

## AS91906, AS91902, AS91903 - L3 Programming Project

[AS91906](#) v1 - 6 Credits - Use complex programming techniques to develop a **computer program**

[AS91902](#) v1 - 4 credits - Use complex techniques to develop a **database**

[AS91903](#) v1 - 4 credits - Use complex techniques to develop a **digital media outcome**

**Due: End of Term 2**

I attest that the work submitted is my own. Any media or code from other sources has only been used to support my work, used with permission and has been fully attributed.

**Name:** Jack Edwards

1. In a separate tab in Chrome, log into your school account ([portal.office.com](https://portal.office.com)), then click “Google Apps” under the waffle in the upper left.
2. Refresh this page (**F5**) then click **File** → **Make a copy ...**
3. Then give your document the title: **LastName, FirstName - AS91883 Develop a Computer Program** (using your name, obviously)
4. Then share it with [edward.pattillo@nayland.school.nz](mailto:edward.pattillo@nayland.school.nz)

### Shortcuts:

- 1.) [Work Journal](#)
- 2.) [Overview](#)
- 3.) [Task](#)
- 4.) [Project Description](#)
- 5.) [Data Structure Planning](#)
- 6.) [Heuristics](#)
- 7.) [Relevant Implications](#)

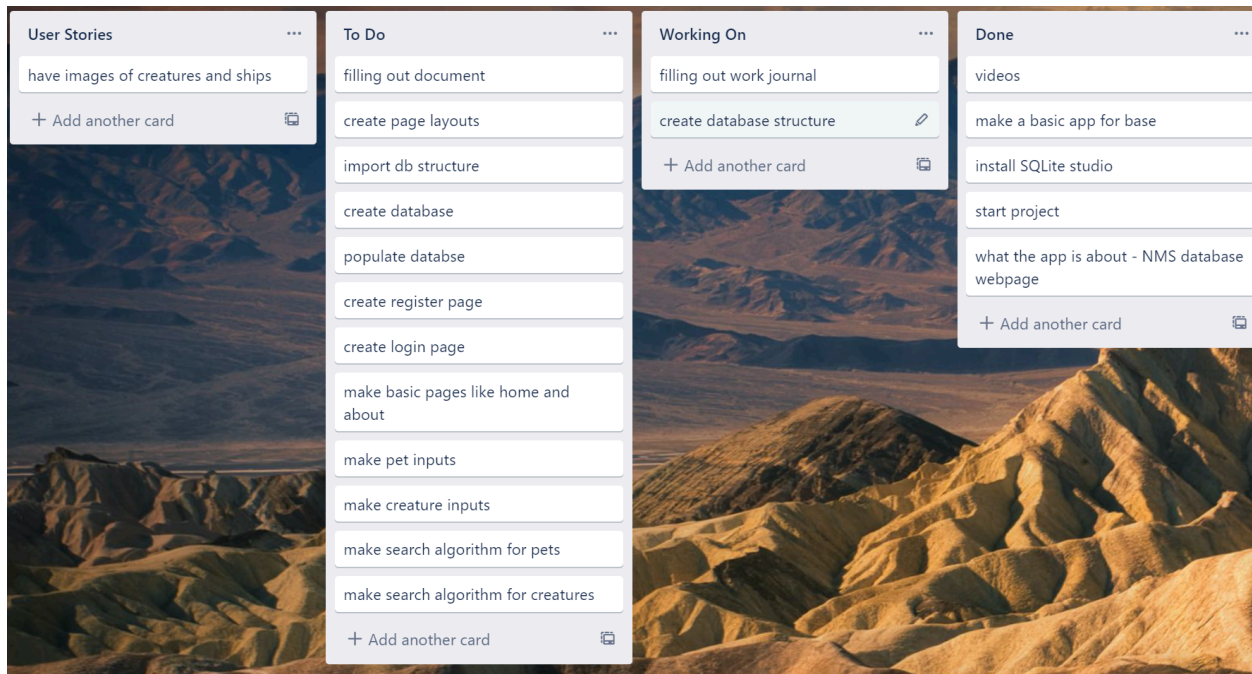
### Resources:

- OOP with Python Flask Video Course ([link](#))
- Relational Database Visualizer ([link](#))
- Khan Academy's SQL tutorials ([link](#))
- Data types list ([SQLite3](#))
- Really good explanation of JSON ([video](#))

# Work Journal

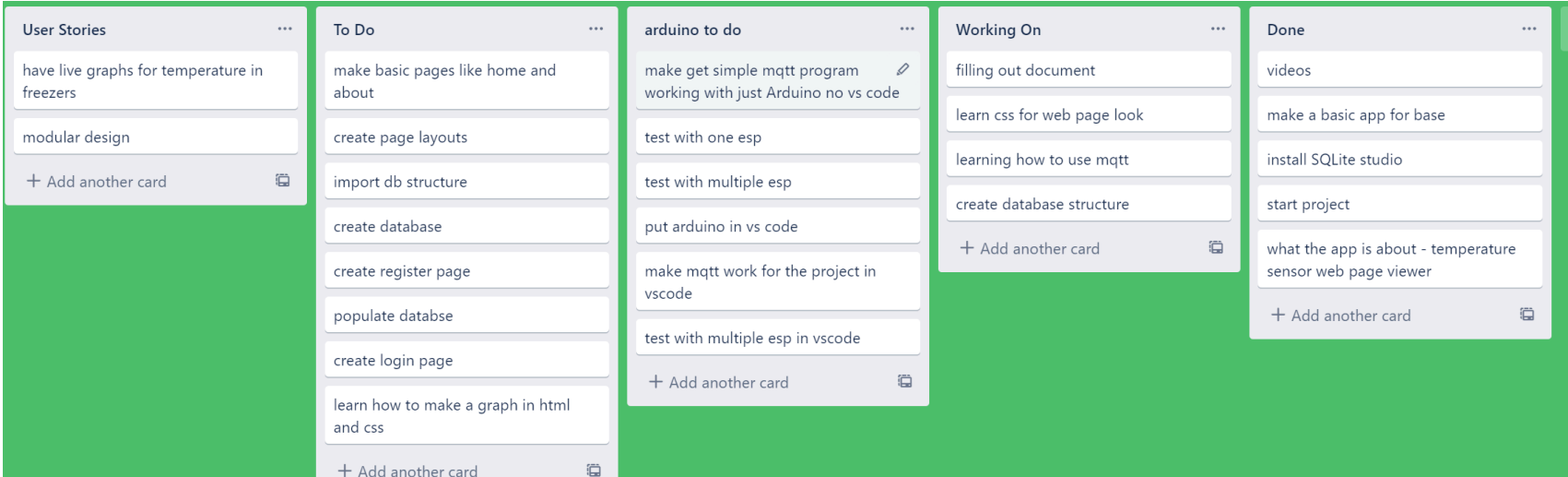
Log what you've done **each day** on the project. *This is important even if you just want a mark of **Achieved**. **Merit** and **Excellence** require that you show how your project developed throughout including but not limited to: independent research, issues you encountered and how you overcame them, ongoing testing procedures you used while working on your project, ongoing evaluation, feedback from potential users and suggestions for improvement.*  
**Add rows as needed.**

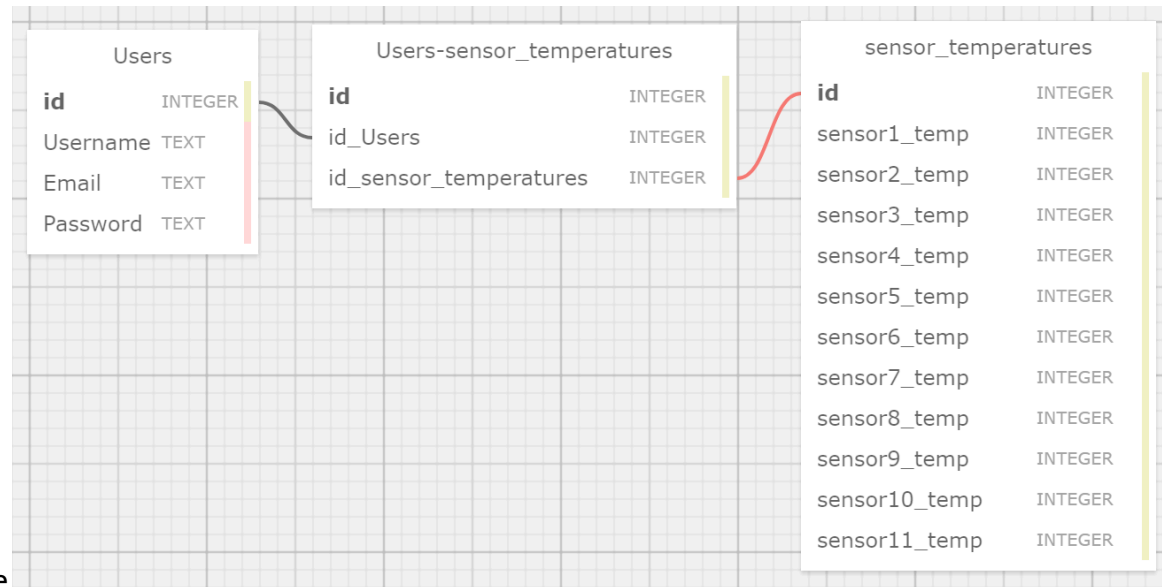
<u>Date</u>	<u>Description of work done</u>
Before 15/03/2021	Made multiple different database structures For NMS databases Made Trello Put basic flask app stuff into code Watched videos Made the base for the app
15/03/2021	Made this document Trello by date



16/03/2021 Decided to change to an Arduino to flask app project and made a new Trello for the project  
Made custom navbar colours and borders

18/03/2021 Did stuff on the style of the web page

22/03/2021	<p>Talked to Mr Daly about the project and updated my Trello and learned about buttons Filled in the project description</p>  <p>The screenshot shows a Trello board with the following tasks:</p> <ul style="list-style-type: none"> <li><b>User Stories:</b> <ul style="list-style-type: none"> <li>have live graphs for temperature in freezers</li> <li>modular design</li> <li>+ Add another card</li> </ul> </li> <li><b>To Do:</b> <ul style="list-style-type: none"> <li>make basic pages like home and about</li> <li>create page layouts</li> <li>import db structure</li> <li>create database</li> <li>create register page</li> <li>populate database</li> <li>create login page</li> <li>learn how to make a graph in html and css</li> <li>+ Add another card</li> </ul> </li> <li><b>arduino to do:</b> <ul style="list-style-type: none"> <li>make get simple mqtt program working with just Arduino no vs code</li> <li>test with one esp</li> <li>test with multiple esp</li> <li>put arduino in vs code</li> <li>make mqtt work for the project in vscode</li> <li>test with multiple esp in vscode</li> <li>+ Add another card</li> </ul> </li> <li><b>Working On:</b> <ul style="list-style-type: none"> <li>filling out document</li> <li>learn css for web page look</li> <li>learning how to use mqtt</li> <li>create database structure</li> <li>+ Add another card</li> </ul> </li> <li><b>Done:</b> <ul style="list-style-type: none"> <li>videos</li> <li>make a basic app for base</li> <li>install SQLite studio</li> <li>start project</li> <li>what the app is about - temperature sensor web page viewer</li> <li>+ Add another card</li> </ul> </li> </ul>
23/03/2021	Learned more CSS and made the navbar look a bit better
25/03/2021	<p>Learn about MQTT using the HiveMQ basic to understand the topic Made sure what temp sensor I was using and checked with John-Paul what he thought about the plan I had and if he suggested another way but he thought the plan I had was a good one</p>
27/03/2021	Bought parts I need to make the project then shipped them to me
29/03/2021	Learnt more CSS and got a basic program to display sensor data in the serial monitor
31/03/2021	Arduino board and sensor shipped to me arrived
2/04/2021	Soldered my circuit up
8/04/2021	Got all libraries I need to make a basic test for everything



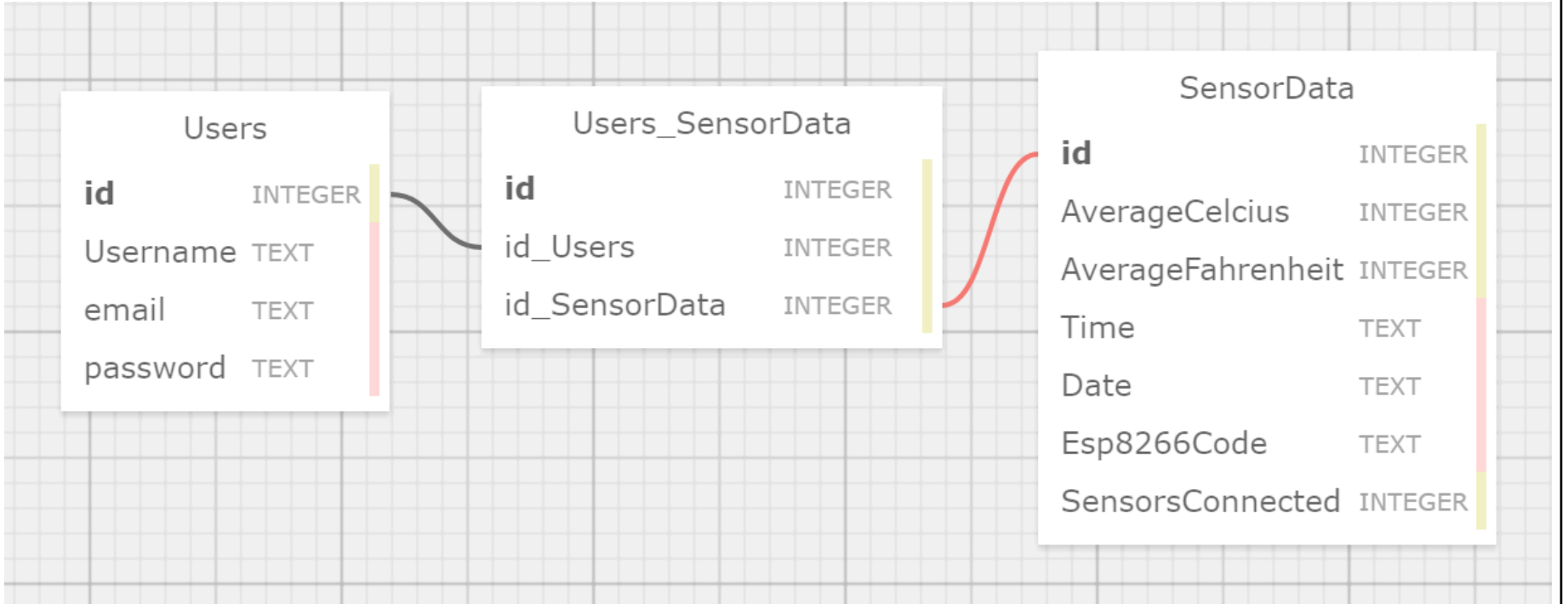
First database structure

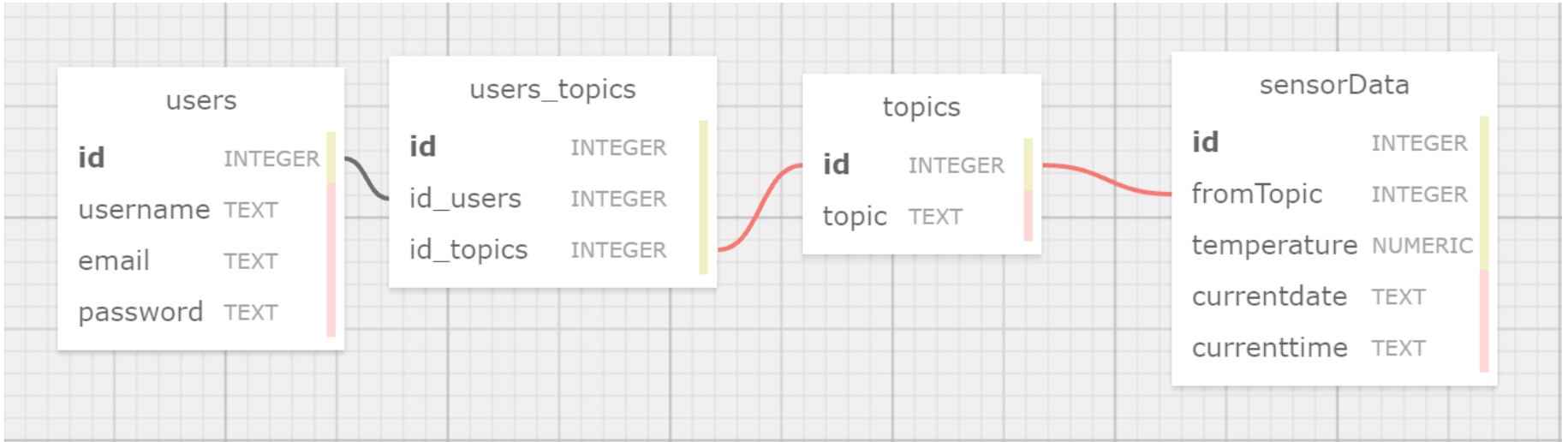
22/07/2021 Started setting up the raspberry pi to run my current program

23/07/2021 Continued working on the raspberry pi

26/07/2021 Started looking at the Arduino program for the esp8266

27/07/2021 Continued working on the Arduino program

29/07/2021	<p>Worked on the database structure</p>  <pre> graph LR     Users[Users] --- id_Users[id_Users] --- Users_SensorData[Users_SensorData]     Users_SensorData --- id[id] --- SensorData[SensorData]   </pre> <p>Worked on a layout for data display</p>
2/08/2021	Worked on the Arduino code more and looked at the python code for the MQTT broker
4/08/2021	Got the program working on raspberry pi 3
15/08/2021	Worked on the Arduino code and switched to using an esp32 instead of an esp8266 broad
2/09/2021	Worked on the Arduino code for the esp32
11/09/2021	Setup the MQTT code on the raspberry pi
12/09/2021	Working on making a test website work on the raspberry pi
11/10/2021 - 13/10/2021	Continued trying to get the test website working

14/10/2021	Got a test website working
15/10/2021	<p>Worked on the database and made sure the temperature sensor was working, as it wasn't before New database structure</p>  <pre> graph LR     users[users] --- id to id_users  users_topics[users_topics]     users_topics --- id_topics to id  topics[topics]     topics --- id to fromTopic  sensorData[sensorData]   </pre> <p>With currentdate being a DATE variable, not TEXT and currenttime being a TIME variable, not TEXT as I couldn't make them DATE or TIME in SQL designer</p>
17/10/2021	put everything that I had so far which is a login system and data collection and display on 1 website
18-19/10/2021	Worked on parts of this document like a flowchart
21/10/2021	Worked on making the data display properly and nicely and adding a topic add page and put current database structure in Tried to add an "add topic" page so the user can be subscribed to multiply topics not yet working
25/10/2021	Tried to connect added topics to the user who added them in the database
26/10/2021	Got the add topic working so it will add a new topic to the user Tried to add a delete topic page but broke the add topic page instead now neither are working
27/10/2021	Got add and delete topics to the viewed topics working but the topic when added through the website can only be 1 character long which I am trying to fix

28/10/2021	Got a bad system to show one topic data connected to a user through two different web pages which aren't very good but does the job Fixed the topic input issue
1/11/2021	Tried to get the topic search thing working
2/11/2021	Got the topic search and display related temperature data working, updated code with comments that weren't copied from previous copied function I used but related to what the code does and uploaded a basic bootstrap webpage with all the backend working
3/11/2021	Made things clearer in meaning and worked on the document
4/11/2021	Added a for loop to the MQTT functions to subscribe all topics in the database and link the sensor data to the topic Added a limit to topics length with a minimum of 4 characters and a maximum of 50 characters as that should give the user enough characters to make good names for topics
8/11/2021	Worked on document
9/11/2021	Worked on the document Looked at putting a graph in Plotly
10/11/2021	Looked into plotly/dash to graph my data
11/11/2021	Tried to get a plotly graph working
14/11/2021	Looked at the graph more
15/11/2021	Worked on graph trying to get it to display my sensor data Put more data into my database as I needed moreover some time and dates not just time for one topic to check if the graph will work properly Going to change the database so it has DateTime as one variable but will leave the old date and time variables in as to not break past things and means I can display them as separate things with greater ease as I did for the data table already on the website Got the graph to work with the new currentDateTime variable on a test website going to put the part which makes the graph into the website Put raspberry pi on over night with esp32 to collect data
16/11/2021	Got the graph on the actual website instead of a test one working with it still having the 20 most recent below in a table Fixed up all the other problems i could find and commented the code Handed in for feedback



--	--

## Overview

This project involves learning how to develop a web application using the Python-based Flask micro-framework. I've made a [comprehensive video course](#) that you need to complete in its entirety until you have a working model of the example at the end of the course. This is important. Don't try to 'cherry-pick' bits out of the videos or watch them casually without the sound on. You'll miss important points that are relevant to NCEA requirements.

## Student/Ākonga instructions

### Task/Hei Mahi

**Your job:** Design a user-friendly Flask application with persistent data. You can choose the context and purpose, provided that Mr. P has discussed this with you and has determined that it falls within the scope for all three standards.

**Required features** (**don't ignore any of these points**):

- **Discuss your project concept with Mr. P**
- The project must allow for a relational data structure, eg. one-to-many and/or many-to-many
- It needs to have an intuitive user interface with conveniences such as autocomplete, dropdowns or checkboxes.
- Data validation (ensuring the user inputs the right things)
- Users can interact with the database in the following ways:
  - **Create** entries
  - **Read** data and see it presented in a useable way
  - **Update** data by allowing for editing
  - **Delete** data entries and all associated entries that become redundant when deleted
- Allow for complex queries on data from multiple tables

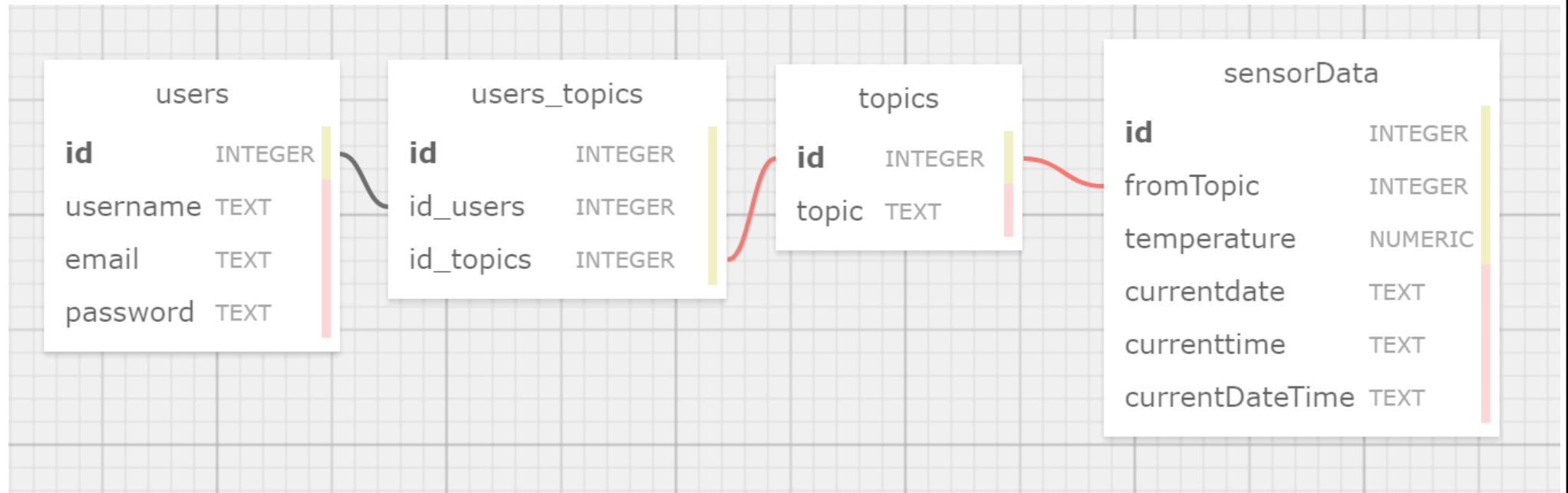
- You must present the queries in a useful way that allows users to visualize the data.
- The application **needs** to be **password protected**
- **Your code needs to be commented** to explain program structure and function and so that decisions you've made are explained and justified.
- Your code needs to use proper indentation, line spacing and naming of variables, *eg. variables starting with capital letters are reserved for classes*
- Your project needs to use meaningful descriptive names for variables, procedures, functions, labels and data file
- You need to get someone else to test your app at a few key points during its development, then collect their feedback into your work journal. Show how you've used their feedback to create a better app. ← *This isn't optional. You need to do this even to get Achieved.*

## Project Description

Who is your project for?	Mr Daly to use to view freezer temperature data easily and use to teach younger students about electrical engineering(and mechatronic engineering) if it is good enough
What specifically will it do for them?	It displays temperature data on a web page in an easy to understand the way
Paste a link to your Trello board here:	<a href="https://trello.com/b/RVie9fM3/3dt-arduino-flask-app">https://trello.com/b/RVie9fM3/3dt-arduino-flask-app</a>

## Data Structure

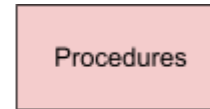
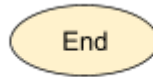
Copy and paste a screenshot of your **final data structure** from [SQL designer \(video\)](#). As you develop this, screenshot it into your work journal with a brief description of what changes you've made.



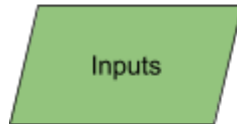
## Flowchart



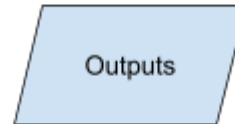
Yellow oval



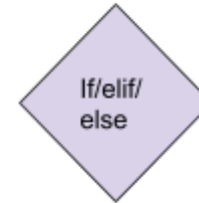
Red rectangle



Green parallelogram



Blue parallelogram



Purple diamond

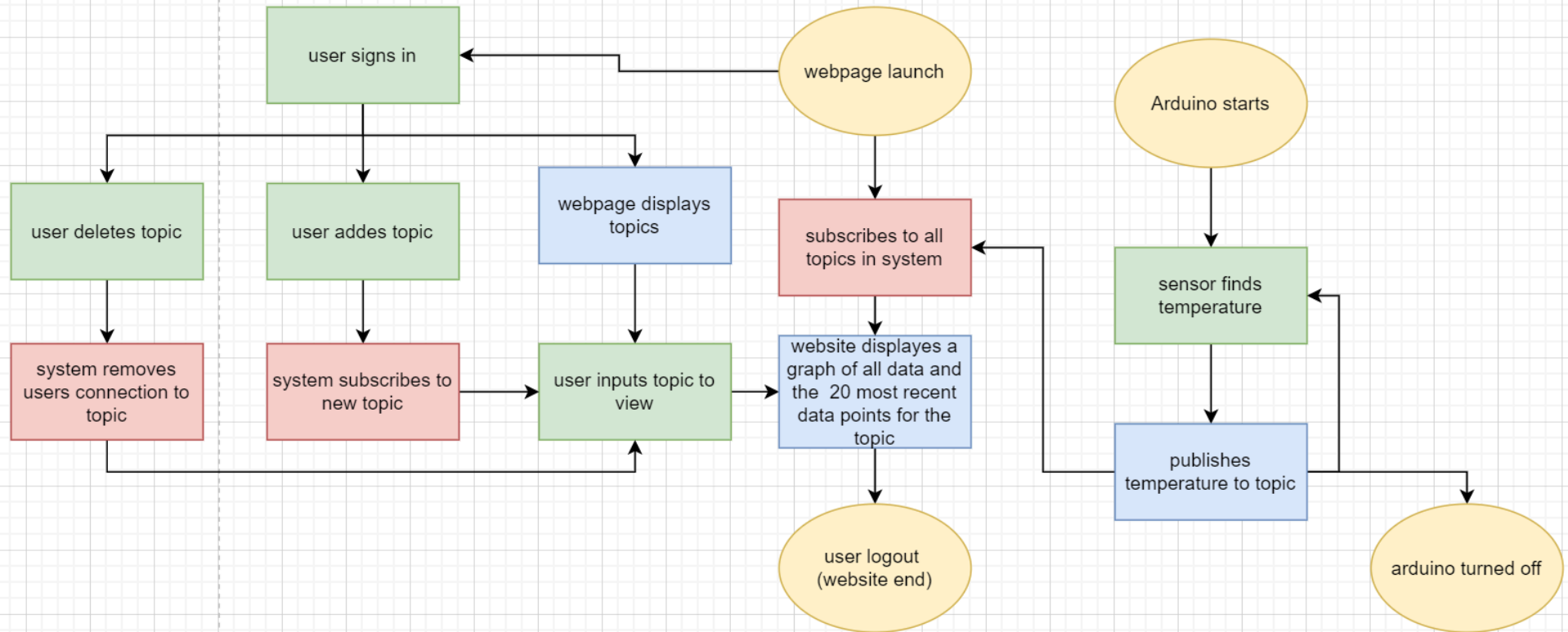
Link your flowchart from [draw.io](https://draw.io)

*(It needs to be shared with [edward.pattillo@nayland.school.nz](mailto:edward.pattillo@nayland.school.nz) just like you would a Google Doc. Give **editing** permission so I can make comments directly onto your flowchart. Please name it: **LastName, FirstName - 1DT Flowchart** )*  
*Here's **an example** of a simple flowchart with functions represented and sensible "if" statements.*

(paste link here)

<https://app.diagrams.net/#G1BVe0W19rQeIFuiUwMwjDWPM9bJ29Fkh8>

Screenshot an image of your final flowchart here:



## Input Testing Cases

Input cases describe different scenarios that occur when the user inputs data into the system you build. There are three basic types of them:

- **Expected:** These are values that you would expect the user to enter in normal circumstances. They fall within the expected range of input.
- **Exceptional (Invalid):** These are values that you wouldn't expect the user to enter in normal circumstances. They fall outside the expected range of input.
- **Boundary:** These are values that are just on the edge of what you would expect the user to enter in normal circumstances. They fall within the range of expected inputs but are almost in the range of unexpected values.

**Example:** In a simple app that gets the user to guess a number between 1 and 10:

- **Expected:** The user inputs the number 3
- **Exceptional (Invalid):** The user inputs the number 15 or enters letters instead of a number.
- **Boundary:** The user inputs the number 10 which is just on the edge of the expected range 1-10.

What is being input? (A short explanation)	Test Input Case (What would the user input?)	Type (expected, boundary or invalid)	Expected Output (How would your system respond?)	Actual Output (fill this in after testing)
Example: The program asks for the user's name.	Eddie	expected	The program will use their first name in further input prompts.	Hey, Eddie. How many games would you like to play?
Arduino sends temperature data	23 or 23.00	expected	Inserts into the database and relates to the topic it was from	Will show in topics most recent sensor data display page
Username	Jack	expected	The program will input the username into the database and display it in the navbar	Successful login/register message
	Two or 1234567890123456	Invalid	The program will tell the user it is either to short or too long	
Password	Ktchfjx4e5	expected	Will sha256 encrypt it then insert into database in user table	Successful register and login screenshot
Email	Two or 51 character email	Invalid	The program will tell the user it is either to short or too long	
	example@gmail.com	expected	Will insert into database in users table	Show register successful screenshot

Topic(add)	esp8266/tem perature	expected	The program will input the topic into the database and subscribe to the topic	<div>You have now added the fresh topic 'tjbvskjclsc nsejn' to the topics you can view.</div>
	esp32/ds18b 20/temperat ure	boundary	The program will connect users to the topic and tell them it has been used before	<div>You have now added the previously used topic 'test4' to the topics you can view.</div>
	temperature	exceptional (invalid)	The program informs the user and gets them to try again	<div>This topic was already in the topics you can view. Try again.</div>
Topic(delete)	esp8266/tem perature	expected	The program will remove the connection between the topic and user but doesn't delete the topic from the database as other users might be using it or the same user might want the data later	<div>You have now deleted the topic from what you can view. You can view it again by adding tjbvskjclsc nsejn it again</div>
	temp1	exceptional (invalid)	Tells the user can't delete a topic as the topic wasn't connected to the user in the first place	<div>Topic is not in the topics you can view meaning you can't delete it. Try again.</div>
Topic(search)	esp32/ds18b 20/temperat ure	expected	The program will retrieve all data and put it on a graph and search for the 20 most recent sensor readings in the database related to the topic	<div><div><div><div>Sensor Data</div><div>This is all data related to topic: esp32/ds18b20/temperature</div><div><div><div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div>&lt;</div></div></div></div></div></div>

Topic(Any)	two	Invalid	The program will say that it is invalid and let you try again	<div>Add Topic</div> <div>minimum of 4 characters and maximum of 50 characters for the topics name</div> <div>Topic</div> <div>two</div> <div>Field must be between 4 and 50 characters long.</div> <div>Submit</div>
	12345678901234567890123456789012345678901(51 characters)	Invalid	The program will say that it is invalid and let you try again	<div>Add Topic</div> <div>minimum of 4 characters and maximum of 50 characters for the topics name</div> <div>Topic</div> <div>123456789012345678901234567890123456789012345678901</div> <div>Field must be between 4 and 50 characters long.</div> <div>Submit</div>

## Heuristics

Remember these? Good. Now you need to point out, with annotated screenshots, how your user interface satisfies these heuristics.

	Usability Heuristic	Keywords / Reference
U	User control and freedom	Eg, support undo/redo
H	Help users recognize, diagnose and recover from errors	Eg, useful error messages



<b>A</b>	<b>A</b> esthetic and minimalist design	Only show users what they need to know, keep the interface clean
<b>V</b>	<b>V</b> isibility of system status	Give users immediate feedback to let them know what the system is doing, eg. loading bar, spinning icon, confirmation message 'saved'
<b>E</b>	<b>E</b> rror prevention	Keep users from making mistakes, eg. dropdown selectors, confirmation screen "Are you sure you want to delete?"
<b>C</b>	<b>C</b> onsistency and standards	Use common software conventions, eg. CTRL+Z = undo, also use consistent iconography
<b>H</b>	<b>H</b> elp and documentation	Help should be easy to access and easy to find exact info from.
<b>A</b>	<b>(A</b> ccelerators) aka Flexibility and efficiency of use	Eg, Keyboard shortcuts for expert users
<b>R</b>	<b>R</b> ecognition rather than recall	Don't force the user to remember things, eg, History tool, Open recent dialogue
<b>M</b>	<b>M</b> atch between systems and real world	The system should speak the users' language. Eg, red for stop, green for go
<b>Screenshot (not of the whole page)</b>		<b>Explain which heuristic is being addressed, and how you feel it has been met.</b>

You have now deleted the topic from what you can view. You can view it again by adding test2 it again

You have now added the previously used topic 'test2' to the topics you can view.

You have now added the fresh topic 'test5' to the topics you can view.

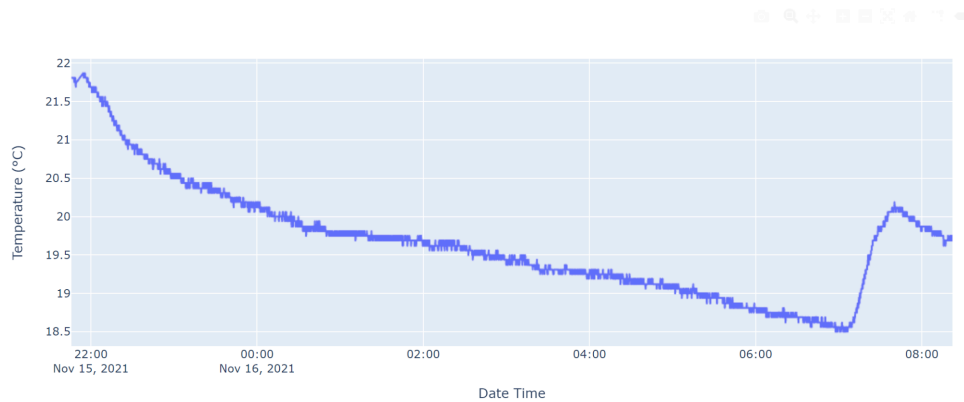
### User control and Freedom,

The user can add and delete topics without the worry of permanently deleting the sensor data related to the topic

Topic ID	Topic
1	esp32/ds18b20/temperature
17	test3
2	test
16	test2

## Sensor Data

This is all data related to topic: esp32/ds18b20/temperature



### Aesthetics and minimalist design,

The design doesn't have images where text can be used with the unit for data shown at the top of the table for format

Average Temperature Celcius (°C)	Date Collected (YYYY-MM-DD)	Time Collected (HH-MM-SS)
22.31 °C	2021-10-17	17:57:43
22.31 °C	2021-10-17	17:57:33
22.37 °C	2021-10-17	17:57:23
22.31 °C	2021-10-17	17:57:13
22.37 °C	2021-10-17	17:57:03
22.31 °C	2021-10-17	17:56:53
22.31 °C	2021-10-17	17:56:43
22.31 °C	2021-10-17	17:56:33
22.31 °C	2021-10-17	17:56:23

### Flash messages

You are now logged in, Jack.

You have now added the previously used topic 'test2' to the topics you can view.

### Visibility of system status,

The design will display a flash message when any topic is added or deleted to tell the user what has happened with green colours meaning it works and red meaning it didn't

Topic

y

Field must be between 4 and 50 characters long.

There was no data assigned to topic 'test3'.

### Error prevention,

The deleting system only deletes the connection between the user and topic as multiple users could be viewing the topic or the same user might want to view the past data at a later date which they can by reading the topic which the system tells the user if the topic has been used

Topic is not in the topics you can view meaning you can't delete it. Try again.

Topic

y

Field must be between 4 and 50 characters long.

### Help users recognize, diagnose and recover from errors

Error messages tell the user what has happened to cause the error shown in flash messages with a red background

Something went wrong. No topic added to viewed topics for user: Jack

There was no data assigned to topic 'test3'.

## Relevant Implications

### Social

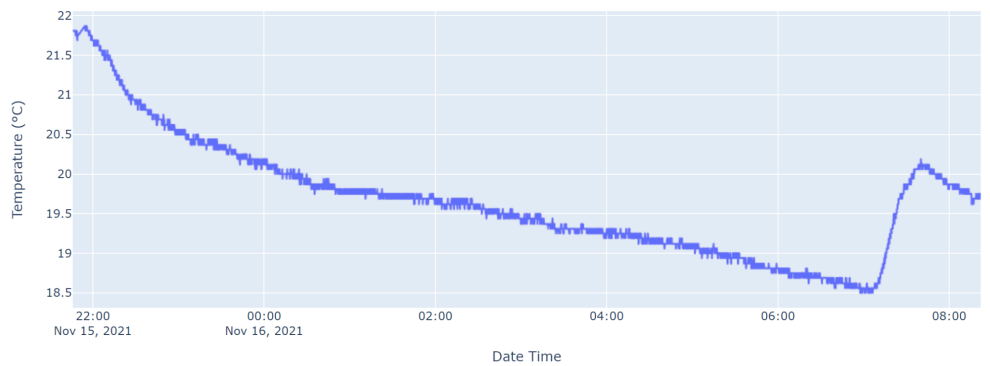
- Is the data outcome suitable for the intended audience (images and text targeted to the right age group etc)?
- What does the client want, and does your database deliver?

### Cultural

- Culturally appropriate image and text, especially if the international audience
- Is the content age-appropriate?

<b>Legal*</b> <ul style="list-style-type: none"> <li>• Did you avoid breaking copyright?</li> <li>• Did you get permission to use images, text?</li> <li>• Did you simply copy and paste from somewhere else?</li> <li>• Did you reference where you got your text/images from if necessary?</li> </ul>	<b>End-user considerations*</b> <ul style="list-style-type: none"> <li>• What does the user want, and can you provide it?</li> <li>• Who is using your database? – old people need bigger fonts and need something easy to use</li> <li>• Appropriate content for gender, age, ethnic groups using your database</li> </ul>
<b>Ethical</b> <ul style="list-style-type: none"> <li>• Site avoids racist, sexist content</li> <li>• Has appropriate words/content</li> </ul>	<b>Aesthetics</b> <ul style="list-style-type: none"> <li>• Making something beautiful, giving it a pleasing appearance</li> <li>• Did you use the company logo, company name on forms and reports?</li> </ul>
<b>Usability*</b> <ul style="list-style-type: none"> <li>• Consistent look and feel for all pages</li> <li>• Hierarchy of headings, headings stand out</li> <li>• Organized content (making it easier to find things)</li> <li>• Readable font</li> <li>• Is the information in the database correct and up to date?</li> <li>• Is the data correct and reliable? E.g Avoid an entry like “Swimmin” (spelling incorrect so this person won’t list up under “Swimming” queries). To address this you could use validation or drop-down lists.</li> </ul>	<b>Sustainability and future-proofing*</b> <ul style="list-style-type: none"> <li>• Using correct naming conventions (tbl, qry, frm, rpt, no spaces)</li> <li>• Using correct field names, sizes, and types</li> <li>• To save the environment you should ensure that card info fits on the paper (no blank pages get created) and that unnecessary spaces or gaps have been removed</li> </ul>
<b>Privacy*</b> <ul style="list-style-type: none"> <li>• Don’t use photos of people without permission</li> <li>• Ensure that private information is kept private (use a password, include “Confidential” on all forms and/or reports etc)</li> </ul>	<b>Health and safety implications</b> <ul style="list-style-type: none"> <li>• Ensuring the project isn’t too big so won’t overwhelm or wear you out</li> <li>• Taking regular breaks (rest eyes, rest hands from typing, circulate blood etc)</li> </ul>
<b>Functionality*</b> <ul style="list-style-type: none"> <li>• What is the purpose of the database, and does it achieve this?</li> <li>• Do the forms, queries and reports all work as expected?</li> <li>• What does the client want, and does the database deliver?</li> </ul>	<b>Accessibility*</b> <ul style="list-style-type: none"> <li>• Use colour-blind friendly colours</li> <li>• Don’t rely on colour to convey information (eg: Red = error, use an icon or text as well)</li> <li>• Use of limited colours (for colour blindness)</li> <li>• Consideration of dyslexia (special font, left-aligned text, spacing around text and images to separate them)</li> </ul>

For **achieved**, choose *at least* one bullet from **any five** implications above and describe them. Chose ones that relate to your project.  
For **merit/excellence** you will need to show evidence of how your project **addresses** at least eight of these relevant implications.

Relevant Implication	Describe how your project relates to this concept	Show how you addressed this implication (annotated screenshots)
Legal	I have used only python and Arduino libraries that are free to use and programmed my code myself	<pre> 1  # import flash application 2  from flask import Flask, render_template, request, redirect, url_for, flash, session, logging 3  # import our config file config.py 4  from config import Config 5  # import functools 6  from functools import wraps 7  # import our classes in the classes folder 8  from classes.user import User 9  from classes.sensor import Sensor 10 11  import classes.forms 12 13  # libraries used for commented out mqtt program whilst on laptop and not needing testing the progrma with data 14  #import paho.mqtt.client as mqtt 15  #import json 16  import sqlite3 17 18  # create an object with our Config class properties 19  config = Config() </pre>
Functionality	The purpose of my project was to display temperature sensor data from an Arduino esp board which sends the data over MQTT to a raspberry pi which then runs a local website to display the data, which my project does	<h3>Sensor Data</h3> <p>This is all data related to topic: esp32/ds18b20/temperature</p> 

Average Temperature Celcius (°C)	Date Collected (YYYY-MM-DD)	Time Collected (HH-MM-SS)
22.31 °C	2021-10-17	17:57:43
22.31 °C	2021-10-17	17:57:33
22.37 °C	2021-10-17	17:57:23
22.31 °C	2021-10-17	17:57:13
22.37 °C	2021-10-17	17:57:03
22.31 °C	2021-10-17	17:56:53
22.31 °C	2021-10-17	17:56:43
22.31 °C	2021-10-17	17:56:33
22.31 °C	2021-10-17	17:56:23

Sustainability and future-proofing

I have used correct variable names and have systems to add other temperature sensors to the website MQTT subscription list  
I used clear database names for all my database variables

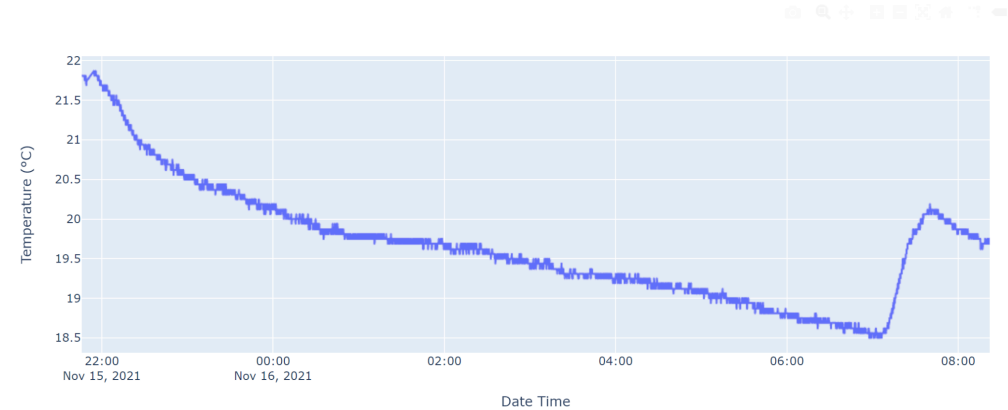
```
CREATE TABLE 'users_topics' (  
  'id' INTEGER DEFAULT NULL PRIMARY KEY AUTOINCREMENT,  
  'id_users' INTEGER DEFAULT NULL REFERENCES 'users' ('id'),  
  'id_topics' INTEGER DEFAULT NULL REFERENCES 'topics' ('id')  
);  
  
CREATE TABLE 'topics' (  
  'id' INTEGER DEFAULT NULL PRIMARY KEY AUTOINCREMENT,  
  'topic' TEXT DEFAULT NULL  
);  
  
CREATE TABLE 'users' (  
  'id' INTEGER DEFAULT NULL PRIMARY KEY AUTOINCREMENT,  
  'username' TEXT DEFAULT NULL,  
  'email' TEXT DEFAULT NULL,  
  'password' TEXT DEFAULT NULL  
);  
  
CREATE TABLE 'sensorData' (  
  'id' INTEGER DEFAULT NULL PRIMARY KEY AUTOINCREMENT,  
  'fromTopic' INTEGER DEFAULT NULL REFERENCES 'topics' ('id'),  
  'temperature' NUMERIC DEFAULT NULL,  
  'currentdate' DATE DEFAULT NULL,  
  'currenttime' TIME DEFAULT NULL  
);
```



Accessibility	The site text is clear with a very basic colour design and says what all data presented means in what units	<h2>Sensor Data</h2> <p>This is the 20 most recent sensor readings from the topic</p> <p>To view another topics data go back to the Sensor Data tab and enter another topic</p>												
End-user considerations	The user wants to be able to subscribe to a topic and then view the data from the topic on a webpage with the most recent data showing up	<h2>Add Topic</h2> <p>minimum of 5 characters and maximum of 50 characters for the topics name</p> <p>Topic</p> <div><input type="text"/></div> <div>Submit</div> <h2>All Topics added to viewed list</h2> <p>these are all the topics you have added to the viewed topics you can view</p> <table><tr><th>Topic ID</th><th>Topic</th></tr><tr><td>1</td><td>esp32/ds18b20/temperature</td></tr><tr><td>17</td><td>test3</td></tr><tr><td>2</td><td>test</td></tr><tr><td>16</td><td>test2</td></tr><tr><td>20</td><td>test5</td></tr></table>	Topic ID	Topic	1	esp32/ds18b20/temperature	17	test3	2	test	16	test2	20	test5
Topic ID	Topic													
1	esp32/ds18b20/temperature													
17	test3													
2	test													
16	test2													
20	test5													

Sensor Data

This is all data related to topic: esp32/ds18b20/temperature



Average Temperature Celcius (°C)	Date Collected (YYYY-MM-DD)	Time Collected (HH-MM-SS)
22.31 °C	2021-10-17	17:57:43
22.31 °C	2021-10-17	17:57:33
22.37 °C	2021-10-17	17:57:23
22.31 °C	2021-10-17	17:57:13
22.37 °C	2021-10-17	17:57:03
22.31 °C	2021-10-17	17:56:53
22.31 °C	2021-10-17	17:56:43
22.31 °C	2021-10-17	17:56:33
22.31 °C	2021-10-17	17:56:23

Usability	The websites colour scheme and design is the same throughout the webpage with everything being clear	Shown in screenshots of the website above
Privacy	The website sha256 encrypts the user's password in the database so there is no entry of the password in the database	<pre># executes some SQL to select the user's details cur.execute("SELECT * FROM users WHERE username = ?",[username]) # retrieves this row from the dictionary and assigns the # result to the variable result result = cur.fetchone()  # if the result has returned as True, it means that # the user already exists in the database so we return # False to app.py where this method was called if result:     return False else:     # if the user doesn't already exist in the database     # we encrypt the password they've submitted using     # the sha256_crypt library     passwordHashed = sha256_crypt.encrypt(str(password))      # create cursor     cur = con.cursor()     # insert new user into database     cur.execute("INSERT INTO users (username,password,email) VALUES (?,?,?)",[username,passwordHashed,email])     # this commits the row in to the users table     con.commit()     cur.close()     # now we return True to app.py     return True</pre>

Ethical	The website doesn't use sexist or racist words or content and it doesn't use any inappropriate words or content	Shown by screenshots of the website above
---------	---	---