task one by one. So I started by collecting the data from the pi sensehat, using it to do calculations to get the rolling averages, formatting it to save to CSV file and save into an sql database and only then sending to be displayed on my node-red dashboard. Next stage i split that python program into two parts, one to collect the data the other to work with it but both were on the same raspberry pi so it was all local host, then i put one of them on my other pie and eventually I had node-red running on the other pi but subscribing the topic from the mqtt broker on where the data was being collected.

**UX testing:**

Unfortunately the people in my team weren’t very inspired by the project so they never showed much interest in checking out my programs or my node-red dashboard etc...

But I’ve got something ***far*** better than a few disinterested students! My friend works for the wildlife trust on flood management (among other things) so I’ve sent him a video of the data collection and the dashboard displaying it. I’ve talked to him about how useful this kind of system would be for him in his job and he was very impressed. In my code I have an “if” statement that will send a warning message if the latest data point is significantly different from the last one so I asked my friend what the level of change would he want to receive a warning message and he told me that there are loads of different things that he would want a warning about a variety of different things each would be slightly different e.g. compare a heatwave to a neighboring field being fertilized?

So I’ve added an event logging feature that will allow users to input things they have seen in the field and help us understand the different data that the environment is giving us:



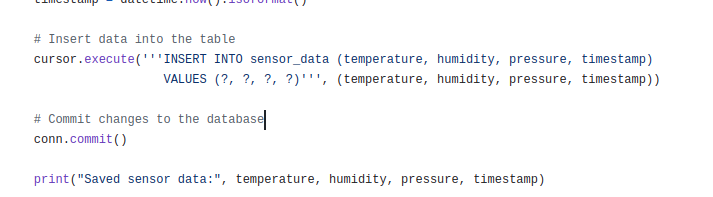
I showed my program and dashboard in action to my fellow IOT students recently and Sadaf gave me some valuable feedback to trim the number of decimals in the data I was displaying. So I immediately found a function to “truncate” the floating point numbers, I customized it to suit my own purposes and applied it to all the data as it was being processed by my python code and now the display is a lot easier on the eye, she also advised me to work on the layout of the dashboard as well.

**Use and develop data collection and processing:**

This was my specialization, i started by processing Paul’s datafile by correcting the date and time columns into a single correctly formatted into an iso compliant date-time collum. I discovered a special type of “double ended array”, I’d tried to do some calculations using node-red until I got to a dead end but that didn’t stop me, I didn’t give up. i always knew that writing a Python program was the ideal way to do the calculations required but to be able to use python I needed to learn about mqtt servers and how to convert python datatypes into JSON objects. Working with the data was a joy, I discovered the python object “deque” and immediately asked everyone i know who works with python if they have ever used a “deque” but nobody had ever heard of it! I worked out the correct syntax for it, I found an innovative way of testing it using an extension to VS code called “jupyter” (it shows every variable in your code. I then set the python to input a fixed number into the deque so i would be able to easily calculate what the rolling average should be at any time and check the deque was working as intended. Collecting the data was also straightforward for me because I’ve done it before. The greater challenge was always reliably connecting to the mqtt broker, subscribing to the topic, converting the data into a JSON object/string then catching that JSON object and unconvert it at the other end. It was essential for the publishing/subscribing to be done efficiently, reliably and precisely or else there would be a backlog or errors preventing the data getting to the end user or if the data turned up at unexpected times then the deque wouldn’t fill up correctly and there would be a loss of data integrity. Also there is no point saving the data to CSV file if it’s a complete mess so it has to be one row at a time and ordered correctly each time.

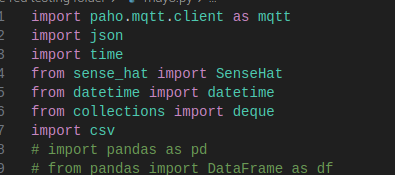
**Persistence layer:**

This project is the first time since I started at Hallam where I have made an intentional effort to store each version of my program so that anyone can see the development process. I’ve been actively using my github account to save each version online and in the cloud. I saved the data to csv file using my python code and also using my node-red flow as well – i wanted to proof that node-red could do the job reliably before i recommended it as an option. I even found a way to decode and reformat each JSON string so I could save the data into an sql database, this is the code that did that job:

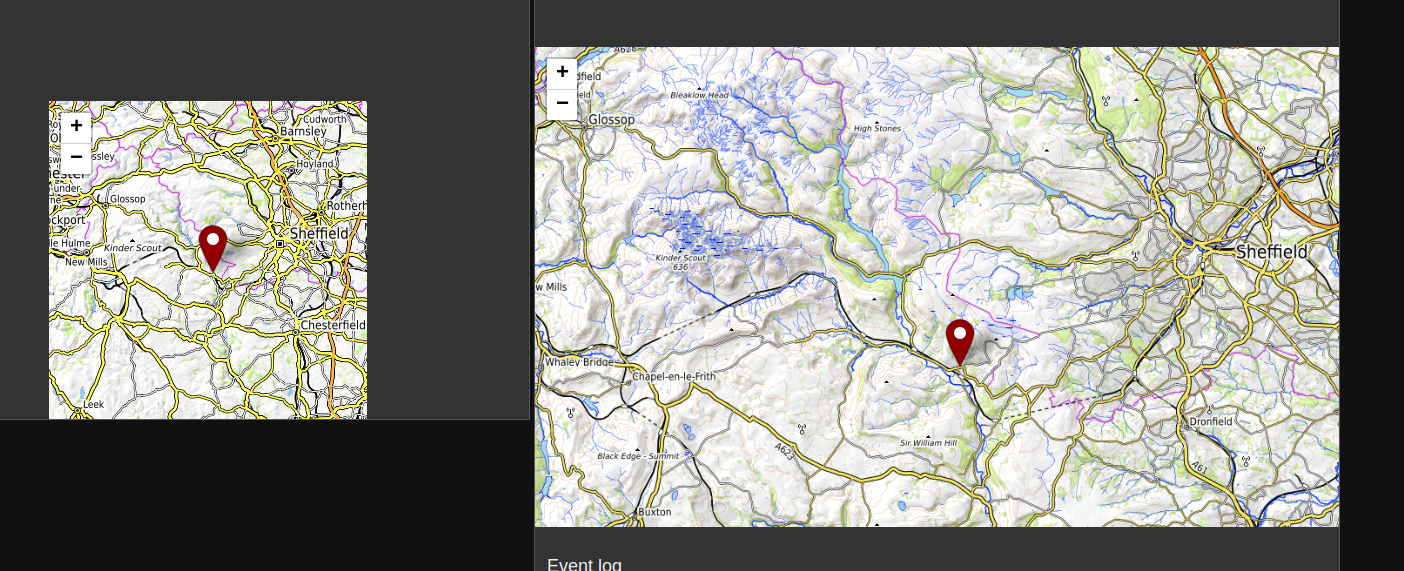
 I’m also working out how to use node-red to save to an SQL db but i haven’t done enough testing on it to rely on it yet – I have saved stuff to a db but if the data isn’t saved efficiently, reliably and predictably then it’s lost it’s integrity and therefore not good enough for prime time.

**Use APIs or external libraries:**

These are some of the external libraries I’ve been using

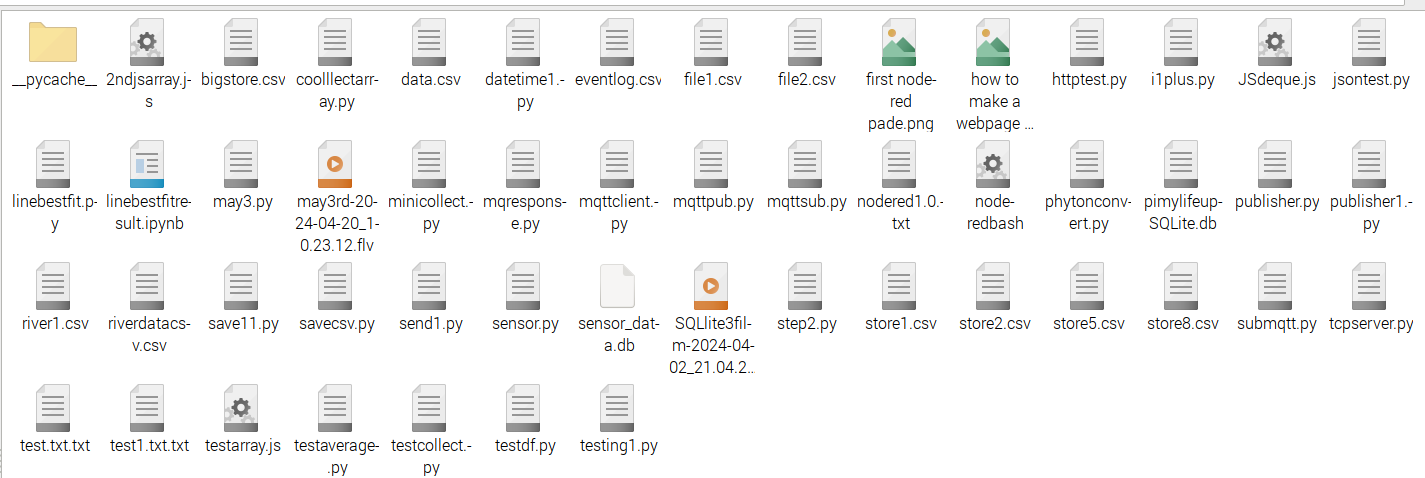


This a screen shot from my interactive map API:



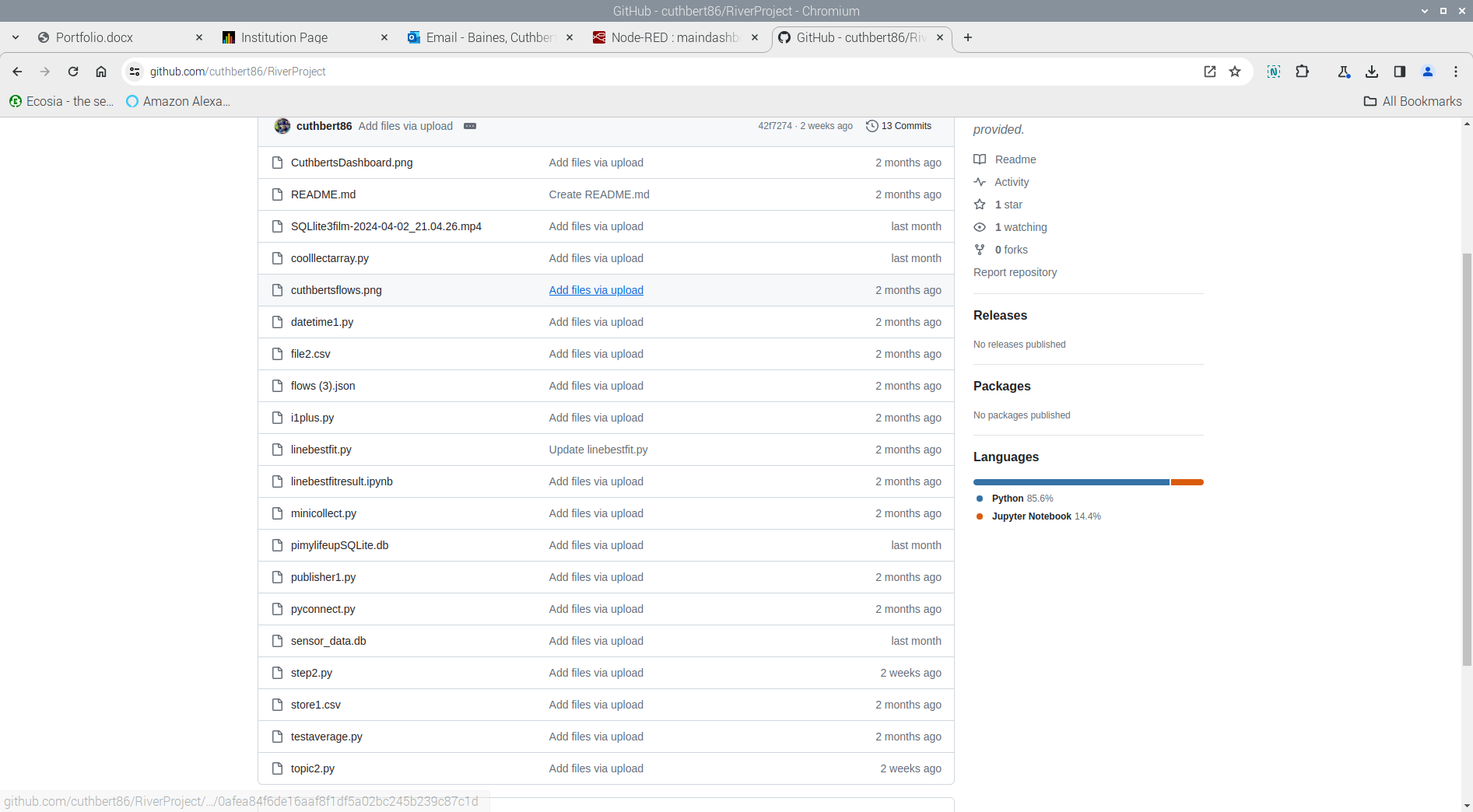
**Version control & backup:**

I’ve kept the old versions of my programs so I can show the “development process”:



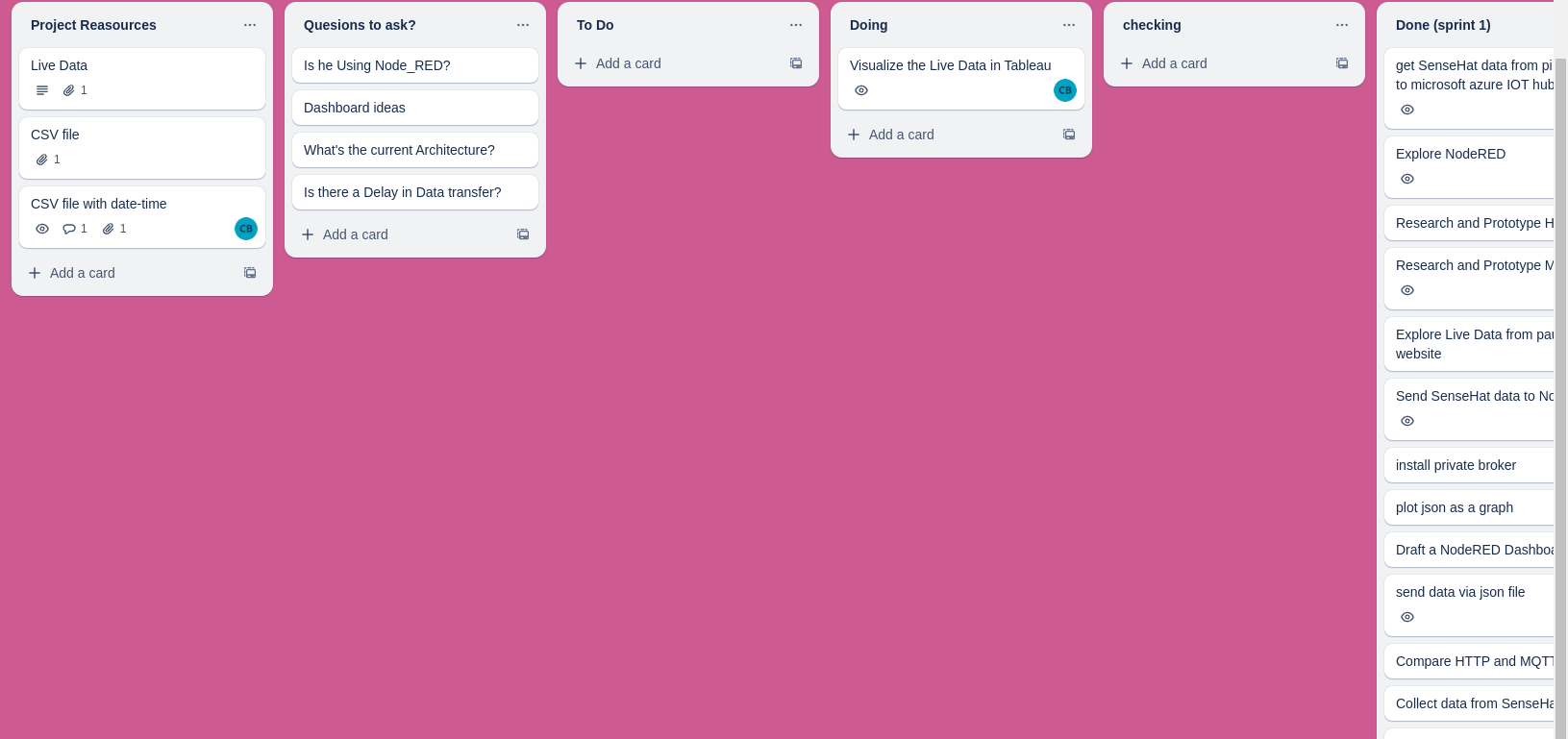
I’ve used github a lot during this project to ensure I’ve always got an online back up.

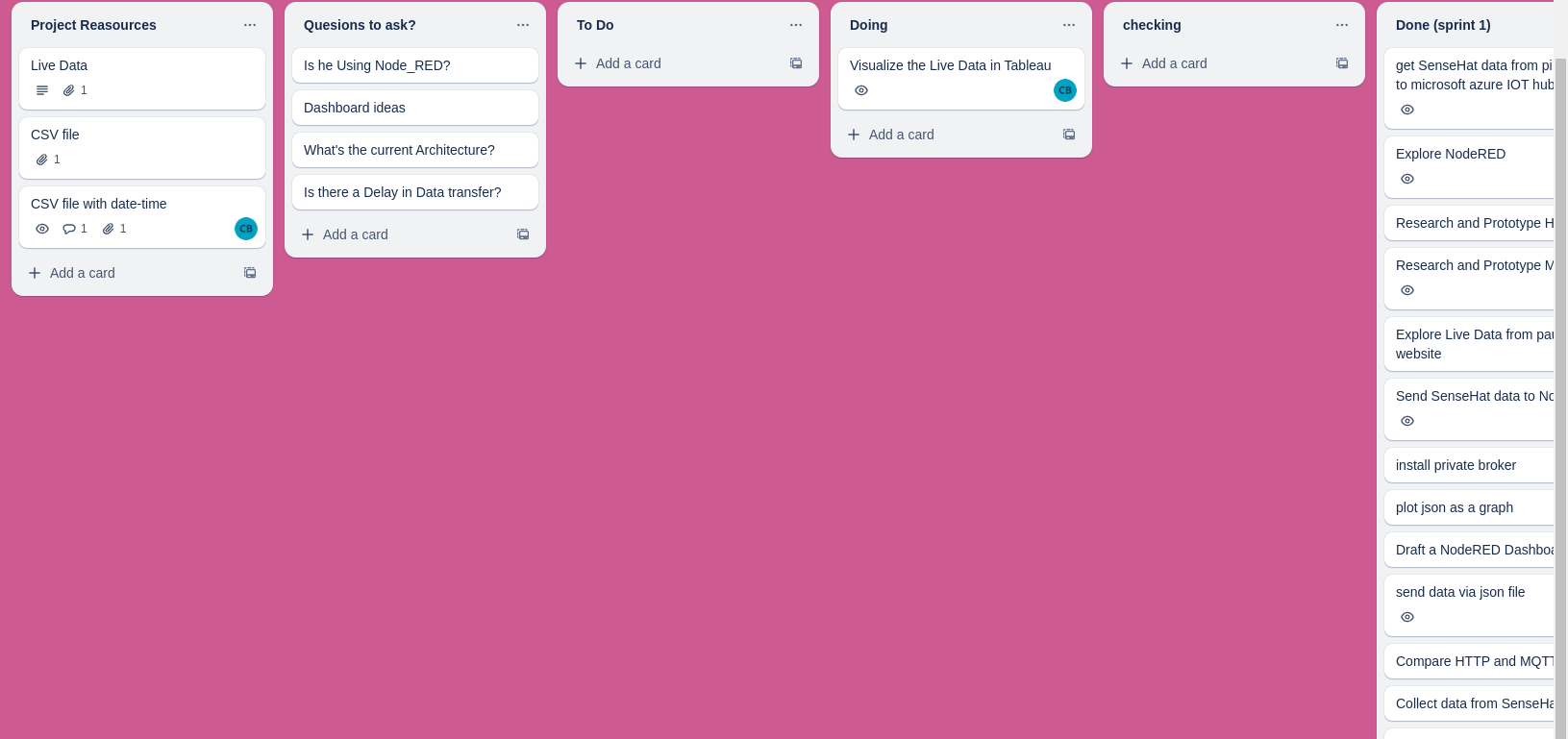
https://github.com/cuthbert86/RiverProject.git



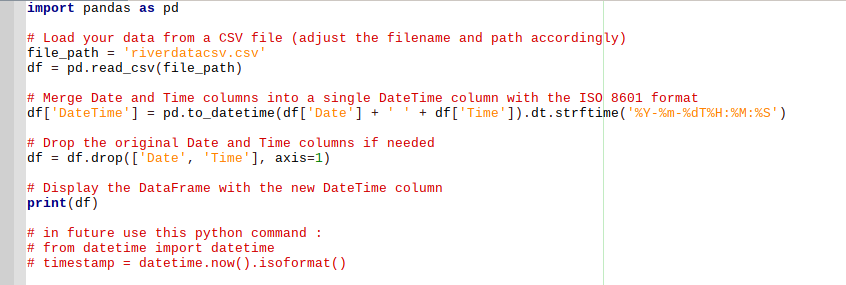
**Agile with team:**

I’m not going to pretend it was easy working with this team, unfortunately they weren’t as inspired by this project as I was and I was specifically told to focus on my area of expertise and not take control of the group. However I did use Trello to keep the group updated about what I was working on:





I did some useful work that the group has benefitted from, I fixed the data file Paul gave us by merging the date and time collum into a single date-time collum which I made available to everyone via Trello. I know that Jack has used that file a lot in his work.



**Agile with Client:**

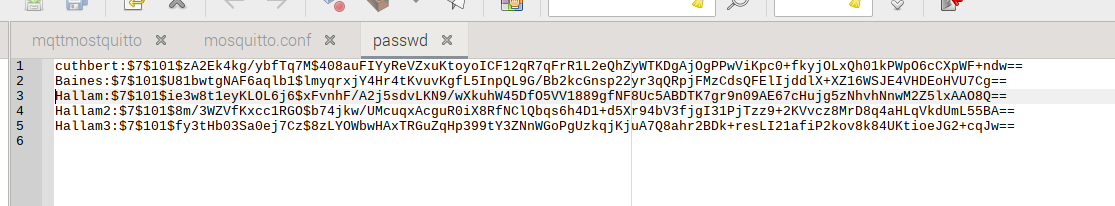
Every time we met with our client I was always the most well prepared member of the team. I always turned up with a notepad full of questions I’d prepared in advance and was the only person in the meeting who took notes. I think our client was specifically vague with his answers because if the question was about biology then that wasn’t his expertise so he didn’t want to give us an incorrect answer and if the question was a technical one he wanted us to use our own imagination to come up with new ideas. So he’s kind of ended up with what he wanted, a collection of innovative ideas that haven’t been refined because we never got the answers to the questions about the biology.

Unfortunately I didn’t meet him enough to get enough feedback to make my system polished enough to be deployed.

**Security:**

There are a number of security features that can be implemented:

1. The node red dashboard could be protected at the front end by using something like django and making the clients sign in to access the data.
2. Typing this into the config file will ensure only those with a username and password will be able to access the mqtt broker: “allow\_anonymous false” “password\_file /etc/mosquitto/passwd”.
3. Then create usernames and passwords:



1. By using port 8883 that will enable you to use SSL which is a more secure way to send your data.
2. It is also possible to encrypt the data using this python code:

from cryptography.fernet import Fernet

cipher\_suite = Fernet(key)

msg = cipher\_suite.encrypt(message.encode())

**Flaws**

1. Never got to see an example of anyone else’s portfolio so this could be amazing or not good enough and I won’t know until I get my results.

2. Not enough feedback: Everytime i got feedback they told me something really obvious that I’d missed so i made the corrections immediately.

3. Not enough contact with our client: While I was in the group I was told not to take control of the group so contacting our client without the entire group agreeing to it. Without contacting our client more regularly there was no way for me to produce something he would really appreciate.