Week 9 Commentary

1. The first program was relatively simple to make, as I created a SQL SELECT statement, wrote the following statement out:

```
$stmt = mysqli_stmt_init($conn);
$sqld = "SELECT * FROM `db_imp` ORDER BY `datetime` DESC";
```

   To that end, I produced the program code that gave a table with 3 columns, and (at this point) only the latest result as recorded from the ElectricImp device as can be previewed below:

ID	Date Time	Temp
2325ccaa5ba5ceee	2019-11-06 14:01:05	{"external" : -44.7 ,"internal

   One of the columns is ID, which is the ID for the ElectricImp device that is collecting this data. The Date Time column is when this row of data was collected, and the Temp is the temperature that was recorded by the temperature sensor.

2. This was also rather simple to do, as I simply altered the `SELECT` statement which worked to show the values from the first task, and added `LIMIT 10`, which limits the output of the query to only 10 rows of data, and as this data is being sorted in descending order of time, with the newest at the start

```
$stmt = mysqli_stmt_init($conn);
$sqld = "SELECT * FROM `db_imp` ORDER BY `datetime` DESC LIMIT 10" ;
```

   As you can see above, at the end of the second line in the SQL query there is "`ORDER BY 'datetime' DESC LIMIT 10`", which when ran sorts all rows in the `db_imp` table by datetime in descending order and shows only the newest 10 rows of data. Like the first task, this code when ran produced the following result:

ID	Date Time	Temp
2325ccaa5ba5ceee	2019-11-06 14:01:05	{"external" : -44.7 ,"internal
2325ccaa5ba5ceee	2019-11-06 14:00:35	{"external" : -45.9 ,"internal
2325ccaa5ba5ceee	2019-11-06 14:00:05	{"external" : -45.5 ,"internal
2325ccaa5ba5ceee	2019-11-06 13:59:35	{"external" : -45.3 ,"internal
2325ccaa5ba5ceee	2019-11-06 13:59:05	{"external" : -44.5 ,"internal
2325ccaa5ba5ceee	2019-11-06 13:58:35	{"external" : -45.1 ,"internal
2325ccaa5ba5ceee	2019-11-06 13:58:05	{"external" : -45.5 ,"internal
2325ccaa5ba5ceee	2019-11-06 13:57:35	{"external" : -45.5 ,"internal
2325ccaa5ba5ceee	2019-11-06 13:57:05	{"external" : -45.3 ,"internal
2325ccaa5ba5ceee	2019-11-06 13:56:35	{"external" : -45.9 ,"internal

3. I research online how to do this, and the following code was added to the php file, allowing for the table to update automatically every 10 seconds:

```
        function refreshTable(){
            $('#tableHolder').load('model/getIotTable.php', function(){
            setTimeout(refreshTable, 30000); //refresh every 30 seconds, the same
time the ElectricImp updates the database table
```

```
            });
```

The code above is a small piece of code, the aim and function of which is to load the table, as shown in the second task, but to update the table every 30 seconds, which is how often the database table updates when

4. In order to display the data as a graph via jQuery, I had to conduct some research on how this could be implemented with minimal bother, as I had not done something of this ilk before in web development. I found one (the page in which it has been sourced from can be viewed here) which allowed data to be displayed in a line graph, using an array as the basis for which to enter the data from. Although the *state* field which contains the temperature does contain the temperature, it is done so inside a json array, and so for ease of use I have extracted the values from each row of the most recent 10 values at time of writing, and then will try to have this dynamic later on.